**SISTEM MANAJEMEN PENGELOLAAN PRESENSI PROYEK DI PT. LUNDIN BERBASIS WEB**

# SAMPUL LUAR

**PROPOSAL PROYEK AKHIR**



**Oleh :**

**TUSFENDI**

**NIM. 361755401175**

**PROGRAM STUDI DIPLOMA III**

**TEKNIK INFORMATIKA**

**POLITEKNIK NEGERI BANYUWANGI**

**2020**

**SISTEM MANAJEMEN PENGELOLAAN PRESENSI PROYEK DI PT. LUNDIN BERBASIS WEB**

# HALAMAN JUDUL

**PROPOSAL PROYEK AKHIR**



**Proyek Akhir Ini Dibuat dan Diajukan untuk Memenuhi Salah Satu Syarat Kelulusan Program Studi Diploma III Teknik Informatika dan Mencapai Gelar Ahli Madya (A.Md)**

**Oleh :**

**TUSFENDI**

**NIM. 361755401175**

**PROGRAM STUDI DIPLOMA III**

**TEKNIK INFORMATIKA**

**POLITEKNIK NEGERI BANYUWANGI**

**2020**

--*Halaman ini sengaja dikosongkan--*

# HALAMAN PENGESAHAN

**PROPOSAL PROYEK AKHIR**

Judul : Sistem Manajemen Pengelolaan Presensi Proyek di PT. Lundin Berbasis Web

Oleh : Tusfendi

NIM. : 361755401175

**Telah diuji pada :**

Hari : Selasa

Tanggal : 11 Februari 2020

Tempat : Laboratorium Program 2

**Mengetahui / Menyetujui :**

|  |  |
| --- | --- |
| Dosen Penguji : | Dosen Pembimbing : |
| 1. I Wayan Suardinata, S. Kom, M.T   NIP. 198010222015041001 | 1. Eka Mistiko Rini, S.Kom, M.Kom   NIP. 198310202014042001 |
| 1. Dedy Hidayat Kusuma, S.T. M.Cs   NIK. 2011.36.079 | 2. Alif Akbar Fitrawan, S.Pd, M.Kom  NIP. 199104202018031002 |

--*Halaman ini sengaja dikosongkan--*

# DAFTAR ISI

Halaman

[HALAMAN JUDUL i](#_Toc32269431)

[HALAMAN PENGESAHAN iii](#_Toc32269432)

[DAFTAR GAMBAR vii](#_Toc32269433)

[DAFTAR TABEL ix](#_Toc32269434)

[BAB 1 PENDAHULUAN 1](#_Toc32269435)

[1.1 Latar Belakang 1](#_Toc32269436)

[1.2 Perumusan Masalah 2](#_Toc32269437)

[1.3 Tujuan 2](#_Toc32269438)

[1.4 Manfaat 3](#_Toc32269439)

[1.5 Batasan Masalah 3](#_Toc32269440)

[BAB 2 TINJAUAN PUSTAKA 5](#_Toc32269441)

[2.1 Landasan Teori 5](#_Toc32269442)

[2.1.1 Metode RAD 5](#_Toc32269443)

[2.1.2 *Hypertext Preprocessor* (PHP) 7](#_Toc32269444)

[*2.1.3* *Framework* *Laravel* 8](#_Toc32269445)

[2.1.4 Basis Data 10](#_Toc32269446)

[2.1.5 MariaDB 11](#_Toc32269447)

[*2.1.6* *Use Case* *Diagram* 11](#_Toc32269448)

[2.1.7 Diagram Aktivitas 13](#_Toc32269449)

[*2.1.8* *Black* *Box Testing* 15](#_Toc32269450)

[2.2 Penelitian Terdahulu 16](#_Toc32269451)

[BAB 3 METODE PENELITIAN 17](#_Toc32269452)

[3.1 Tempat dan Waktu Penelitian 17](#_Toc32269453)

[3.2 Metode Pengembangan Aplikasi 17](#_Toc32269454)

[3.2.1 Perancangan 18](#_Toc32269455)

[3.2.2 Analisis Kebutuhan 18](#_Toc32269456)

[3.2.3 Desain Sistem 18](#_Toc32269457)

[3.2.4 Implementasi Desain 19](#_Toc32269458)

[3.2.5 Pengujian Sistem 19](#_Toc32269459)

[3.2.6 Implementasi Sistem 19](#_Toc32269460)

[3.3 Gambaran Umum Sistem 19](#_Toc32269461)

[3.3.1 Sistem yang Berjalan 19](#_Toc32269462)

[3.3.2 Sistem yang Diusulkan 21](#_Toc32269463)

[3.4 Spesifikasi Sistem 22](#_Toc32269464)

[3.4.1 Aturan Sistem 22](#_Toc32269465)

[3.4.2 Struktur Menu 23](#_Toc32269466)

[*3.4.3* *Use Case Diagram* 24](#_Toc32269467)

[*3.4.4* *Activity Diagram* 28](#_Toc32269468)

[3.4.5 *Mockup* Sistem 30](#_Toc32269469)

[3.4.6 Relasi Antar Tabel 39](#_Toc32269470)

[DAFTAR PUSTAKA 46](#_Toc32269471)

# DAFTAR GAMBAR

Halaman

[Gambar 2. 1 Alur Metode RAD 5](#_Toc31615018)

[Gambar 3. 1 Metode RAD 18](#_Toc31864718)

[Gambar 3. 2 Alur Presensi Proyek Kapal saat ini 20](#_Toc31864719)

[Gambar 3. 3 Sistem yang akan dibuat 21](#_Toc31864720)

[Gambar 3. 4 Struktur Menu 23](#_Toc31864721)

[Gambar 3. 5 *Use case diagram* 24](#_Toc31864722)

[Gambar 3. 6 *Activity Diagram Login* 28](#_Toc31864723)

[Gambar 3. 7 *Activity Diagram* Presensi Proyek 29](#_Toc31864724)

[Gambar 3. 8 Halaman *Scan QR Code* Pegawai 30](#_Toc31864725)

[Gambar 3. 9 Pesan Gagal *Scan QR Code* Pegawai 31](#_Toc31864726)

[Gambar 3. 10 Halaman *Scan QR Code* Pekerjaan pada Pegawai 32](#_Toc31864727)

[Gambar 3. 11 Halaman Gagal *Scan QR Code* Pekerjaan 32](#_Toc31864728)

[Gambar 3. 12 Pesan Sukses Presensi Proyek 33](#_Toc31864729)

[Gambar 3. 13 Halaman *Login* 33](#_Toc31864730)

[Gambar 3. 14 Beranda Manajer departemen 34](#_Toc31864731)

[Gambar 3. 15 Beranda Admin Proyek 34](#_Toc31864732)

[Gambar 3. 16 Halaman Proyek pada Admin Proyek 35](#_Toc31864733)

[Gambar 3. 17 Halaman Tambah Proyek pada Admin Proyek 35](#_Toc31864734)

[Gambar 3. 18 Halaman Lokasi pada Admin Proyek 36](#_Toc31864735)

[Gambar 3. 19 Halaman Tambah Lokasi pada Admin Proyek 36](#_Toc31864736)

[Gambar 3. 20 Halaman Pegawai pada Admin Proyek 37](#_Toc31864737)

[Gambar 3. 21 Halaman Tambah Pegawai pada Admin Proyek 37](#_Toc31864738)

[Gambar 3. 22 Halaman Jam Kerja pada Admin Proyek 38](#_Toc31864739)

[Gambar 3. 23 Halaman Tambah Jam Kerja pada Admin Proyek 38](#_Toc31864740)

[Gambar 3. 24 Halaman *Cetak QR Code* Pegawai 39](#_Toc31864741)

[Gambar 3. 25 Halaman *Cetak QR Code* Pekerjaan 39](#_Toc31864742)

[Gambar 3. 26 *Entity Relationalship Diagram* 40](#_Toc31864743)

--*Halaman ini sengaja dikosongkan--*

# DAFTAR TABEL

Halaman

[Tabel 3. 1 Jadwal Penelitian 17](#_Toc32822646)

[Tabel 3. 2 Aturan Sistem 22](#_Toc32822647)

[Tabel 3. 3 Aktor *Use Case Diagram* 24](#_Toc32822648)

[Tabel 3. 4 Keterangan *Use Case Login* 25](#_Toc32822649)

[Tabel 3. 5 Keterangan *Use Case* Melihat Data Rekaputulasi Proyek 25](#_Toc32822650)

[Tabel 3. 6 Melihat Laporan Harian 25](#_Toc32822651)

[Tabel 3. 7 Melihat Data Keterlambatan Presensi 26](#_Toc32822652)

[Tabel 3. 8 Keterangan *Use Case* Memindai Kode Pegawai 26](#_Toc32822653)

[Tabel 3. 9 Keterangan *Use Case* Melakukan Presensi Pekerjaan 26](#_Toc32822654)

[Tabel 3. 10 Keterangan *Use Case* Mengelola Jam kerja 27](#_Toc32822655)

[Tabel 3. 11 Keterangan *Use Case* Mengelola Data Proyek Kapal 27](#_Toc32822656)

[Tabel 3. 12 Keterangan *Use Case* Mengelola Data Presensi Proyek Kapal 27](#_Toc32822657)

[Tabel 3. 13 Keterangan *Use Case* Mengelola Data Pegawai 27](#_Toc32822658)

[Tabel 3. 14 Keterangan *Use Case* Mengelola Data Pekerjaan Proyek 28](#_Toc32822659)

[Tabel 3. 15 Tabel *user* 40](#_Toc32822660)

[Tabel 3. 16 Tabel pegawai 40](#_Toc32822661)

[Tabel 3. 17 Tabel jabatan\_pegawai 41](#_Toc32822662)

[Tabel 3. 18 Tabel lokasi 41](#_Toc32822663)

[Tabel 3. 19 Tabel proyek 41](#_Toc32822664)

[Tabel 3. 20 Tabel Pekerjaan 42](#_Toc32822665)

[Tabel 3. 21 Tabel pekerjaan\_proyek 43](#_Toc32822666)

[Tabel 3. 22 Tabel jam\_kerja 43](#_Toc32822667)

[Tabel 3. 23 Tabel riwayat\_presensi 44](#_Toc32822668)

[Tabel 3. 24 Tabel kelompok pegawai 45](#_Toc32822669)

--*Halaman ini sengaja dikosongkan--*

# BAB 1

**PENDAHULUAN**

## Latar Belakang

PT. Lundin merupakan salah satu perusahaan manufaktur kapal swasta yang berlokasi di Banyuwangi, Jawa Timur. Berdiri sejak Tahun 1997, Lundin sudah memproduksi lebih dari 250 kapal. PT. Lundin merupakan produsen kapal yang mengambil basis teknologi di Swedia dan telah mengekspor lebih dari 17 galangan kapal ke sejumlah negara di benua Eropa. PT. Lundin mendapat fasilitas penangguhan pengenaan bea masuk dan pajak dalam rangka impor dari Direktorat Jenderal Bea Cukai Kementerian Keuangan. PT. Lundin masuk dalam area kawasan berikat. Pabrik ini berlokasi di Jalan Lundin no 1, Sukowidi, Banyuwangi.

Salah satu departemen di PT. Lundin adalah departemen produksi yang memiliki seorang manajer departemen, admin proyek dan beberapa supervisor yang memimpin kelompok kerja. Kelompok kerja terdiri dari 10 sampai 15 orang pegawai produksi. Setiap pegawai produksi biasanya mengerjakan 2 sampai 3 pekerjaan proyek perharinya. *Jobdesk* proyek kapal diberikan oleh manajer departemen kepada kepada supervisor. Supervisor membagi pekerjaan dengan kelompok kerjanya. Masing-masing kelompok kerja memiliki tugas dan bagian tertentu, seperti kelompok kerja *welding,* kelompok kerjapembuatan lambung kapal (*hull*) dan kelompok kerja instalasi mesin. Admin proyek bertugas untuk mengelola presensi proyek kapal.

Sistem pencatatan presensi proyek yang berjalan saat ini dilakukan secara manual. Admin proyek mencetak data pegawai untuk pencatatan aktivitas kerja disetiap proyek. Pencatatan presensi proyek dilakukan dalam 2 sesi dengan lama setiap sesi kerjanya adalah 4 jam kerja. Sesi pertama pada pagi hari hingga sebelum jam istirahat dan sesi kedua setelah istirahat hingga jam pulang. Pencatatan presensi proyek dilakukan dengan berkeliling dan membutuhkan waktu yang lama. Admin proyek mencatat pekerjaan pegawai berdasarkan proyek dan deskripsi proyek yang disampaikan oleh setiap pegawai produksi. Data presensi proyek tersebut akan dimasukan ke *form* dalam bentuk *excel* setiap selesai berkeliling disetiap sesi 1 dan 2. Data yang dimasukan diantaranya adalah sesi kerja, identitas pegawai, proyek dan pekerjaan proyek. Lama proses *input* data berlangsung sekitar 1 jam, hal tersebut terjadi karena admin proyekharus mengetik manual data kedalam *form*.

Berdasarkan hasil observasi diatas, pada sistem presensi proyek yang berjalan di PT. Lundin masih terdapat beberapa permasalahan diantaranya adalah proses memasukkan data presensi ke *form* yang membutuhkan waktu lama dan kurang akuratnya data karena rawan kesalahan ketika proses *input* dan data terbatas pada parameter sesi 1 serta 2. Hal tersebut berdampak pada *monitoring* rekapilutasi presensi proyek oleh manajer departemen tidak dapat dilakukan secara *real time.* Manajer departemen juga membutuhkan proses dan waktu untuk melihat berapa lama total jam yang dibutuhkan untuk pekerjaan pada setiap proyek kapal dan lama waktu untuk setiap proyek. Hal tersebut memperlambat produktivitas manajer departemen.

Solusi yang diajukan untuk beberapa permasalahan tersebut adalah dengan diajukannya proyek akhir berjudul “Sistem Manajemen Pengelolaan Presensi Proyek di PT. Lundin Berbasis Web”. Diharapkan dengan adanya sistem manajemen yang dibuat dalam proyek akhir ini, dapat digunakan oleh departemen produksi di PT. Lundin. Presensi proyek tidak lagi terbatas pada pembagian sesi waktu yang yang dilakukan saat ini sehingga data lebih akurat dan manajer departemen dapat melakukan *monitoring* rekapitulasi presensi pekerjaan proyek secara *real-time* pada sistem berbasis web yang akan dibuat.

## Perumusan Masalah

Perumusan masalah yang diangkat pada Proyek Akhir ini adalah Bagaimana membuat Sistem Manajemen Pengelolaan Presensi Proyek di PT. Lundin Berbasis Web ?

## Tujuan

Adapun tujuan dari dilakukannya penulisan Proyek Akhir ini adalah mendapatkan sistem manajemen pengelolaan data presensi proyek kapal di PT. Lundin.

## Manfaat

Adapun manfaat yang diperoleh adalah sebagai berikut :

1. Bagi Admin Proyek
   1. Dapat memberikan efektivitas waktu dalam proses pencatatan pekerjaan proyek kapal.
   2. Dapat memberikan data pekerjaan proyek yang akurat.
2. Bagi pegawai produksi
   1. Pencatatan pekerjaan dari pegawai produksi tidak lagi terbatas pada sesi waktu yang diterapkan saat ini.
   2. Pegawai bisa langsung melakukan presensi proyek tanpa menunggu admin proyekyang mendatanginya.
3. Bagi manajer departemen
   1. Dapat memantau laporan presensi proyek secara *realtime.*
   2. Dapat melihat riwayat pekerjaan proyek kapal dengan cepat dan akurat.
4. Bagi proyek kapal
   1. Tidak akan lagi data proyek kapal yang menjadi satu dengan proyek lainnya dalam sesi waktu yang sama.

## Batasan Masalah

Adapun batasan masalah dari proyek akhir ini adalah sebagai berikut:

1. Sistem manajemen presensi proyek berbasis web untuk departemen produksi PT. Lundin digunakan di dalam jaringan *local.*

--*Halaman ini sengaja dikosongkan--*

# BAB 2

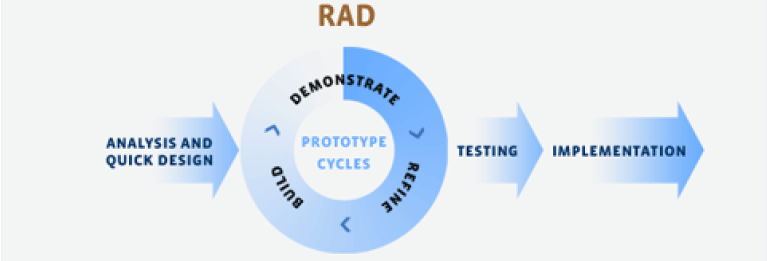
**TINJAUAN PUSTAKA**

## Landasan Teori

### Metode RAD

Metode *RAD* *(Rapid Application Development)* sebagai salah satu *alternative* dari metode *SDLC (System Development Life Cycle).* Metode *SDLC* banyak diterapkan untuk mengatasi keterlambatan yang terjadi apabila menggunakan metode konvensional. Metode *RAD* merupakan metode yang sesuai digunakan untuk menghasilkan sistem perangkat lunak dengan kebutuhan yang mendesak dan waktu yang singkat dalam penyelesaiannya. Metode *RAD* merupakan pendekatan berorientasi objek untuk menghasilkan sebuah sistem dengan sasaran utama mempersingkat waktu pengerjaan aplikasi dan proses agar sesegera mungkin memberdayakan sistem perangkat lunak tersebut secara tepat dan cepat. Metode *RAD (Rapid Application Development)* menggunakan metode *iteratif* (berulang) dalam mengembangkan sistem. Model bekerja dengan sistem dikonstruksikan di awal tahap pengembangan dengan tujuan menetapkan kebutuhan pengguna. (Kosasi & Yuliani, 2015).

Dalam pengembangan sistem informasi normal, memerlukan waktu minimal 180 hari, namun dengan menggunakan metode *RAD*, sistem dapat diselesaikan dalam waktu 30-90 hari. Dapat dilihat gambar alur metode *RAD* pada Gambar 2.1.



Gambar 2. 1 Alur Metode RAD

Model RAD memiliki 3 tahapan sebagai berikut.

1. Rencana Kebutuhan (*Requirement Planning*): *User* dan *analyst* melakukan pertemuan untuk mengidentifikasi tujuan dari sistem dan kebutuhan informasi untuk mencapai tujuan. Pada tahap ini merupakan hal terpenting yaitu adanya keterlibatan dari kedua belah pihak.
2. Proses Desain Sistem (*Design Sistem*): Pada tahap ini keaktifan *user* yang terlibat menentukan untuk mencapai tujuan karena pada proses ini melakukan proses desain dan melakukan perbaikan-perbaikan apabila masih terdapat ketidaksesuaian desain antara *user* dan *analyst*. Seorang *user* dapat langsung memberikan komentar apabila terdapat ketidaksesuaian pada desain, merancang sistem dengan mengacu pada dokumentasi kebutuhan *user* yang dibuat pada tahap sebelumnya. Keluaran dari tahapan ini adalah spesifikasi *software* yang meliputi organisasi sistem secara umum, struktur data dan yang lain.
3. Implementasi (*Implementation*): Tahapan ini adalah tahapan *programmer* yang mengembangkan desain suatu program yang telah disetujui oleh *user* dan *analyst.* Sebelum diaplikasikan pada suatu organisasi terlebih dahulu dilakukan proses pengujian terhadap program tersebut apakah ada kesalahan atau tidak. Pada tahap ini *user* biasa memberikan tanggapan akan sistem yang sudah dibuat serta mendapat persetujuan mengenai sistem tersebut.

Beberapa Keunggulan Model *RAD* :

1. Setiap fungsi *mayor* dapat dimodulkan dalam waktu tertentu kurang dari 3 bulan dan dapat dibicarakan oleh tim *RAD* yang terpisah dan kemudian diintegrasikan sehingga waktunya lebih efisien.
2. *RAD* mengikuti tahapan pengembangan sistem sepeti umumnya, tetapi mempunyai kemampuan untuk menggunakan kembali komponen yang ada (*reusable object*) sehingga pengembang tidak perlu membuat dari awal lagi dan waktu lebih singkat .

Kelemahan Model *RAD* :

1. Proyek yang besar dan berskala, *RAD* memerlukan sumer daya manusia yang memadai untuk menciptakan jumlah tim yang baik.
2. *RAD* menuntut pengembang dan pelanggan memiliki komitmen dalam aktivitas *rapid fire* yang diperlukan untuk melengkapi sebuah sistem dalam waktu yang singkat. Jika komitmen tersebut tidak ada maka proyek *RAD* akan gagal. (Rukmana & Desiyani, 2017)

### *Hypertext Preprocessor* (PHP)

PHP ( singkatan dari *Hypertext Preprocessor*) adalah bahasa pemrograman umum yang dibuat oleh Ramus Ledrorf dan dirilis pertama kali pada tahun 1995. Meskipun sebenarnya PHP dapat digunakan untuk membuat aplikasi berjenis *desktop, Command-Line Interface* (CLI), dan aplikasi *mobile* (berbasis android); tapi pada umumnya PHP lebih banyak digunakan untuk mengembangkan aplikasi berbasis web. Sampai saat ini, banyak sekali situs web dinamis yang dikembangkan menggunakan PHP dibandingkan dengan Bahasa-bahasa lainnya seperti ASP.NET, Java, Ruby, Python, maupun Perl. (Raharjo, 2018)

PHP memiliki sintaks yang sederhana dan berbasis pada Bahasa pemrograman C dan Perl. PHP mudah dipelajari, bahkan oleh pemula sekalipun, PHP memilikipustaka yang lengkap (baik pustaka standar maupun pustaka yang disediakan oleh pihak ketiga) sehingga banyak permasalahan pemrograman yang dapat diselesaikan secara mudah dengan PHP. (Raharjo, 2018)

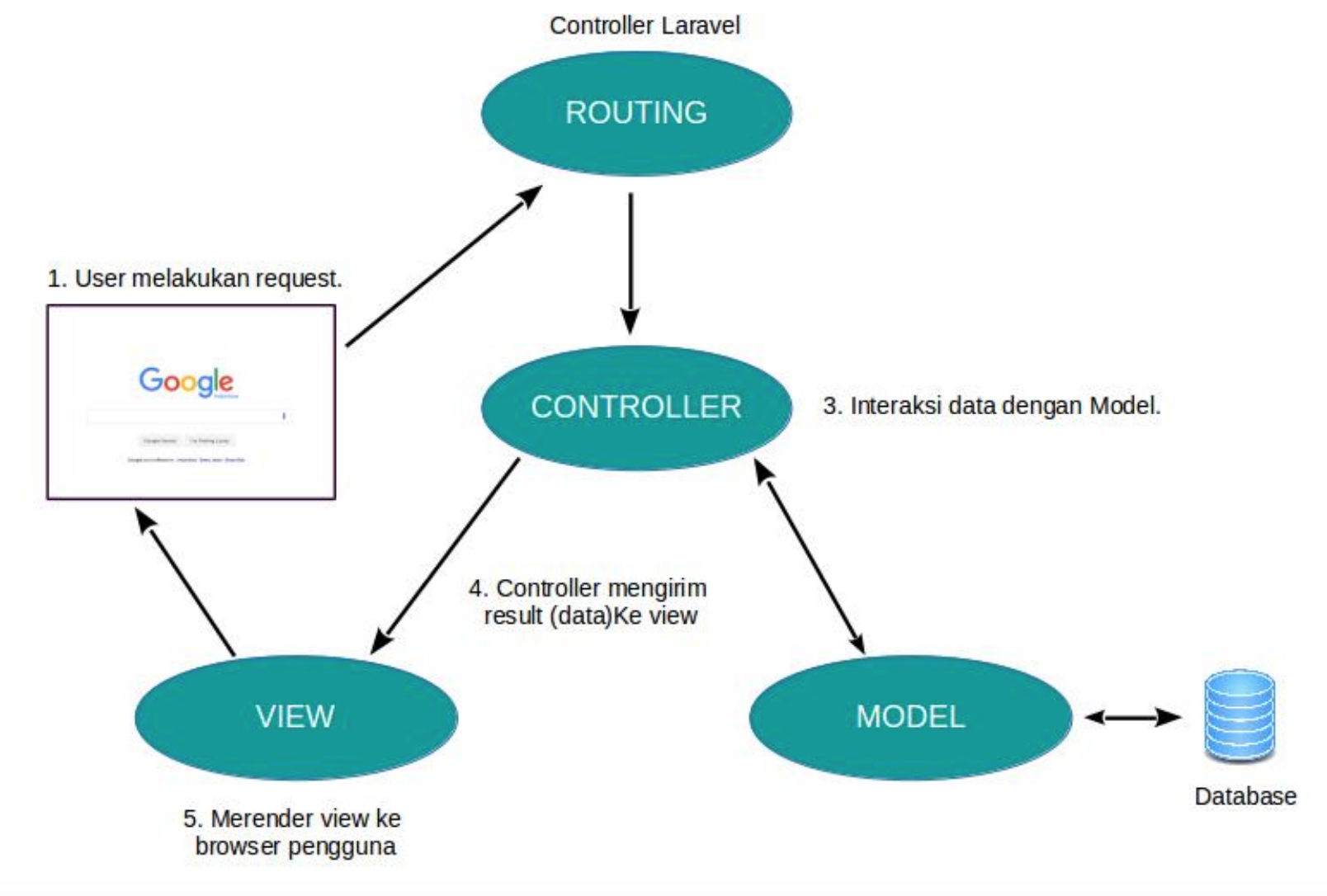
PHP terdapat 3 model penggunaan yaitu Website dan Aplikasi (*Server Side Scripting), Command Line Scripting*, Aplikasi Desktop (GUI). Pada model *Server Side Scripting* merupakan model yang penggunaannya paling umum dibutuhkan 3 hal yaitu PHP, sebuah server web, dan browser web. Untuk *platform* Linux dan MacOs X, dapat menggunakan server web Apache, server web IIS digunakan untuk *platform* Windows. Hal yang perlu dilakukan adalah menulis script PHP, mengupload ke server yang disewa, dan melihat hasilnya pada browser.

Untuk menulis aplikasi pada desktop GUI juga dapat menggunakan PHP. Dan jika menggunakan *Command Line Scripting*, misalnya untuk menulis s*cript* yang membangkitkan gambar secara otomatis secara *offline,* atau memproses file teks yang dikendalikan oleh beberapa argument yang dikirimkan kepada *script,* maka dibutuhkan *Command Line Executable*. Dalam kasus ini, tidak diperlukan server dan browser.

### *Framework* *Laravel*

*Framework laravel* adalah sebuah kerangka kerja *open source* yang diciptakan oleh Taylor Otwell. *Laravel* merupakan *framework bundle*, migrasi dan memiliki *artisan CLI (Command Line Interface)* yang menawarkan seperangkat alat dan arsitektur aplikasi yang menggabungkan banyak fitur terbaik dari kerangka kerja seperti *Codeigniter, Yii, ASP.NET MVC, Ruby on Rails, Sinatra* dan lain-lain. *Laravel* memiliki banyak fitur yang akan meningkatkan kecepatan pengembangan web. (Handika & Ayi Purbasari, 2018)

MVC adalah sebuah pendekatan perangkat lunak yang memisahkan aplikasi logika dari presentasi. MVC memisahkan aplikasi berdasarkan komponen-komponen aplikasi, seperti manipulasi data, *controller* dan *user interface.* Pada pembangunan website menggunakan *framework laravel* perlu mengenal sistematika alur kerja MVC, berikut ilustrasi dari konsep kerja MVC pada gambar 2.2 :



Gambar 2. 2 Alur kerja MVC Pada *Laravel*

Ada 5 konsep arsitektur pada *framework laravel* yang masing-masing mempunyai fungsi diantaranya:

1. *Routes* : berfungsi sebagai pemberi akses pada setiap request sesuai alur yang telah di tentukan. Didalam routes memiliki 4 instruksi standar, yakni: a. *Get* : berfungsi untuk memanggil request.

b. *Put* : berfungsi untuk mengambil data sesuai request.

c. *Post* : berfungsi untuk menambahkan data sesuai request.

d. *Delete* : berfungsi untuk menghapus data sesuai request.

1. *Controller* : adalah bagian yang menjadi penghubung antara *model* dan *view.* *Controller* memiliki perintah-perintah yang berfungsi untuk memproses bagaimana data ditampilkan dari *Model* ke *View* atau sebaliknya. Struktur *controller* pada penulisan kode program di *laravel* yakni:

a*. Index* : Berfungsi untuk menampilkan data keseluruhan.

b. *Create* : Berfungsi untuk memanggil form yang berisikan kolom inputan.

c. *Store :* Berfungsi untuk menyimpan data ke dalam table.

d. *Show :* Berfungsi untuk menampilkan data sesuai *Id.*

e. *Edit :* Memanggil data sesuai *Id* yang berisikan form inputan untuk proses update.

f. *Update :* Berfungsi untuk meng-*update* data pada *table.*

g. *Delete :* Berfungsi untuk menghapus data sesuai *Id*.

1. Model : merupakan sekumpulan data yang memiliki fungsi-fungsi untuk mengelola suatu table pada sebuah database. Struktur pemodelan data pada *laravel* yakni memiliki fungsi yang terdiri dari *table, primaryKey* dan *fillable.* Dimana ketiga fungsi tersebut harus di *protected.* Pada bagian *table* harus diisi dengan nama *table* yang sesuai pada *database,* di bagian *primaryKey* harus diisi sesuai *primary key* pada table tersebut dan pada bagian *fillable* diisi dengan bagian-bagian yang mencakup dalam *table* tersebut.
2. *View* : merupakan *file* yang berisi kode html *(HyperText Markup Language)* yang berfungsi untuk menampilkan suatu data ke dalam *browser*. Format *view* pada *larave*l harus menggunakan istilah *blade,* contohnya seperti: *view.blade.php.*
3. *Migrations :* merupakan proses perancangan suatu *table,* dalam hal ini *migrations* berfungsi sebagai *blueprint database* atau dapat diistilahkan sebagai penyedia sistem kontrol untuk skema *database.*

*Framework laravel* memiliki keunggulan tersendiri yang menjadikannya lebih baik dari pada framework lainnya, berikut ini merupakan kelebihan dari laravel yaitu, *performance* lebih cepat*, reload* data lebih stabil, memiliki keamanan data, menggunakan fitur canggih seperti *blade* menggunakan konsep *HMVC (Hierarchical Model View Controller),* tersedianya *library-library* yang sudah siap untuk digunakan dan adanya fitur pengelolaan *migrations* untuk pembuatan skema *table* pada *database.* (Handika & Ayi Purbasari, 2018)

### Basis Data

Menurut Sucipto (2017) *Database* adalah kumpulan data yang dihubungkan secara bersama-sama, dan gambaran dari data yang dirancang untuk memenuhi kebutuhan informasi dari suatau organisasi. Secara umum *database* dapat diartikan sebagai sebuah tempat pemyimpanan data sebagai pengganti dari sistem konvesional yang berupa dokumen file. Perancangan *database* sendiri terdapat tiga fase utama yaitu perancangan *database* konseptual yang merupakan proses membangun model dari data yang digunakan dalam sebuah organisasi dan tidak tergantung pada petimbangan fisik, perancangan *database logical*, merupakan proses membangun model dari informasi yang digunakan dalam perusahaan berdasarkan model data spesifikasi, dan terbebas dari *DBMS (Databse Management System)* tententu dan pertimbangan fisik lainnya.

Basis data merupakan suatu kesatuan dari kumpulan informasi yang secara utuh disimpan di dalam perangkat keras (komputer) secara sistematis sehingga dapat diolah menggunakan perangkat lunak. Dapat diartikan bahwa basis data adalah kumpulan dari catatan-catatan, atau potongan dari pengetahuan. Sistem basis data merupakan suatu sistem yang menyusun dan mengelola *record-record* menggunakan komputer untuk menyimpan atau merekam serta memelihara data operasional lengkap. Sehingga membantu suatu perusahaan/organisasi mampu menyediakan informasi yang diperlukan pemakai untuk proses mengambil keputusan. terdapat operasi-operasi dasar yang dapat dilakukan dengan *database* antara lain :

### MariaDB

MariaDB adalah sistem manajemen database relasional yang dikembangkan dari MySQL. MariaDB dikembangkan oleh komunitas pengembang yang sebelumnya berkontribusi untuk *database* MySQL. Salah satu alasan pengembang MySQL membangun MariaDB adalah telah diakuisisinya MySQL oleh Oracle sehingga menyebabkan MySQL menjadi produk yang berlisensi *proprietary.* Dengan diakuisisinya MySQL oleh Oracle, maka pengembangan MySQL pun sudah tidak leluasa lagi. Hal ini yang menyebabkan pengembang MySQL sebelumnya mulai membangun MariaDB. MariaDB tetap mempertahankan kompatibilitas dan API layaknya MySQL dulu. Jika di MySQL ada InnoDB maka di MariaDB ada XtraDB yang menjadi mesin penyimpanan baru. Adapun Aria digunakan untuk transaksi *database* transaksional maupun non-transaksional. (Warman & Rizki Ramdaniansyah, 2018)

MariaDB merupakan salah satu database server yang digunakan untuk menyimpan dan memanajemen data. MariaDB tidak jauh berbeda dengan MySQL, karena MariaDB merupakan versi pengembangan terbuka dan mandiri dari MySQL. Sejak diakuisisinya MySQL oleh Oracle pada September 2010, Monty Program sebagai penulis awal kode sumber MySQL memisahkan diri dari pengembangan dan membuat versi yang lebih mandiri yakni MariaDB. Sampai saat ini, sudah banyak yang telah melakukan migrasi dari MySQL ke MariaDB, contohnya saja perusahaan raksasa Google dan juga situs besar seperti Wikipedia. Salah satu kelebihan MariaDB adalah karena performannya yang cukup bagus dan tidak berat serta kompatibel dengan MySQL. MariaDB juga kompatibel dengan berbagai macam platform seperti LINUX, Windows, MacOS, FreeBSD, Solaris. (Warman & Rizki Ramdaniansyah, 2018)

### *Use Case* *Diagram*

*Use case* diagram diharapkan dapat menggambarkan fungsionalitas sebuah sistem. Yang mana menekankan “apa” yang dikerjakan sistem, dan bukan “bagaimana”. Jadi dalam sebuah *use case* itu merepresentasikan sebuah interaksi antara aktor dengan sistem. Dapat diartikan *use case* adalah sebuah pekerjaan seperti *login* ke sistem, membuat sebuah daftar belanja dan lain-lain. Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu. Jadi *use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat.

Kelebihan dari *use case* diagram sendiri dapat membantu mengkomunikasikan rancangan dengan klien, menyusun *requirement* sebuah sistem, dan merancang *test case* untuk semua *feature* yang ada pada sistem. Sebuah *use case* dapat meng-*include* fungsionalitas *use case* lain sebagai bagian dari proses dalam dirinya. Secara umum diasumsikan bahwa *use case* yang di-*include* akan dipanggil setiap kali *use case* yang meng-*include* dieksekusi secara normal. Sebuah *use case* dapat di-*include* oleh lebih dari satu *use case* lain, sehingga duplikasi fungsionalitas dapat dihindari dengan cara menarik keluar fungsionalitas yang *common*. Sebuah *use case* juga dapat meng-*extend* *use case* lain dengan *behaviour*-nya sendiri. Sementara hubungan generalisasi antar *use case* menunjukkan bahwa *use case* yang satu merupakan spesialisasi dari yang lain.

Pada *Use Case* *Diagram* terdapat komponen/simbol-simbol yang ditunjukkan pada Tabel 2.1. (Yusmiarti, 2016).

Tabel 2. 1 *Use Case Diagram*

|  |  |  |
| --- | --- | --- |
| **No.** | **Simbol** | **Deskripsi** |
| 1. |  | Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja diawal frase nama *.* |
| 2. | Aktor/*actor* | Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama aktor. |
| 3. | Asosiasi/*association* | Komunikasi antara aktor dan yang berpartisilpasi pada *use case* atau *use case* memiliki interaksi dengan aktor. |
| 4. | Ekstensi/*extend* | Relasi tambahan kesebuah *use case* dimana *use case* yang ditambahkan dapat berdiri sendiri walau tanpa *use case* tambahan itu, mirip dengan prinsip inheritance pada pemrograman berorientasi objek, biasanya *use case* tambahan memiliki nama depan yang sama dengan *use case* yang ditambahkan. |
| 5. | Generalisasi/*generalization* | Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah *use case* dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya. |
| 66. | Menggunakan / *include* | Relasi *use case* tambahan ke sebuah *use case* dimana *use case* yang ditambahkan memerlukan *use case* untuk menjalankan fungsinya atau sebagai syarat dijalankan *use case* ini. |

### Diagram Aktivitas

Diagram aktivitas (*Activity diagram*) menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang. Struktur diagram mirip seperti *flowchart* atau *Data Flow Diagram* pada perancangan terstruktur. *Activity diagram* berfungsi untuk menggambarkan *workflow* / aliran kerja dari suatu proses bisnis. Suatu aliran kerja bisa saja dituangkan dalam bentuk narasi / teks, akan tetapi jika aliran kerjanya sudah kompleks maka kita akan kesulitan untuk membayangkan bagaimana proses itu terjadi. (Harisantyo, et al., 2015).

Menurut Rembulan (2015) Sebuah aktivitas dapat direalisasikan oleh satu use case atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara use case menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas.

Pada *Activity* diagram terdapat komponen/simbol-simbol yang ditunjukkan pada Tabel 2.2. (Yusmiarti, 2016)

Tabel 2. 2 *Activity Diagram*

|  |  |  |
| --- | --- | --- |
| No. | Simbol | Deskripsi |
| 1. | Status awal | Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal |
| 2. | Aktivitas  aktivitas | Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja. |
| 3. | Percabangan / *Decision* | Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu |
| 4. | Penggabungan/ JoinAsosiasi | Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu. |
| 5. | Status akhir | Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir. |
| 6. | Swimlane | Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi. |

### *Black* *Box Testing*

*Black Box Testing* merupakan teknik pengujian perangkat lunak yang berfokus pada spesifikasi fungsional dari perangkat lunak. Keuntungan dalam penggunaan metode *Blackbox Testing* adalah penguji tidak perlu memiliki pengetahuan tentang bahasa pemrograman tertentu, pengujian yang dilakukan dari sudut pandang pengguna, ini membantu untuk mengungkapkan ambiguitas atau inkonsistensi dalam spesifikasi persyaratan, dan *programmer* dan *tester* keduanya saling bergantung sama lain. Sedangkan kekurangan dari metode ini adalah uji kasus sulit didesain tanpa spesifikasi yang jelas, kemungkinan memiliki pengulangan tes yang sudah dilakukan oleh *programmer*, dan beberapa bagian *back end* tidak diuji sama sekali. (Jaya, 2018)

*Black Box Testing* bukanlah solusi alternatif dari *White Box Testing* tapi lebih merupakan pelengkap untuk menguji hal-hal yang tidak dicakup oleh *White Box Testing*. (Mustaqbal, et al., 2015) *Black Box Testing* cenderung untuk menemukan hal-hal berikut:

1. Fungsi yang tidak benar atau tidak ada.
2. Kesalahan antarmuka (*interface errors*).
3. Kesalahan pada struktur data dan akses basis data.
4. Kesalahan performansi (*performance errors*).
5. Kesalahan inisialisasi dan terminasi.

Pengujian didesain untuk menjawab pertanyaan - pertanyaan berikut:

1. Bagaimana fungsi-fungsi diuji agar dapat dinyatakan valid?
2. Masukan seperti apa yang dapat menjadi bahan kasus uji yang baik?
3. Apakah sistem sensitif pada masukan tertentu?
4. Bagaimana sekumpulan data dapat diisolasi?
5. Berapa banyak rata-rata data dan jumlah data yang dapat ditangani sistem?
6. Efek apa yang dapat membuat kombinasi data ditangani spesifik pada operasi sistem?

Saat ini terdapat banyak metoda atau teknik untuk melaksanakan *Black Box Testing*, antara lain:

1. *Equivalence Partitioning*
2. *Boundary Value Analysis/Limit Testing*
3. *Comparison Testing*
4. *Sample Testing*
5. *Robustness Testing*
6. *Behavior Testing*
7. *Requirement Testing*
8. *Performance Testing*
9. Uji Ketahanan (*Endurance Testing*)
10. Uji Sebab-Akibat (*Cause-Effect Relationship Testing*)

## Penelitian Terdahulu

Dalam merancang Sistem Manajemen Pengelolaan Presensi Proyek di PT. Lundin Berbasis Web, Sistem ini akan dibuat dengan memanfaatkan *QR Code* sebagai inputan autitentifikasi pegawai dan pekerjaan proyek. Penulis menggunakan referensi dari penelitian dan sistem sejenis yang pernah dibuat sebelumnya.

Penelitian yang terkait dari pembuatan proyek akhir dijelaskan Tabel 2.3:

Tabel 2. 3 Penelitian Terdahulu

|  |  |  |  |
| --- | --- | --- | --- |
| Unsur Perbandingan | (Pamukti, 2019) | (Putri & Supriyono, 2019) | (Nugraha, 2018) |
| Judul : | Penggunaan *QR* *Code* dalam Presensi Berbasis *Android* | Rancang Bangun Sistem Presensi Berbasis QR Code Menggunakan Framework Codeigniter (Studi Kasus Kehadiran Asisten Praktikum) | Sistem Informasi Manajemen Presensi Kehadiran Menggunakan Qr Code Berbasis Web Dan Sms Gateway Di Smk Muhammadiyah 2 Sukoharjo |
| Fitur : | 1. Pemindaian *QR Code* untuk presensi mahasiswa menggunakan *android* | 1. *Scanning* *QR Code*dari kartu tiap asisten pada *webcam*di *Personal Computer*(PC). | * 1. SMS gateway untuk melaporkan kergiatan siswa, guru dan karyawan sudah berjalan sesuai rancangan   2. Pemindaian *QR Code* dengan *webcam* |
| Teknologi : | *Android* | Web | Web |

# BAB 3

**METODE PENELITIAN**

## Tempat dan Waktu Penelitian

Tempat penelitian Proyek Akhir ini akan dilakukan di PT. Lundin (*North Sea Boats*), Jalan Lundin No 1, Sukowidi, Kecamatan Kalipuro, Kabupaten Banyuwangi, Jawa Timur. Proyek Akhir ini dikerjakan secara urut yang ditunjukkan pada Tabel 3.1.

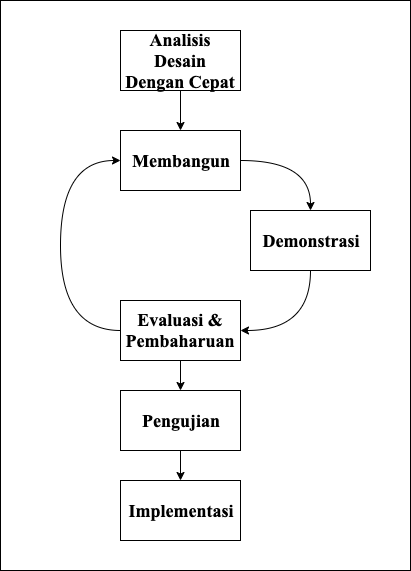
Tabel 3. 1 Jadwal Penelitian

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| No | Kegiatan | 2019 | | | | | | | | | | | | | | | | | | | | | | | | |
| I | | | | | II | | | | | III | | | | | IV | | | | | V | | | | |
| 1 | Studi literatur dan pemahaman alur sistem |  |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |
| 2 | Analisa kebutuhan dan perancangan sistem |  |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |
| 3 | Desain Sistem |  |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |
| 4 | Pembuatan sistem |  |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |
| 5 | Pengujian Sistem |  |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |
| 6 | Implementasi Sistem |  |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |
| 7 | Laporan Proyek Akhir |  |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |

## Metode Pengembangan Aplikasi

Metode yang digunakan dalam pengerjaan proyek akhir ini yaitu metode *Rapid Application Development* *(RAD).* Metode *RAD* menggunakan metode berulang dalam mengembangkan sistem. *RAD* menekankan pada siklus pembangunan pendek, singkat, dan cepat. Jadi metode *RAD* sesuai untuk menghasilkan sistem perangkat lunak dengan kebutuhan yang mendesak dan menyingkat waktu dalam penyelesaiannya. *RAD* juga mempunyai kemampuan untuk menggunakan kembali komponen yang ada (*reusable object*) sehingga pengembang tidak perlu membuat dari awal lagi dan waktu lebih singkat.

Secara umum, bagaimanapun, model ini dianggap memiliki lima tahap yang berbeda seperti yang ditunjukkan pada Gambar 3.1.



Gambar 3. 1 Metode RAD

### Perancangan

Hasil dari perencanaan sistem, menggunakan sistem berbasis *web* yang dapat diakses dan sebagai media *monitoring* anggaran usulan pekerjaan yang *real time* dan *up to date*, dengan menggunakan sistem manajemen pada pengguna membedakan hak akses.

### Analisis Kebutuhan

Analisa kebutuhan merupakan tahapan pengumpulan kebutuhan yang dilakukan secara intensif untuk menspesifikasikan kebutuhan perangkat lunak agar dapat dipahami dan menyesuaikan dengan perangkat lunak yang dibutuhkan oleh pengguna. Pengumpulan data dan referensi dilakukan dengan melakukan studi kakus metode presensi proyek kapal di PT. Lundin.

### Desain Sistem

Tahapan dilakukan dengan perancangan sistem dari permasalahan yang ada dengan menggunakan perangkat pemodelan sistem seperti diagram *use case*, diagram hubungan entitas (*entity relationship diagram*), serta perancangan struktur sistem sehingga akan lebih memudahkan dalam pembuatannya.

### Implementasi Desain

Pada tahap implementasi desain, dilakukan implementasi hasil dari desain sistem dengan menggunakan bahasa pemrograman PHP dengan *framework Laravel* dalam perancangan sistem manajemen berbasis web dan *MariaDB* untuk mengelola *database*.

### Pengujian Sistem

Setelah sistem terbentuk, dilakukan pengujian untuk mengetahui kekurangan dan kelemahan sistem yang kemudian dilakukan pengkajian ulang dan perbaikan terhadap aplikasi menjadi lebih baik dan sempurna. Pengujian dilakukan juga untuk mengetahui seberapa kuat aplikasi yang telah dibangun. Pengujan aplikasi dilakukan dengan metode *black box testing* yang merupakan pengujian yang melihat aplikasi hanya dari luar saja tanpa tau yang terjadi di dalam aplikasi.

### Implementasi Sistem

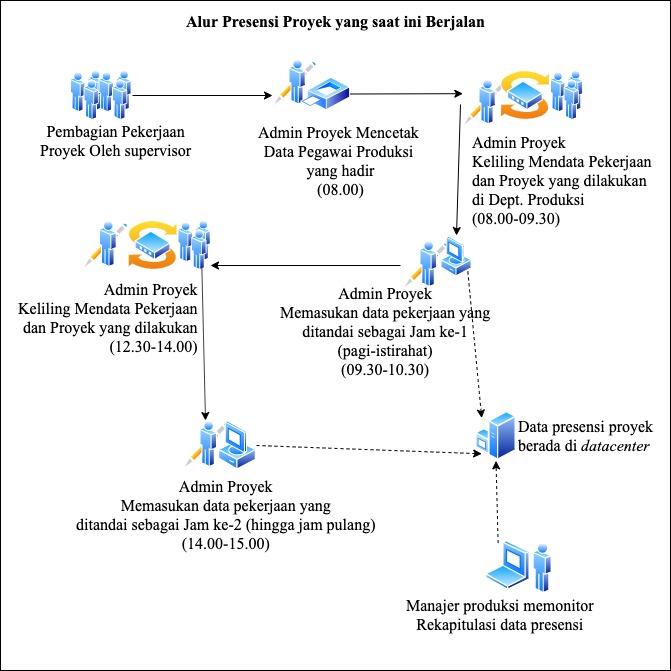
Implementasi merupakan suatu proses untuk menerapkan sistem manajemen presensi proyek kapal di PT. Lundin yang telah dibuat. Pada tahap ini, implementasi dilakukan dengan beberapa proses, yaitu:

1. Memperkenalkan aplikasi kepada departemen produksi.
2. Melatih alur presensi sistem kepada pegawai produksi dan pengelolaan data kepada *Admin Proyek.*

## Gambaran Umum Sistem

Gambaran umum diperlukan dalam melakukan pembuatan ataupun pengembangan suatu sistem. Pada gambaran umum sistem terdapat dua pokok bahasan yang dijelaskan yaitu, gambaran umum sistem yang berjalan dan gambaran umum sistem yang diusulkan. Tujuan dari pembahasan gambaran umum sistem ini yaitu untuk mengetahui pembaharuan atau pengembangan yang akan kita lakukan terhadap sistem yang telah berjalan sebelumnya.

### Sistem yang Berjalan



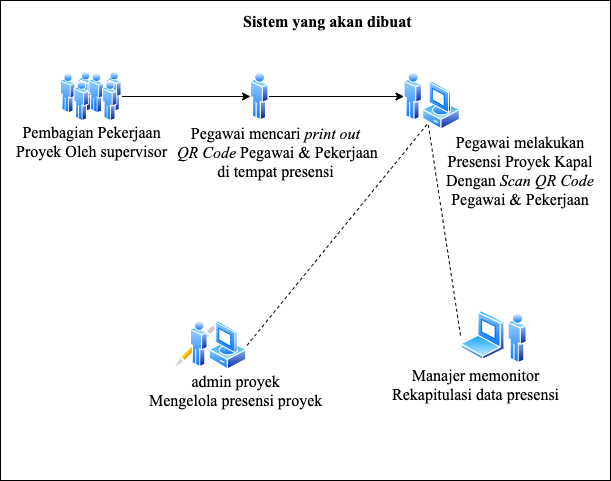
Gambar 3. 2 Alur Presensi Proyek Kapal saat ini

Gambar 3.2. merupakan alur presensi proyek kapal di PT. Lundin yang saat ini berlaku. Supervisor membagikan perkerjaan proyek pada pegawai dalam kelompok kerjanya, ketika seorang pegawai telah menyelesaikan pekerjaan proyek, maka pegawai tersebut akan meminta pekerjaan proyek lainnya kepada supervisor. Sistem pencatatan presensi proyek yang berjalan saat ini dilakukan secara manual.

Admin proyek mencetak data pegawai untuk pencatatan aktivitas kerja disetiap proyek. Dalam sehari biasanya setiap pegawai mengerjakan 2 hingga 3 pekerjaan proyek yang berbeda. Pencatatan presensi kerja proyek oleh admin dilakukan dalam 2 sesi. Sesi pertama pada pagi hari hingga sebelum jam istirahat dan sesi kedua setelah istirahat hingga jam pulang. Pencatatan presensi proyek dilakukan dengan berkeliling dan menghabiskan waktu yang lumayan lama, berkisar antara 45 menit hingga 1 jam. Lama waktu yang dibutuhkan tersebut dikarenakan jumlah pegawai produksi yang lebih dari 125 orang dan berada di beberapa lokasi yang tidak berdekatan.

Admin proyek mencatat pekerjaan pegawai berdasarkan proyek dan deskripsi proyek yang disampaikan oleh setiap pegawai. Data presensi proyek tersebut akan dimasukan ke *form* dalam bentuk *excel* setiap selesai berkeliling disetiap sesi 1 dan 2. Data yang dimasukan diantaranya adalah sesi kerja, identitas pegawai, proyek dan pekerjaan proyek. Lama proses *input* data berlangsung sekitar 1 jam, hal tersebut terjadi karena admin proyekharus mengetik manual data kedalam *form*. Manajer departemen akan melihat data presensi proyek setelah data tersebut selesai di-*input* oleh admin proyek. Data presensi ini biasanya akan diolah kembali oleh manajer departemen untuk mendapatkan data rekapitulasi lama pengerjaan setiap proyek.

### Sistem yang Diusulkan



Gambar 3. 3 Sistem yang akan dibuat

Gambar 3.3 menunjukkan sistem presensi proyek yang diusulkan. Sistem ini memanfaatkan komputer sebagai media presensi proyek oleh pegawai produksi secara *self-service*. Pegawai departemen produksi melakukan presensi proyek dengan memindai kartu *QR Code* pegawai yang berisi kode dari masing-masing pegawai, lalu memindai kartu *QR Code* perkerjaan proyek di komputer yang disediakan. Komputer akan ditempatkan sesuai dengan tempat bekerja pegawai atau di titik tertentu yang berdekatan dengan area pekerja. Admin proyek mengelola presensi proyek. Manajer departemen dapat secara *real-time* melihat data rekapitulasi lama pengerjaan proyek dan melihat detail pengerjaannya.

## Spesifikasi Sistem

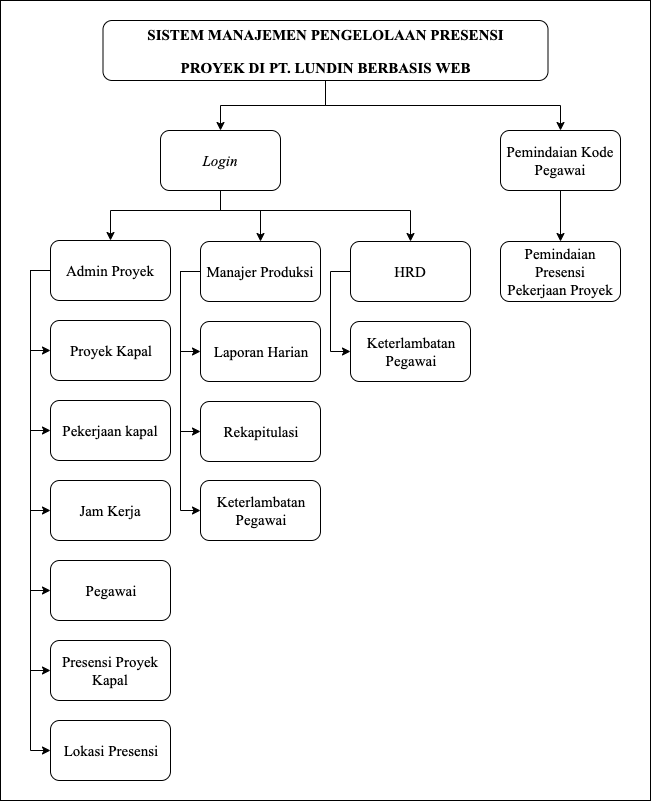
### Aturan Sistem

Sistem Manajemen Pengelolaan Presensi Proyek di PT. Lundin Berbasis Web memiliki beberapa aturan agar sistem dapat berjalan sesuai dengan kebutuhan pengguna. Dibuatnya aturan sistem agar dapat membatasi dan membedakan kebutuhan pengguna yang berbeda. Aturan sistem ditunjukkan pada Tabel 3.2

Tabel 3. 2 Aturan Sistem

|  |  |
| --- | --- |
| *User* | Terdapat 4 *user*  pada sistem manajemen presensi proyek ini, yakni : manajer departemen, admin proyek, HRD dan pegawai. |
| *Login* | Proses login dapat dilakukan oleh manajer departemen dan admin proyek. |
| *Scan QR Code ID* Pegawai | *Scan QR Code ID* pegawai bertujuan untuk mengidentifikasi pegawai yang akan memasukan pekerjaan proyek kapal. |
| Pengelolaan Data Presensi | Akses pengelolaan data presensi proyek kapal diberikan kepada admin proyek. Pengelolaan data berupa perubahan dan penambahan data proyek,waktu pekerjaan, lokasi proyek serta data presensi proyek. |
| Data Rekapitulasi Proyek | Manajer departemen memiliki akses untuk melihat rekapitulasi berdasarkan nama proyek, jenis pekerjaan dan nama pegawai. |
| Laporan Harian | Laporan harian adalah data presensi proyek setiap harinya yang dapat diakses secara daring. |
| Data Keterlambatan Pegawai | Data keterlambatan pegawai dapat diakses oleh manajer departemen dan HRD. Data ini menampilkan pegawai yang terlambat melakukan presensi proyek beserta waktu keterlambatannya. |

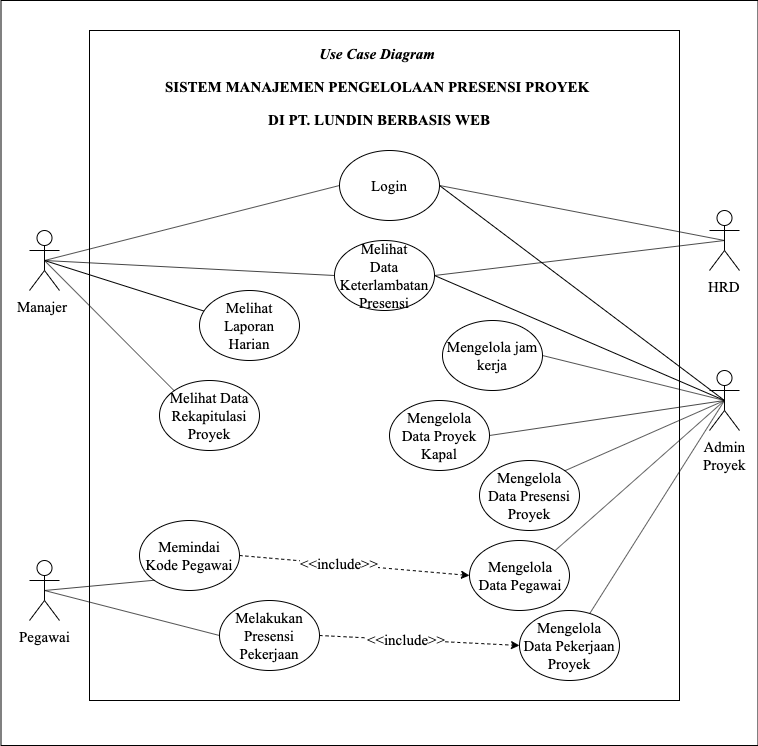
### Struktur Menu



Gambar 3. 4 Struktur Menu

Gambar 3.4 menunjukkan struktur menu pada Sistem Manajemen Pengelolaan Presensi Proyek di PT. Lundin Berbasis Web. Penggunaan sistem mengharuskan manajer departemen dan admin proyek melakukan login karena menu-menu dapat diakses setelah pengguna tersebut melakukan login. Beberapa menu sisi admin proyekadalah proyek kapal, pekerjaan kapal, jam kerja, pegawai dan presensi proyek kapal. Menu yang terdapat pada manajer departemen adalah laporan harian, rekapitulasi data dan data keterlambatan presensi pegawai. Halaman muka untuk HRD adalah data keterlambatan presensi pegawai. Halaman muka untuk pegawai produksi adalah halaman presensi proyek berupa halaman pemindaian *QR Code ID* Pegawai dan halaman pemindaian *QR Code* pekerjaan proyek.

### *Use Case Diagram*



Gambar 3. 5 *Use case diagram*

Pada Gambar 3.5 ditunjukkan pemodelan *Use Case* Sistem Manajemen Pengelolaan Presensi Proyek di PT. Lundin Berbasis Web. Terdapat 3 aktor dalam aplikasi ini, yaitu manajer departemen, admin proyek*,* dan pegawai produksi. Penjelasan tiap aktor ditunjukkan pada Tabel 3.3, serta penjelasan dari setiap use case ditunjukkan pada 3.4, Tabel 3.5, Tabel 3.6, Tabel 3.7, Tabel 3.8, Tabel 3.9, Tabel 3.10, Tabel 3.11, Tabel 3.12, Tabel 3.13, Tabel 3.14.

Tabel 3. 3 Aktor *Use Case Diagram*

|  |  |  |
| --- | --- | --- |
| No | Aktor | Deskripsi |
| 1 | Manajer | Merupakan seorang kepala di departemen produksi. |
| 2 | Admin Proyek | Admin Proyek memiliki tugas untuk mengelola presensi proyek kapal di departemen produksi. |
| 3 | Pegawai | Pegawai merupakan semua pegawai di departemen produksi kecuali manajer dan admin proyek. |
| 4 | HRD | HRD merupakan pengguna sistem diluar departemen produksi. HRD memiliki tugas untuk memantau pekerja. |

Tabel 3. 4 Keterangan *Use Case Login*

|  |  |
| --- | --- |
| *Use Case Name* | *Login* |
| *Actor* | Manajer, Admin proyek, HRD |
| *Description* | Validasi untuk masuk ke sistem |
| *Normal Course* | 1. Sistem menampilkan *form login* 2. Memasukkan *Username* dan *Password* yang sudah terdaftar pada sistem |
| *Alternative Course* | *Login* tidak berhasil karena *username* dan *password* tidak terdaftar pada *database* sistem |
| *Pre-Condition* | *Login* |
| *Post-Condition* | - |

Tabel 3. 5 Keterangan *Use Case* Melihat Data Rekaputulasi Proyek

|  |  |
| --- | --- |
| *Use Case Name* | Melihat Data Rekapitulasi Proyek |
| *Actor* | Manajer |
| *Description* | Melihat data rekapitulasi proyek kapal beserta riwayat pekerjaan setiap kapal dan pegawai. Data rekapitulasi dapat difilter berdasarkan nama proyek, jenis pekerjaan dan nama pegawai. |
| *Normal Course* | Melihat rekapitulasi proyek kapal beserta riwayat pekerjaan setiap kapal dan pegawai |
| *Alternative Course* | - |
| *Pre-Condition* | *Login* sebagai manajer |
| *Post-Condition* | - |

Tabel 3. 6 Melihat Laporan Harian

|  |  |
| --- | --- |
| *Use Case Name* | Melihat Laporan Harian |
| *Actor* | Manajer |
| *Description* | Melihat data laporan harian berdasarkan presensi pekerja secara *real-time* |
| *Normal Course* | Melihat data laporan harian berdasarkan presensi pekerja secara *real-time* |
| *Alternative Course* | - |
| *Pre-Condition* | *Login* sebagai manajer |
| *Post-Condition* | - |

Tabel 3. 7 Melihat Data Keterlambatan Presensi

|  |  |
| --- | --- |
| *Use Case Name* | Melihat Data Keterlambatan Presensi |
| *Actor* | Manajer, HRD |
| *Description* | Melihat data keterlambatan presensi. Data ditampilkan berdasarkan pegawai. |
| *Normal Course* | Melihat data keterlambatan presensi. |
| *Alternative Course* | - |
| *Pre-Condition* | *Login* sebagai manajer |
| *Post-Condition* | - |

Tabel 3. 8 Keterangan *Use Case* Memindai Kode Pegawai

|  |  |
| --- | --- |
| *Use Case Name* | Memindai Kode Pegawai |
| *Actor* | Pegawai |
| *Description* | Memindai kode pegawai yang berupa kartu *QR Code* |
| *Normal Course* | Melakukan *Scan QR Code* pegawai |
| *Alternative Course* | *Scanning* gagal apabila *QR Code* tidak terdaftar sebagai nomor pegawai |
| *Pre-Condition* | Admin proyek mendaftarkan pegawai |
| *Post-Condition* | Menampilkan data pegawai dan halaman *Scan Code* pekerjaan |

Tabel 3. 9 Keterangan *Use Case* Melakukan Presensi Pekerjaan

|  |  |
| --- | --- |
| *Use Case Name* | Melakukan Presensi Pekerjaan |
| *Actor* | Pegawai |
| *Description* | Memasukan id pekerjaan proyek kapal dengan media *QR Code* yang di *scan* |
| *Normal Course* | Melakukan *Scan QR Code* pekerjaan |
| *Alternative Course* | *Scanning* gagal apabila *QR Code* tidak terdaftar sebagai id pekerjaan proyek yang aktif |
| *Pre-Condition* | Masuk sebagai pegawai dengan me-*Scan QR Code,* Admin proyek mendaftarkanpekerjaan proyek |
| *Post-Condition* | Presensi proyek berhasil disimpan |

Tabel 3. 10 Keterangan *Use Case* Mengelola Jam kerja

|  |  |
| --- | --- |
| *Use Case Name* | Mengelola Jam kerja |
| *Actor* | Admin Proyek |
| *Description* | Mengelola jadwal jam kerja pegawai |
| *Normal Course* | Melihat, menghapus, menambahkan, mengubah jadwal jam kerja pegawai. |
| *Alternative Course* | - |
| *Pre-Condition* | *Login* sebagaiAdmin Proyek |
| *Post-Condition* | Data berhasil diubah atau disimpan dan menjadi aturan dalam batas jam kerja pegawai. |

Tabel 3. 11 Keterangan *Use Case* Mengelola Data Proyek Kapal

|  |  |
| --- | --- |
| *Use Case Name* | Mengelola Data Proyek Kapal |
| *Actor* | Admin Proyek |
| *Description* | Mengelola data proyek kapal |
| *Normal Course* | Melihat, menambahkan dan mengubah proyek kapal |
| *Alternative Course* | - |
| *Pre-Condition* | *Login* sebagaiAdmin Proyek |
| *Post-Condition* | Data berhasil diubah atau disimpan |

Tabel 3. 12 Keterangan *Use Case* Mengelola Data Presensi Proyek Kapal

|  |  |
| --- | --- |
| *Use Case Name* | Mengelola Data Presensi Proyek Kapal |
| *Actor* | Admin Proyek |
| *Description* | Mengelola data presensi proyek kapal |
| *Normal Course* | Melihat, mengubah, menambahkan, menghapus presensi proyek kapal. |
| *Alternative Course* | - |
| *Pre-Condition* | *Login* sebagaiAdmin Proyek |
| *Post-Condition* | Perubahan data berhasil disimpan |

Tabel 3. 13 Keterangan *Use Case* Mengelola Data Pegawai

|  |  |
| --- | --- |
| *Use Case Name* | Mengelola Data Pegawai |
| *Actor* | Admin Proyek |
| *Description* | Pengelolaan data pegawai |
| *Normal Course* | Melihat, menambahkan, mengubah, menghapus daftar pegawai, serta mencetak *QR Code* Pegawai. |
| *Alternative Course* | - |
| *Pre-Condition* | *Login* sebagai Admin Proyek |
| *Post-Condition* | Perubahan data berhasil disimpan atau *QR Code* berhasil dicetak. |

Tabel 3. 14 Keterangan *Use Case* Mengelola Data Pekerjaan Proyek

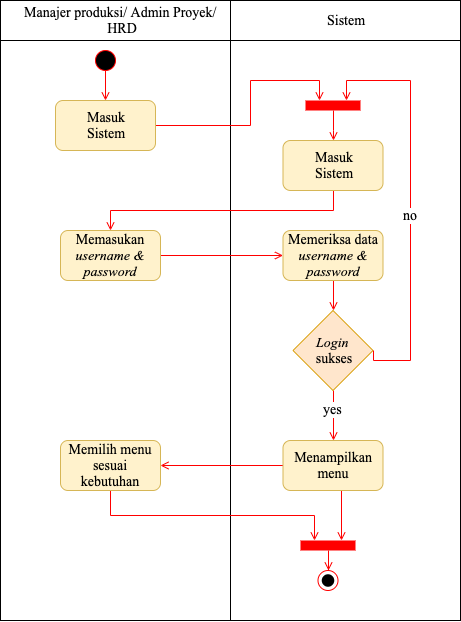
|  |  |
| --- | --- |
| *Use Case Name* | Mengelola Data Pekerjaan Proyek |
| *Actor* | Admin Proyek |
| *Description* | Pengelolaan Data Pekerjaan Proyek |
| *Normal Course* | Melihat, menambahkan, mengubah, menghapus daftar proyek, serta mencetak *QR Code* Pekerjaan. |
| *Alternative Course* |  |
| *Pre-Condition* | *Login* sebagai Admin Proyekdan meng-*input* proyek |
| *Post-Condition* | Perubahan data berhasil disimpan atau *QR Code* berhasil dicetak |

### *Activity Diagram*

Beberapa alur kerja pada sistem dapat digambarkan seperti dibawah ini :

1. *Activity diagram Login*

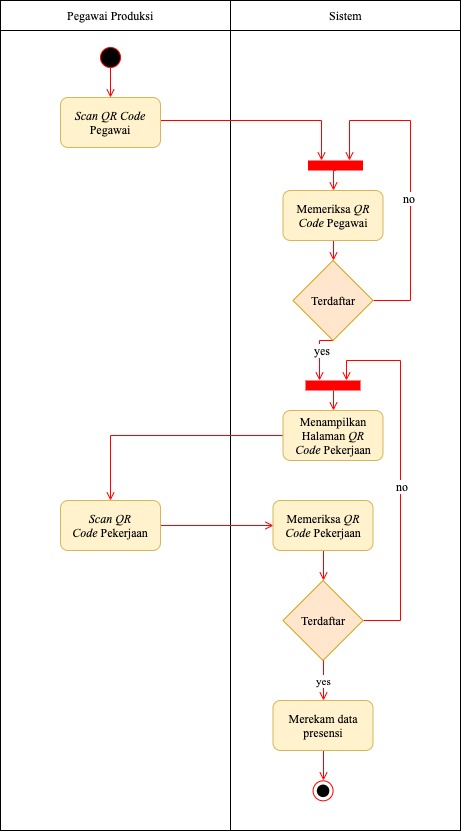
*Form login* digunakan untuk masuk pada halaman sesuai hak akses. Terdapat dua *user* pada sistem presensi yakni manajer departemen, admin proyek dan HRD.



Gambar 3. 6 *Activity Diagram Login*

Gambar 3.6 menunjukkan proses login aplikasi dengan cara:

1. Memasukkan *username* dan *password*.
2. Ketika *login* berhasil maka akan menampilkan pesan berhasil login dan aplikasi akan mengarahkan ke tampilan beranda sistem presensi proyek.
3. Apabila data *username* dan *password* tidak benar, maka akan menampilkan pesan data *username* dan *password* salah dan tetap berada pada halaman *login.*
4. Proses login selesai.
5. *Activity diagram* Presensi Proyek



Gambar 3. 7 *Activity Diagram* Presensi Proyek

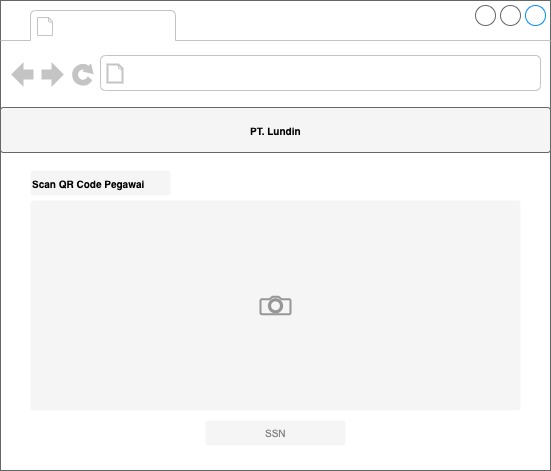
Gambar 3.7 menunjukkan proses pemindaian *QR Code* dari pegawai produksi melalui sistem dengan cara :

1. Melakukan *Scan QR Code* pegawai.
2. Apabila *scan QR Code* pegawai gagal, maka akan memberikan pesan *QR Code* pegawai salah dan tetap berada di halaman tersebut.
3. Ketika *Scan QR Code* berhasil maka akan menampilkan halaman selanjutnya yakni data diri pegawai dan instruksi *scan QR Code* pekerjaan.
4. Apabila *scan QR Code* pekerjaan gagal, maka akan memberikan pesan *QR Code* pkerjaan salah dan tetap berada di halaman tersebut.
5. Ketika *Scan QR Code* pekerjaan berhasil maka data presensi pegawai terkirim dan tersimpan.

### *Mockup* Sistem

*Mockup* sistem manajemen presensi yang akan dibuat memiliki 3 tampilan berbeda untuk setiap level pengguna, berikut desain halaman dari setiap tampilan yang akan ditampilkan pada gambar 3.8 sampai gambar 3.23 :

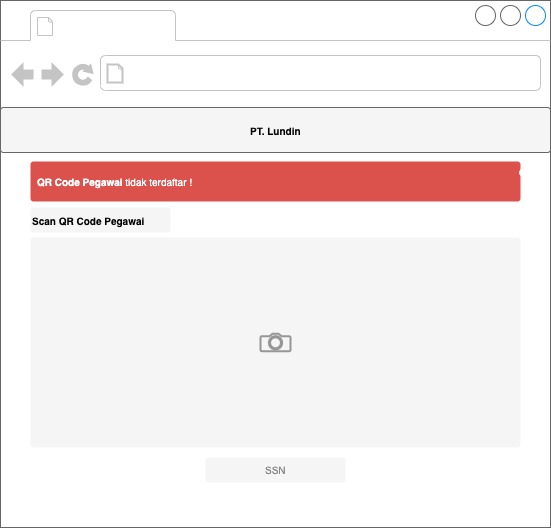
* 1. Halaman Pemindaian *QR Code* Pegawai



Gambar 3. 8 Halaman *Scan QR Code* Pegawai

Pada Gambar 3.8 merupakan halaman *scan QR Code* oleh pegawai produksi. Pegawai memindai *QR Code* yang disiapkan oleh *Admin Proyek.*

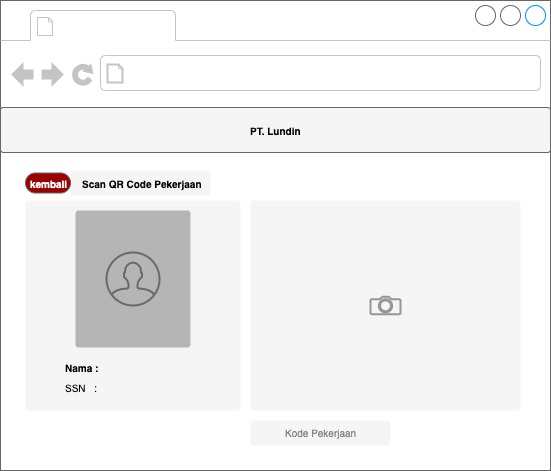
* 1. Halaman Pesan Gagal *Scan QR Code* Pegawai



Gambar 3. 9 Pesan Gagal *Scan QR Code* Pegawai

Pada Gambar 3.9 merupakan pesan gagal *scan QR Code* oleh pegawai produksi.

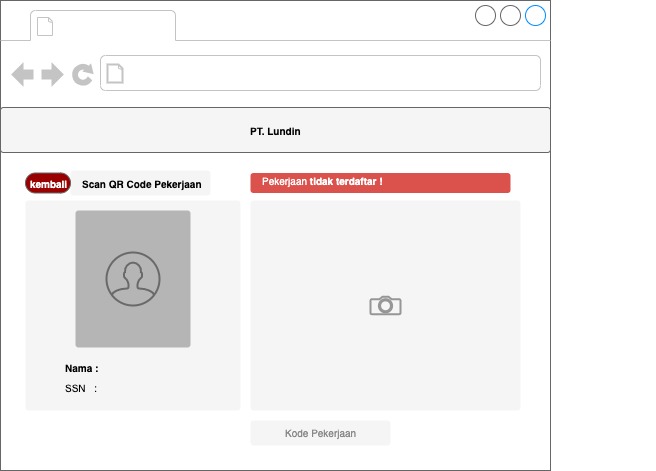
1. Halaman *Scan QR Code* Pekerjaan pada Pegawai



Gambar 3. 10 Halaman *Scan QR Code* Pekerjaan pada Pegawai

Pada Gambar 3.10 merupakan halaman yang muncul setelah sukses melakukan pemindaian kode pegawai. Pada halaman ini Pegawai diminta untuk melakukan pemindaian *QR Code* pekerjaa.. Di halaman ini terdapat data pegawai seperti foto, nama dan SSN pegawai. Pegawai memindai *QR Code* pekerjaan yang disiapkan oleh Admin Proyek.

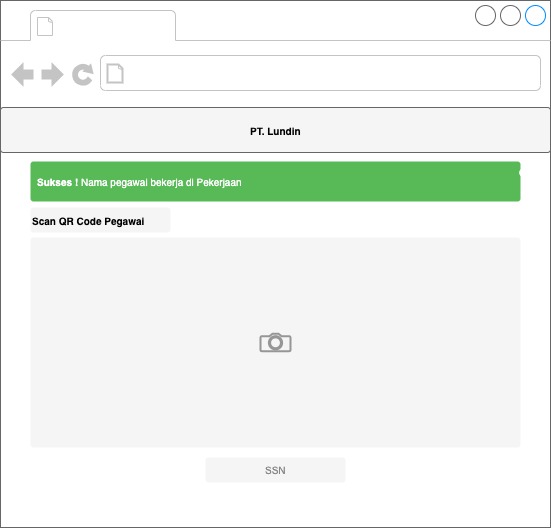
1. Halaman Gagal *Scan QR Code* Pekerjaan



Gambar 3. 11 Halaman Gagal *Scan QR Code* Pekerjaan

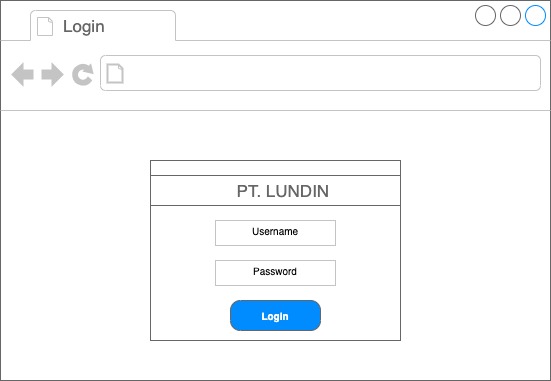
Pada Gambar 3.11 merupakan pesan gagal *scan QR Code* pekerjaan oleh pegawai produksi.

1. Pesan Sukses Presensi Proyek



Gambar 3. 12 Pesan Sukses Presensi Proyek

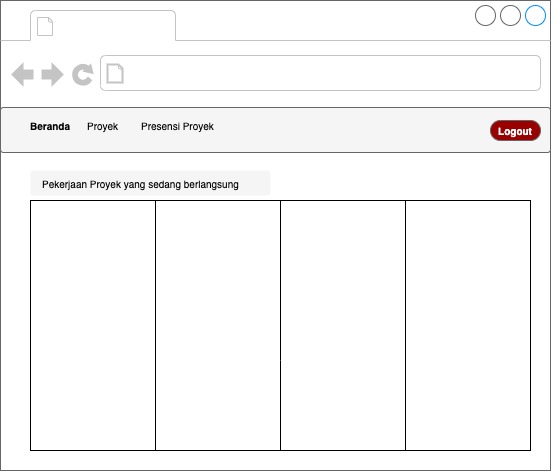
1. Halaman *Login*



Gambar 3. 13 Halaman *Login*

Pada Gambar 3.13 merupakan tampilan utama saat pengguna, yakni manajer departemen, admin proyek dan HRDmembuka sistem. Pengguna harus melakukan proses *login* terlebih dahulu untuk mengakses fitur sistem.

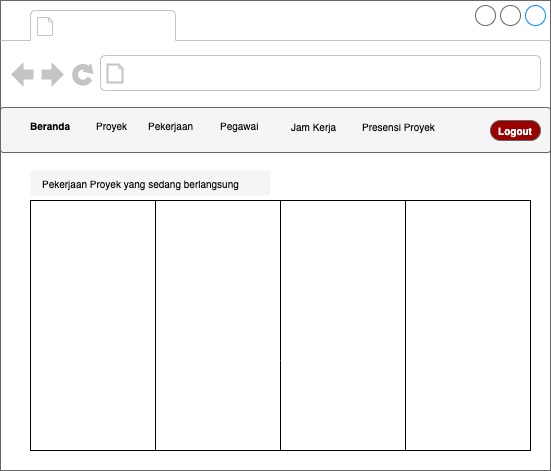
1. Beranda Manajer departemen



Gambar 3. 14 Beranda Manajer departemen

Pada Gambar 3.14 merupakan tampilan utama saat manajer departemen berhasil *login* dan membuka sistem. Di halaman ini terdapat tabel yang menampilkan pekerjaan proyek yang sedang berlangsung.

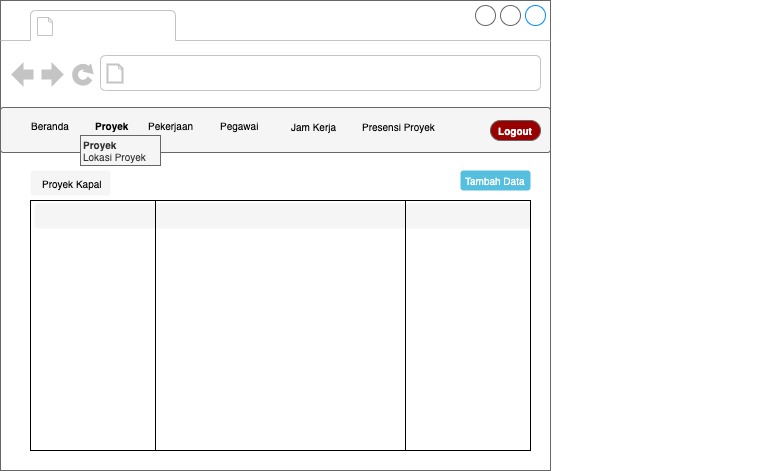
1. Beranda Admin Proyek



Gambar 3. 15 Beranda Admin Proyek

Pada Gambar 3.15 merupakan tampilan utama saat Admin Proyek berhasil *login* dan membuka sistem. Di halaman ini terdapat tabel yang menampilkan pekerjaan proyek yang sedang berlangsung.

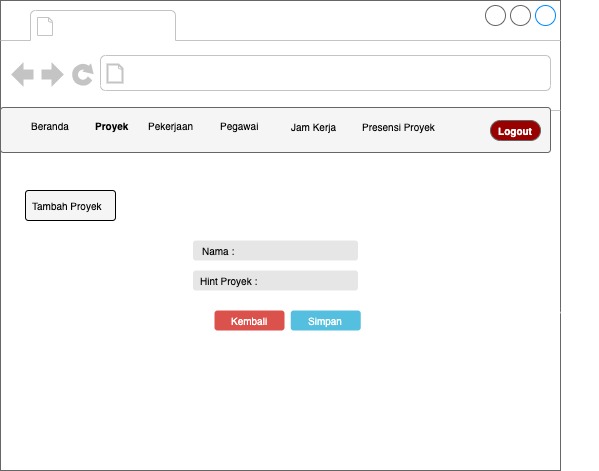
1. Halaman Data Proyek pada Admin Proyek



Gambar 3. 16 Halaman Data Proyek pada Admin Proyek

Pada Gambar 3.16 merupakan tampilan halaman proyek pada Admin Proyek. Di halaman ini terdapat tabel yang menampilkan daftar proyek. Terdapat tombol untuk menambahkan proyek baru.

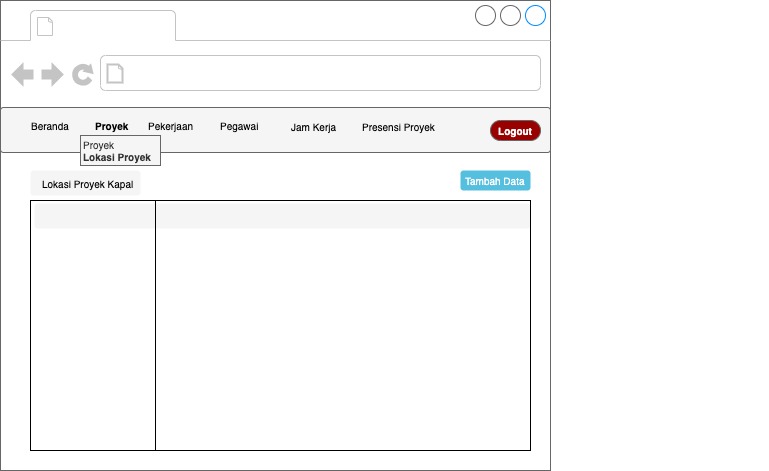
1. Halaman Tambah Proyek pada Admin Proyek



Gambar 3. 17 Halaman Tambah Proyek padaAdmin Proyek

Pada Gambar 3.17 merupakan tampilan halaman tambah proyek pada Admin Proyek. Di halaman ini pengguna menambahkan nama atau deskripsi proyek serta *hint* proyek.

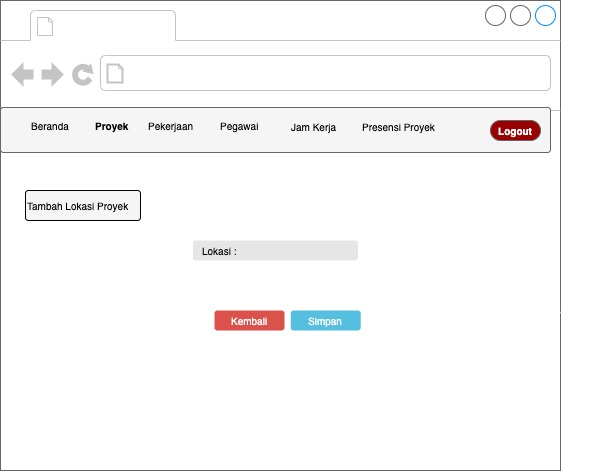
1. Halaman Data Lokasi pada Admin Proyek



Gambar 3. 18 Halaman Data Lokasi pada Admin Proyek

Pada Gambar 3.18 merupakan tampilan halaman lokasi presensi pada Admin Proyek. Di halaman ini terdapat tabel yang menampilkan daftar lokasi presensi proyek. Terdapat tombol untuk menambahkan lokasi presensi proyek baru.

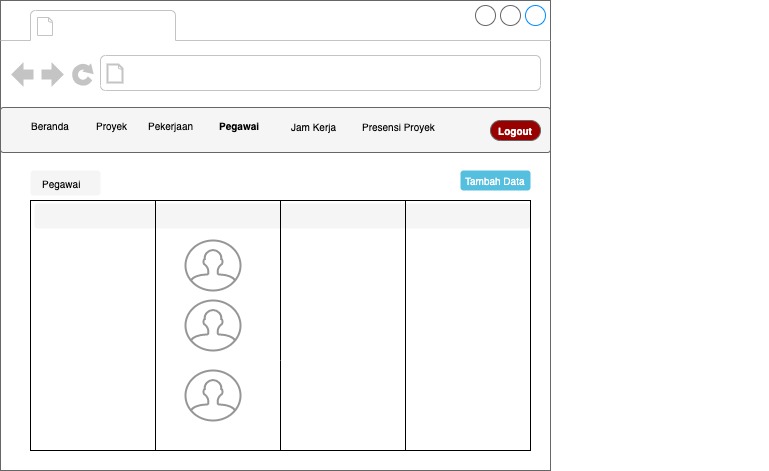
1. Halaman Tambah Lokasi pada Admin Proyek



Gambar 3. 19 Halaman Tambah Lokasi pada Admin Proyek

Pada Gambar 3.19 merupakan tampilan halaman tambah lokasi pada Admin Proyek. Di halaman ini pengguna menambahkan lokasi presensi proyek yang baru.

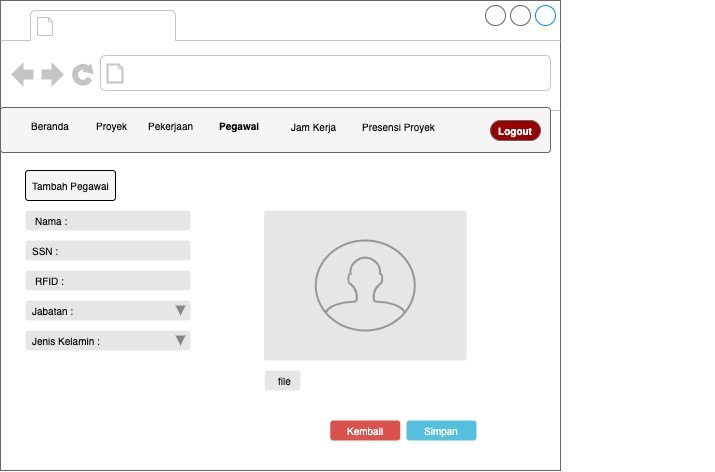
1. Halaman Data Pegawai pada Admin Proyek



Gambar 3. 20 Halaman Data Pegawai pada Admin Proyek

Pada Gambar 3.20 merupakan tampilan halaman pegawai pada Admin Proyek. Di halaman ini terdapat tabel yang menampilkan daftar pegawai. Terdapat tombol untuk menambahkan pegawai baru.

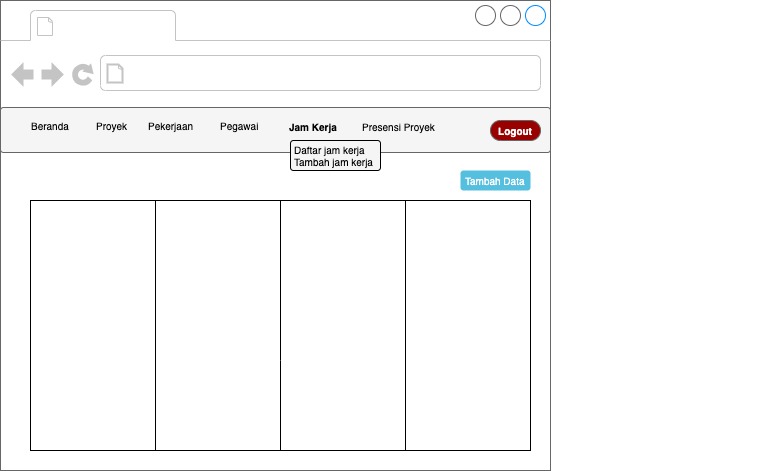
1. Halaman Tambah Pegawai pada Admin Proyek



Gambar 3. 21 Halaman Tambah Pegawai pada Admin Proyek

Pada Gambar 3.21 merupakan tampilan halaman tambah pegawai pada Admin Proyek. Di halaman ini pengguna menambahkan nama pegawai SSN (nomor pegawai), *RFID,* jabatan pegawai, jenis kelamin pegawai dan foto pegawai.

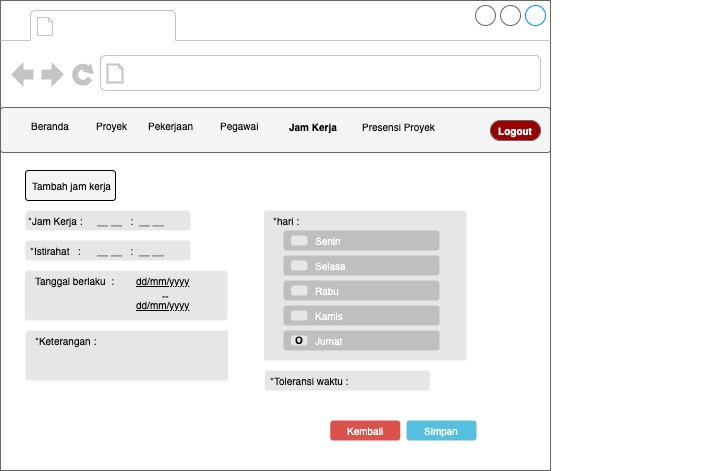
1. Halaman Data Jam Kerja pada Admin Proyek



Gambar 3. 22 Halaman Data Jam Kerja pada Admin Proyek

Pada Gambar 3.22 merupakan tampilan halaman jam kerja pegawai pada Admin Proyek. Di halaman ini terdapat tabel yang menampilkan daftar jam kerja pegawai. Terdapat tombol untuk menambahkan jam kerja baru.

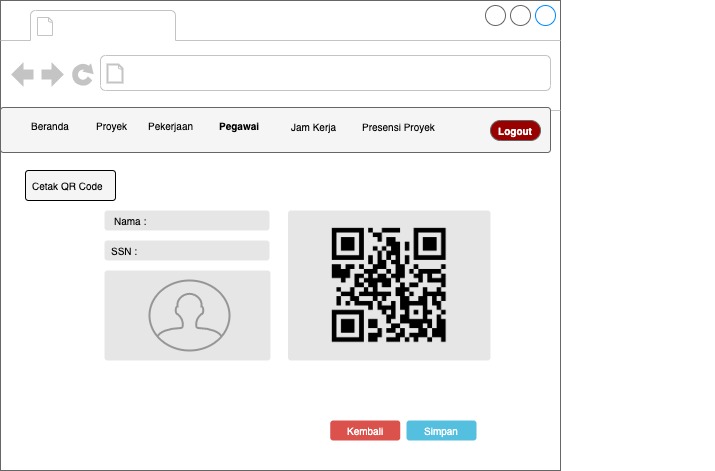
1. Halaman Tambah Jam Kerja pada Admin Proyek



Gambar 3. 23 Halaman Tambah Jam Kerja pada Admin Proyek

Pada Gambar 3.23 merupakan tampilan halaman tambah jam kerja pegawai pada Admin Proyek. Di halaman ini pengguna menambahkan jam kerja dimulai dan diakhiri, waktu istirahat mulai dan berakhir, tanggal berlaku jam kerja, keterangan dari jam kerja, hari berlaku jam kerja, serta toleransi keterlambatan pegawai.

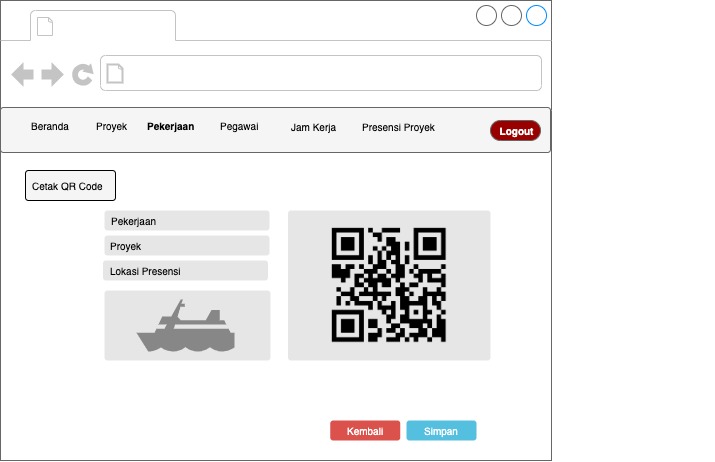
1. Halaman Cetak *QR Code* Pegawai pada Admin Proyek



Gambar 3. 24 Halaman *Cetak QR Code* Pegawai

Pada Gambar 3.24 merupakan halaman untuk mencetak *QR Code* Pegawai oleh Admin Proyek*. QR Code* ini menampilkan nama, SSN, foto pegawai dan *QR Code* id pegawai .

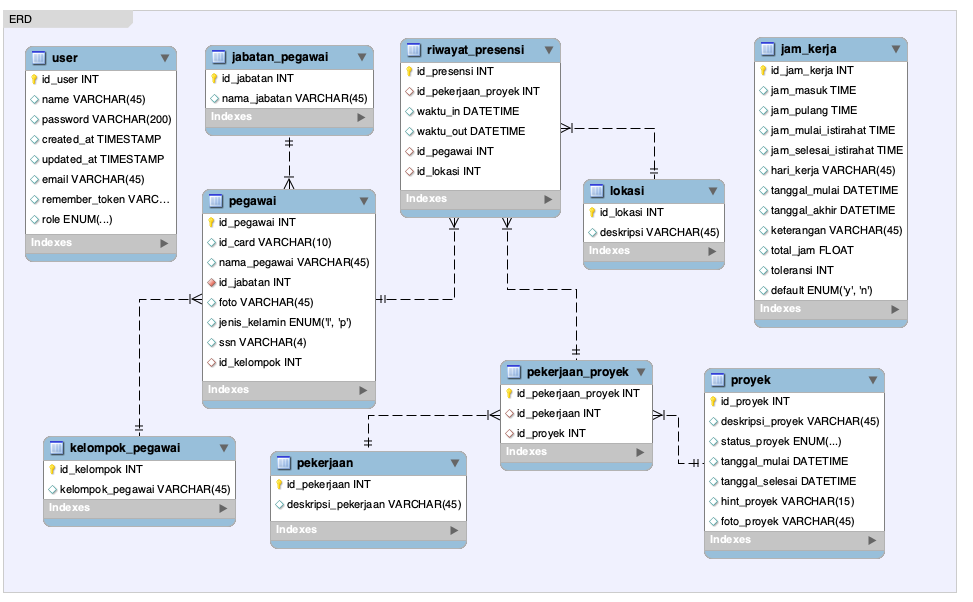
1. Halaman Cetak *QR Code* Pekerjaan pada Admin Proyek



Gambar 3. 25 Halaman *Cetak QR Code* Pekerjaan

Pada Gambar 3.25 merupakan halaman untuk mencetak *Cetak QR Code* Pekerjaan pada Admin Proyek*. QR Code* ini menampilkan pekerjaan, proyek, lokasi presensi, foto proyek dan *QR Code.* *QR Code* ditempatkan di komputer presensi

### Relasi Antar Tabel



Gambar 3. 26 *Entity Relationalship Diagram*

Gambar 3.26 merupakan rancangan basisdata menggunakan ERD. Perancangan menggunakan ERD ini merupakan langkah awal dalam membuat basisdata setelah melakukan proses analisa data. Pada sistem ini memiliki beberapa tabel, dimana tiap tabel dijelaskan pada Tabel 3.13 sampai Tabel 3.22.

Tabel 3. 15 Tabel *user*

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama Atribut** | **Tipe Data** | **Kata Kunci** | **Keterangan** |
| ide\_user | Int(5) | PK | Kode unik untuk pengguna |
| name | Varchar(45) |  | Nama pengguna |
| password | Varchar(200) |  | Kata sandi untuk pengguna |
| created\_at | Timestamp |  | Tanggal dimasukkannya pengguna |
| updated\_at | Timestamp |  | Tanggal *update* pada data |
| remember\_token | Varchar(45) |  | *Field default* bawaan dari *laravel* |
| email | Varchar(45) |  | Berisi email dari pengguna |
| role | Enum(‘manajer’,‘admin’) |  | Level pengguna dibagi berdasarkan *role.* |

Tabel 3.15 merupakan tabel *user* yang digunakan untuk untuk menyimpan data pengguna, ide\_user diatur *auto increment,* sehingga pada *field* ini akan terisi otomatis dan bersifat unik. Tabel ini dibuat secra otomatis menggunakanan *Laravel.*

Tabel 3. 16 Tabel pegawai

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama Atribut** | **Tipe Data** | **Kata Kunci** | **Keterangan** |
| ide\_pegawai | Int(11) | PK | Nomor unik yang otomatis terisi |
| ide\_card | Varchar(10) |  | Berisi nomor *rfid* pegawai |
| nama\_pegawai | Varchar(200) |  | Nama lengkap pegawai |
| ide\_jabatan | Int(5) | FK | Merupakan ide jabatan yang di *foreign key* dengan id\_jabatan di tabel jabatan |
| foto | Varchar(45) |  | Berisi nama dari *file* foto pegawai |
| jenis\_kelamin | Enum(‘l’,’p’) |  | Jenis kelamin bertipe data enum l untuk laki-lagi dan p untuk perempuan |
| ssn | Varchar(4) |  | Merupaka nomor pegawai dengan Panjang nomor 4 digit |
| id\_kelompok | Int(11) | FK | Merupakan id kelompok kerja yang di *foreign key* dengan id\_kelompok di tabel kelompok\_pegawai |

Tabel 3.16 merupakan tabel pegawaiyang digunakan untuk untuk menyimpan data pegawai, id\_pegawai diatur *auto increment*, sehingga pada field ini akan terisi otomatis dan bersifat unik, id\_jabatan merupakan *foreign key* dari id\_jabatan di tabel jabatan.

Tabel 3. 17 Tabel jabatan\_pegawai

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama Atribut** | **Tipe Data** | **Kata Kunci** | **Keterangan** |
| id\_jabatan | Int(11) | PK | Nomor unik yang otomatis terisi |
| deskripsi | Varchar(45) |  | Deskripsi dari jabatan |

Tabel 3.17 merupakan tabel jabatan\_pegawai yang digunakan untuk untuk menyimpan data jabatan\_pegawai, id\_jabatan diatur *auto increment,* sehingga pada *field* ini akan terisi otomatis dan bersifat unik.

Tabel 3. 18 Tabel lokasi

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama Atribut** | **Tipe Data** | **Kata Kunci** | **Keterangan** |
| id\_lokasi | Int(11) | PK | Nomor unik yang otomatis terisi |
| deskripsi | Varchar(45) |  | Deskripsi dari lokasi komputer untuk presensi proyek |

Tabel 3.18 merupakan tabel lokasi yang digunakan untuk untuk menyimpan data lokasi komputer presensi, id\_lokasi diatur *auto increment,* sehingga pada *field* ini akan terisi otomatis dan bersifat unik.

Tabel 3. 19 Tabel proyek

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama Atribut** | **Tipe Data** | **Kata Kunci** | **Keterangan** |
| id\_proyek | Int(11) | PK | Nomor unik yang otomatis terisi |
| deskripsi\_proyek | Varchar(45) |  | Berisi detail proyek; seperti nama kapal atau seri kapal |
| status\_proyek | Enum(‘done’,’onprogres’) |  | Status berfungsi untuk memfilter proyek yang sedang berjalan atau sudah selesai. |
| tanggal\_mulai | Int(5) |  | Berisi tanggal ketika menambahkah proyek |
| tanggal\_selesai | Datetime |  | Berisi tanggal selesai proyek, tanggal diinputkan manual oleh Admin Proyek |
| hint\_proyek | Varchar(15) |  | Merupakan kode unik/nomor yang dibuat untuk setiap kapal (mirip dengan *IMEI* pada telepon) |
| Foto\_proyek | Varchar(45) |  | Merupakan *field* untuk menyimpan nama foto proyek. |

Tabel 3.19 merupakan tabel proyek yang digunakan untuk untuk menyimpan data proyek kapal, id\_proyek diatur *auto increment,* sehingga pada *field* ini akan terisi otomatis dan bersifat unik.

Tabel 3. 20 Tabel Pekerjaan

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama Atribut** | **Tipe Data** | **Kata Kunci** | **Keterangan** |
| id\_pekerjaan | Int(11) | PK | Nomor unik yang otomatis terisi |
| Deskripsi\_pekerjaan | Varchar(45) |  | Status berfungsi untuk memfilter proyek yang sedang berjalan atau sudah selesai. |

Tabel 3.20 merupakan tabel pekerjaan yang digunakan untuk untuk menyimpan data pekerjaan, id\_pekerjaan diatur *auto increment*.

Tabel 3. 21 Tabel pekerjaan\_proyek

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama Atribut** | **Tipe Data** | **Kata Kunci** | **Keterangan** |
| id\_pekerjaan\_proyek | Int(11) | PK | Nomor unik yang otomatis terisi |
| id\_proyek | Int(11) | FK | Merupakan id proyek yang di *foreign key* dengan id\_proyek di tabel proyek |
| id\_pekerjaan | Int(11) | FK | Merupakan id proyek yang di *foreign key* dengan id\_pekerjaan di tabel pekerjaan |

Tabel 3.21 merupakan tabel pekerjaan proyek yang digunakan untuk untuk menyimpan data pekerjaan proyek yang ada, id\_pekerjaan diatur *auto increment*, sehingga pada *field* ini akan terisi otomatis dan bersifat unik, pada tabel pekerjaan memiliki 2 *field foreign key*.

Tabel 3. 22 Tabel jam\_kerja

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama Atribut** | **Tipe Data** | **Kata Kunci** | **Keterangan** |
| id\_jam\_kerja | Int(11) | PK | Nomor unik yang otomatis terisi |
| Jam\_masuk | Time |  | Jam masuk dari pegawai |
| Jam\_pulang | Time |  | Jam pulang dari pegawai |
| Jam\_mulai\_istirahat | Time |  | Jam mulai istirahat pegawai |
| Jam\_selesai\_istirahat | Time |  | Jam selesai istirahat pegawai |
| Hari\_kerja | Varchar(45) |  | Dalam form input berisi *checkbox* hari kerja yang akan dipilih. Data didalam field berupa *array* |
| Tanggal\_mulai | Datetime |  | Tanggal mulai dari diterapkannya jam kerja |
| Tanggal\_akhir | Datetime |  | Tanggal berakhir dari diterapkannya jam kerja |
| Keterangan | Varchar(45) |  | Ketaranag jam kerja. Contoh : “jam kerja bulan puasa” |
| Total\_jam | Float |  | Merupakan nilai total jam dalam bentuk desimal |
| Toleransi | Int(11) |  | Merupakan waktu toleransi keterlambatan *scan* pekerjaan di awal hari. Nilai berupa *integer* yang nantinya di konversi kedalam menit/waktu. |
| Default | Enum(‘y’,’n’) |  | Jika nilai ‘y’ maka menjadi jam kerja *default* |

Tabel 3.22 merupakan tabel jam\_kerja yang digunakan untuk untuk menyimpan data jam kerja pegawai, id\_jam\_kerja diatur *auto increment*, sehingga pada field ini akan terisi otomatis dan bersifat unik.

Tabel 3. 23 Tabel riwayat\_presensi

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama Atribut** | **Tipe Data** | **Kata Kunci** | **Keterangan** |
| id\_presensi | Int(11) | PK | Nomor unik yang otomatis terisi |
| id\_pegawai | Int(11) | FK | Merupakan id dari pegawai yang di *foreign key* dengan id\_pegawai di tabel pegawai |
| Id\_pekerjaan\_proyek | Int(11) | FK | Merupakan id pekerjaan yang di *foreign key* dengan id\_pekerjaan\_proyek di tabel pekerjaan\_proyek |
| id\_lokasi | Int(11) | FK | Merupakan id lokasi yang di *foreign key* dengan id\_lokasi di tabel lokasi |
| waktu\_in\_scan | Datetime |  | Waktu *scan* input pekerjaan |
| waktu\_out\_scan | Datetime |  | Waktu *out* merupakan *field* yang terisi otomatis ketika jam istirahat, jam pulang atau seseorang pegawai melakukan *scan* pekerjaan baru |

Tabel 3.23 merupakan tabel riwayat\_presensi yang digunakan untuk untuk menyimpan data riwayat presensi proyek kapal, id\_presensi diatur *auto increment*, sehingga pada field ini akan terisi otomatis dan bersifat unik, pada tabel ini memiliki 3 *field foreign key*.

Tabel 3. 24 Tabel kelompok pegawai

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama Atribut** | **Tipe Data** | **Kata Kunci** | **Keterangan** |
| id\_kelompok | Int(11) | PK | Nomor unik yang otomatis terisi |
| Kelompok\_pegawai | Varchar(20) |  | Deskripsi dari kelompok kerja |

Tabel 3.24 merupakan tabel kelompok kerja yang digunakan untuk untuk menyimpan data kelompok kerja dari pegawai, id\_kelompok\_pegawai diatur *auto increment,* sehingga pada *field* ini akan terisi otomatis dan bersifat unik.

--*Halaman ini sengaja dikosongkan--*

# DAFTAR PUSTAKA

Ermatita, Heroza, R. I. & Jannah, M., 2017. *Pengembangan Sistem Absensi Menggunakan QR Code Reader Berbasis Android (Studi Kasus : Fakultas Ilmu Komputer Jurusan Sistem Informasi UNSRI),* Palembang: s.n.

Handika, I. G. & Ayi Purbasari, 2018. Pemanfaatan Framework Laravel Dalam Pembangunan Aplikasi E-Travel Berbasis Website. *Konferensi Nasional Sistem Informasi,* pp. 1329-1334.

Harisantyo, B. et al., 2015. *Makalah Pemrograman Berbasis Objek Diagram Activity,* Depok: Universitas Gunadrama Fakultas Teknologi Industri Teknik Informatika.

Jaya, T. S., 2018. Pengujian Aplikasi dengan Metode Blackbox Testing Boundary Value Analysis (Studi Kasus: Kantor Digital Politeknik Negeri Lampung). *Jurnal Informatika: Jurnal Pengembangan IT (JPIT),* 03(02), pp. 45-46.

Kosasi, S. & Yuliani, I. D. A. E., 2015. Penerapan Rapid Application Development Pada Sistem Penjualan Sepeda Online. *Jurnal Simetris,* 6(1), pp. 28-29.

Mustaqbal, M. S., Firdaus, R. F. & Rahmadi, H., 2015. Pengujian Aplikasi Menggunakan Black Box Testing Boundary Value Analysis. *Jurnal Ilmiah Teknologi Informasi Terapan,* 1(3), p. 34.

Noviyasari, C., t.thn. *Unikom.* [Online]   
Available at: repository.unikom.ac.id   
[Diakses 5 Februari 2020].

Nugraha, A. S., 2018. *Sistem Informasi Manajemen Presensi Kehadiran Menggunakan Qr Code Berbasis Web Dan Sms Gateway Di Smk Muhammadiyah 2 Sukoharjo.* Surakarta: s.n.

Pamukti, P. G., 2019. *Aplikasi Absensi Mahasiswa Dengan Scan Qr Code Berbasis Android.* Banyuwangi: s.n.

Putri, D. P. M. & Heru Supriyono, 2019. Rancang Bangun Sistem Presensi Berbasis QR Code Menggunakan Framework Codeigniter (Studi Kasus Kehadiran Asisten Praktikum). *Inssypro (informatioan system processing),* 4(1), pp. 1-9.

Raharjo, B., 2018. *Belajar Otodidak Framework Codeigniter.* Revisi penyunt. Bandung: Penertbit INFORMATIKA Bandung.

Rembulan, A. D., 2015. *Makalah Unified Modeling Language (UML),* Cirebon: Stikom Poltek.

Rukmana, A. & Desiyani, I. D., 2017. *Metodologi Dan Metode Rapid Application Development (RAD),* Sumedang: Sekolah Tinggi Manajemen Informatika dan Komputer (STMIK) Sumedang.

Sucipto, 2017. Perancangan Active Database System pada Sistem Informasi Pelayanan Harga Pasar. *Jurnal Intensif,* 1(1), pp. 35-36.

Warman, I. & Rizki Ramdaniansyah, 2018. Analisis Perbandingan Kinerja Query Database Management System (Dbms) Antara Mysql 5.7.16 Dan Mariadb 10.1. *Jurnal TEKNOIF,* 6(1), pp. 32-41.

Yusmiarti, K., 2016. Perancangan Sistem Distribusi Produk Teh Hitam Berbasis Web Pada PTPN VII Gunung Dempo Pagar Alam. *Jurnal Informatika,* 4(2), pp. 3-4.