

**PEMBUATAN SISTEM PRESENSI DAN EVALUASI PADA
LEMBAGA KURSUS DAN PELATIHAN BELAJAR
MESSENGER ACADEMY SURABAYA**

KERJA PRAKTEK



UBAYA
UNIVERSITAS SURABAYA

Disusun Oleh :

- 1. JUSTIN ANDREW LESMANA
160420026**
- 2. RUTH FLODIAN RAHAKBAUW
160420076**

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNIK UNIVERSITAS SURABAYA
SURABAYA
APRIL 2024**

BAB 1

PENDAHULUAN

1. JUDUL

Pembuatan Sistem Presensi dan Evaluasi Pada Lembaga Kursus dan Pelatihan Messenger Academy Surabaya

2. LATAR BELAKANG

Messenger Academy merupakan suatu Lembaga Kursus dan Pelatihan yang memiliki berbagai pelatihan bakat, minat dan bimbingan belajar berstandar *World System*. Messenger Academy memiliki pengajar sebanyak 25 hingga 30 pengajar dan setiap pengajar dapat mengajar rata-rata 2 sampai 4 murid baik untuk kelas individu maupun kelas kelompok. Setiap pertemuan yang dilakukan antara pengajar dan murid akan dicatat dengan mengisi presensi laporan setiap pertemuan dan akan dijadikan laporan setiap bulan. Dalam meningkatkan layanan untuk murid maupun orang tua, lembaga kursus dan pelatihan Messenger Academy ingin adanya Sistem Presensi Online.

Messenger Academy mewajibkan pengajar untuk mengisi presensi setiap murid setelah pertemuan berlangsung, selanjutnya admin wajib untuk membuat laporan berdasarkan presensi yang telah diisi oleh pengajar. Saat ini presensi dan pembuatan laporan masih dilakukan secara manual oleh pengajar dan admin. Pengajar mengisi presensi dengan menggunakan Google Form dan wajib mengisi presensi sebanyak empat dalam satu bulan untuk setiap murid. Setelah itu admin akan melakukan rekap keseluruhan pertemuan dan dari Excel rekap dijadikan laporan dalam format Word.

Terkadang karena banyaknya murid dan presensi yang diisi secara manual, pengajar seringkali tidak mengisi presensi sehingga dapat menghambat pembuatan laporan di akhir bulan.

Oleh karena itu, dengan sistem presensi online dapat membantu mempermudah baik admin, pengajar, hingga orang tua dan murid dalam presensi dan laporan. Laporan dapat diketahui langsung oleh orang tua sehingga tidak menunggu akhir bulan untuk memantau perkembangan belajar anak. Pengajar tidak perlu mengisi form yang banyak dan admin tidak perlu melakukan rekap bulanan.

3. RUMUSAN MASALAH

Bagaimana membantu Messenger Academy dalam presensi dan evaluasi kegiatan kursus?

4. TUJUAN

Tujuan dari kerja praktek ini adalah membuat sistem presensi dan evaluasi untuk membantu memudahkan presensi dan memudahkan pembuatan dan pemberitahuan laporan evaluasi pertemuan setiap bulan.

5. MANFAAT

1. Membantu pengajar dalam membuka absensi berdasarkan kelas yang diajar
2. Memudahkan murid melakukan presensi dari kelas yang diambil
3. Memudahkan orang tua untuk melihat presensi anaknya

4. Membantu pihak manajemen dalam mengatur data murid yang ingin ditampilkan dalam batasan tertentu (misal berdasarkan tahun)

6. RUANG LINGKUP

1. Tempat dilakukan di Lembaga Kursus, Pelatihan, dan Bimbingan Belajar Messenger Academy
2. Sistem akan digunakan oleh role pengajar, admin, dan orang tua serta murid
3. Sistem akan terdapat fitur untuk menambah data peserta, pelatihan, dan orang tua
4. Sistem dapat digunakan untuk melakukan absensi dan mencatat evaluasi pertemuan
5. Sistem dapat menampilkan laporan bulanan terkait pertemuan

7. METODOLOGI PENELITIAN

Metodologi penelitian yang dilakukan dalam penelitian tugas akhir ini adalah sebagai berikut.

1. Persiapan

Langkah dalam perencanaan pembuatan sistem yang dilakukan sebagai berikut.

- Mencari informasi melalui wawancara dengan pihak Pusat Konsultasi dan Layanan Psikologi Universitas Surabaya yang menyediakan tes gaya belajar dan keterampilan belajar.
- Membuat proposal untuk merencanakan sistem dan menjelaskan garis besar sistem.
- Melakukan studi literatur untuk mendapatkan informasi terkait implementasi serupa yang sudah pernah dilakukan.

- Menentukan rencana fitur yang akan ada pada sistem dengan membuat gambaran sistem.

2. Analisis

Analisis dilakukan dengan melakukan observasi dan wawancara dengan pihak terkait tes gaya belajar dan keterampilan belajar untuk dapat menganalisis kondisi saat ini, masalah, dan merancang fitur sistem analisis tes gaya belajar dan keterampilan belajar sesuai dengan kebutuhan.

3. Desain

Berdasarkan hasil analisis, desain yang akan dilakukan adalah mendesain Entity Relationship Diagram (ERD) untuk penyimpanan data. Setelah itu, membuat desain alur pada sistem dengan Business Process Model and Notation (BPMN) yang bertujuan sebagai gambaran cara kerja sistem untuk pengguna. Selanjutnya, membuat gambaran User Interface (UI) dengan guna memberikan gambaran awal tampilan dan cara penggunaan aplikasi.

4. Implementasi

Dari hasil desain yang sudah ditentukan, maka dilakukan implementasi dengan memastikan proses persiapan tes, proses tes, dan hasil analisis tes gaya belajar dan keterampilan belajar sesuai dengan perhitungan pada Pusat Konsultasi dan Layanan Psikologi.

5. Uji Coba dan Evaluasi

Tahap ini melibatkan tim Pusat Konsultasi dan Layanan Psikologi yang merupakan pelajar untuk melakukan uji coba pada sistem. Tim akan melakukan uji coba proses persiapan tes, proses jalannya tes, dan proses analisis.

6. Penyusunan Laporan

Penulisan laporan akan dilakukan secara berkala sesuai dengan proses pembuatan aplikasi. Laporan akan berisi hasil dari keseluruhan penelitian dan jawaban dari rumusan masalah.

8. SISTEMATIKA PENELITIAN

Sistematika penulisan bertujuan sebagai gambaran isi dari laporan penelitian ini. Penelitian ini terdiri dari tujuh bab sebagai berikut

1. BAB 1: PENDAHULUAN

Berisi mengenai penjelasan yang mendorong penulisan tugas akhir tentang pembuatan sistem analisis tes gaya belajar dan keterampilan belajar yang ditujukan untuk PKLP. Mulai dari latar belakang, rumusan masalah, tujuan, manfaat, ruang lingkup, metodologi penelitian, dan sistematika penulisan penelitian.

2. BAB 2: DASAR TEORI

Berisi mengenai penjelasan landasan teori terkait hal-hal dalam pembuatan tugas akhir sistem analisis tes gaya belajar dan keterampilan belajar pada pelajar untuk PKLP.

3. BAB 3: ANALISIS SISTEM

Berisi mengenai analisis keadaan saat ini, analisis masalah, analisis sistem serupa, serta analisis kebutuhan sistem sebagai kesimpulan dalam kebutuhan fitur yang digunakan pada pembuatan sistem ini.

4. BAB 4: DESAIN SISTEM

Berisi mengenai desain sistem yang akan dibuat berdasarkan fitur yang sudah ditentukan. Bagian ini menjelaskan mengenai desain dan alur kerja sistem.

5. BAB 5: IMPLEMENTASI

Berisi mengenai implementasi sistem berdasarkan rancangan yang dilakukan pada bab sebelumnya. Pada implementasi ini akan menjelaskan penerapan yang akan digunakan pada implementasi sistem.

6. BAB 6: VERIFIKASI DAN VALIDASI

Berisi kegiatan verifikasi dan validasi sistem yang telah diimplementasi. Bagian ini menjelaskan hasil uji coba yang dilakukan.

7. BAB 7: SARAN DAN KESIMPULAN

Berisi mengenai kesimpulan atas penelitian yang telah dilakukan. Kemudian saran yang didapatkan akan digunakan dalam penelitian yang akan datang.

BAB 2

LANDASAN TEORI

Bab ini menjelaskan mengenai teori dasar dalam pembuatan tugas akhir yang berjudul Pembuatan Sistem Presensi dan Evaluasi pada Lembaga Kursus dan Pelatihan. Teori dasar yang akan dibahas meliputi sistem presensi dan sistem evaluasi.

2.1 SISTEM PRESENSI

Peran yang ada dalam sistem presensi dan evaluasi dalam pendidikan dibagi menjadi admin, pengajar, dan murid (Munthe et al., 2021). Admin bertugas untuk mengatur CRUD (Create Read Update Delete) pada database terkait serta manajemen database berupa backup, reset, upload database. Pengajar memiliki fitur untuk membuka presensi, mengecek kehadiran, dan mengupload kehadiran murid. Sedangkan murid memiliki fitur berupa melakukan presensi, mengecek jadwal, dan menyesuaikan kehadiran

Dalam proyek kerja praktek kami admin memiliki peran untuk memasukkan pengajar dan mahasiswa ke dalam database, mengubah presensi, mengupdate kehadiran mahasiswa, mengubah hak akses pengajar, dan membuka kelas baru. Selain itu, admin juga bertanggung jawab atas database yang dimilikinya dengan cara manajemen database tersebut. Pengajar hanya dapat membuka kelas di kelas yang diajar oleh pengajar tersebut. Apabila setelah pengajar mengupload kehadiran murid dan terjadi ketidaksesuaian, maka akan status kehadiran akan mengikuti kehadiran murid yang diupload pengajar. Jika masih ada ketidaksesuaian, murid dapat mengajukan penyesuaian kehadiran.

Menurut Wantoro (2016), sistem presensi ini memungkinkan untuk melakukan rekap dengan cepat dan dalam jangka waktu yang tepat karena semua data terdapat dalam database, namun backup database harus secara rutin dilakukan agar data tetap valid. Rekap akan lebih mudah dilakukan apabila data dapat dipilih secara langsung dari database. Dalam database kerja praktek kami dicatat tanggal/ waktu presensi, kelas, dan setiap muridnya sehingga data tersebut dapat digunakan untuk mencari data sesuai kebutuhan rekap tiap kelasnya. Backup database digunakan apabila admin melakukan kesalahan dalam CRUD pada database sehingga data dapat dilihat perubahannya dari database backup dan mengembalikan data yang tidak valid tersebut.

2.2 SISTEM EVALUASI

Proses pengumpulan data nilai dan proses input data nilai merupakan langkah-langkah yang digunakan dalam sistem evaluasi (Satria & Ardiansyah, 2023). Data nilai dan prosesnya dapat dilakukan oleh masing-masing pengajar sesuai kebutuhannya. Contohnya jika suatu kelas yang terdapat 4 kali penilaian, pengajar dapat memasukkan data sebanyak 4 kali untuk setiap muridnya. Kemudian dari data nilai yang akan diinput ke dalam sistem dan sistem akan menghitung rata-rata tiap murid dan akan menampilkan semua data nilai beserta rata-rata tiap murid. Dengan demikian penilaian dapat disesuaikan dengan masing-masing pengajarnya. Dalam proyek kami, sistem evaluasi hanya berbasis pada evaluasi tentang pendapat pengajar mengenai perkembangan pesertanya selama mengikuti kursus.

Perkembangan belajar murid sebaiknya dapat diakses kapanpun oleh orang tuanya (Merdekawati et al., 2022). Sistem evaluasi ini sebaiknya segera diperbarui setelah penilaian diinput ke dalam sistem. Hal ini bertujuan untuk mempercepat pembagian hasil penilaian murid sehingga murid dapat segera mengetahui hasil pembelajarannya dan mengambil

tindakan atas nilai yang didapatkan agar penilaian selanjutnya akan mendapatkan hasil yang lebih baik. Orang tua juga dapat memotivasi anaknya agar penilaian berikutnya murid dapat berkembang.

2.3 PENGGUNAAN CSV UNTUK INPUT BANYAK DATA

Inkonsistensi dan masalah keamanan data merupakan kelemahan saat menggunakan CSV (*Comma Separated Value*) sehingga dibutuhkan *database* untuk menyimpan data tersebut dengan aman (sitasi). Pada sistem kami, CSV nantinya akan digunakan untuk memasukkan data pengajar, orang tua, peserta, dan pelatihan. CSV ini akan digunakan untuk memasukkan setiap data yang ada pada CSV ke dalam database. Dengan CSV ini, sistem kami akan memanfaatkan kekurangan dari file CSV untuk mempermudah dan mempercepat proses input

BAB 3

ANALISIS SISTEM

Bab ini menjelaskan mengenai analisis yang akan digunakan dalam pembuatan sistem. Berdasarkan informasi yang didapatkan dari pihak Lembaga Kursus dan Pelatihan Messenger Academy Surabaya dengan observasi dan wawancara, analisis sistem akan meliputi analisis kondisi yang terjadi saat ini, analisis pada sistem sejenis, analisis permasalahan yang ada, serta analisis kebutuhan yang dibutuhkan oleh sistem.

3.1 ANALISIS KONDISI SAAT INI

Proses untuk menganalisis kondisi saat ini dilakukan dengan menggunakan metode observasi pada *website* Messenger Academy dan observasi pada proses kegiatan pelatihan *online* maupun *offline* untuk dapat mengetahui lebih dalam Messenger Academy. Selain itu, metode wawancara dengan admin maupun guru yang melakukan presensi untuk dapat mengetahui sistem presensi yang berjalan saat ini.

3.1.1 HASIL OBSERVASI

Lembaga Kursus dan Pelatihan Messenger Academy Surabaya merupakan pusat pelatihan bakat, minat dan bimbingan belajar terlengkap dan berstandar internasional disertai pendidikan karakter. Kelas kursus dan pelatihan yang ditawarkan yaitu *Public Speaking, Cinematography, Poster Design, Creative Animation, Photography & Video Editing, Youtuber, Fashion Design, Vocal, Dance/ Ballet, Make up, Creative Craft, Graphic Design, Mandarin Language, Korean Language, Japanese Language, English Language, Tutoring (BimBel), E-Sport, Piano Music, Guitar Music, Other Music, University Companion, Coding/ Programming, Science.*

Setiap kursus dan pelatihan terdiri dari beberapa level, yaitu *Beginner*, *Intermediate*, *Advanced*. Setiap kelulusan akan mendapatkan sertifikat kelulusan sesuai dengan level yang dijalani. Terdapat beberapa program kelas yang dapat diikuti di Messenger Academy, yaitu

1. Training Program

Training Program merupakan program kerjasama yang diadakan oleh Messenger Academy bersama dengan instansi pendidikan maupun perusahaan untuk memberikan pembekalan dan pembelajaran kepada peserta didik sesuai dengan kebutuhan. Program ini terbuka bagi sekolah, universitas, maupun perusahaan untuk memperlengkapi siswa, mahasiswa, dan karyawan dengan program pembelajaran bakat dan minat dari Messenger Academy.

2. Holiday Program

Holiday Program merupakan kelas jangka pendek yang terbuka pada saat liburan sekolah dan selesai hanya dalam 2 bulan (8x pertemuan). Holiday program dibuka untuk kelas-kelas tertentu, seperti *Creative Animation*, *Youtuber*, *Public Speaking*, *Make Up*, *Acting*, dan *Photography*. Kelas holiday program dapat diikuti secara *online* maupun *offline*. Program ini terbuka untuk jenjang pendidikan TK - SD - SMP - SMA - Universitas.

3. Open Road Program

Workshop Open Road Class merupakan program Workshop yang diadakan setiap 2-3 bulan sekali dengan menyuguhkan pelatihan-pelatihan menarik dalam satu kali pertemuan dan diikuti oleh peserta dari seluruh Indonesia.

4. Character Building

Character Building merupakan salah satu program Messenger Academy yang diadakan untuk memberikan pendidikan karakter kepada semua student Messenger Academy. Character Building diadakan setiap 1x dalam sebulan (di akhir bulan) secara online.

3.1.2 HASIL WAWANCARA

Wawancara dilakukan untuk dapat mengetahui lebih dalam proses yang terjadi sehingga dapat mempertimbangkan kebutuhan sistem yang sesuai dan memudahkan pelaksanaan presensi. Wawancara dilakukan dengan admin dan guru untuk mengetahui bagaimana proses presensi yang berjalan saat ini. Selanjutnya wawancara dilakukan dengan pengelola yang membantu proses berjalannya presensi dan laporan kepada orang tua.

Wawancara dilakukan dengan mengajukan pertanyaan terkait proses presensi yang saat ini sedang berjalan. Saat ini presensi masih dilakukan secara manual dengan menggunakan dua cara, yaitu dengan presensi menggunakan kertas maupun *Google Form*. Presensi diisi ketika kelas selesai dilakukan. Apabila presensi tidak dilakukan oleh pengajar, maka akan berpengaruh pada upah pengajar. Hal tersebut karena laporan sangat dibutuhkan untuk dapat dilaporkan pada orang tua murid.

Presensi manual dengan menggunakan kertas dilakukan pada saat kelas dilakukan secara tatap muka dengan mendatangi meja admin untuk meminta kertas yang sesuai dengan kelas yang diajarkan. Setiap bulan admin harus mencetak kertas presensi untuk dapat melakukan presensi. Kertas presensi akan direkap oleh admin setiap akhir bulan untuk dijadikan laporan dalam bentuk PDF dan dikirimkan ke setiap orang tua murid.

Presensi manual dengan menggunakan *Google Form* biasanya digunakan untuk pengajar yang melakukan kelas secara *online*. Setelah kelas, pengajar harus mengakses *Google Form*

untuk mengisi presensi dan akan direkap oleh admin pada akhir bulan untuk dijadikan laporan dalam bentuk PDF. Menurut admin, menggunakan *Google Form* lebih efektif untuk mencari *history* pengajaran bulan sebelumnya karena jauh lebih mudah daripada presensi dengan menggunakan kertas. Oleh karena itu, sistem presensi akan lebih membantu agar seluruh kelas dapat dijadikan satu sistem presensi dan dalam melakukan rekap maupun laporan akan lebih efektif.

3.2 ANALISIS SISTEM SEJENIS

Analisis pada sistem sejenis dilakukan untuk dapat mengetahui analisis yang dilakukan pada penelitian terdahulu terkait sistem presensi dan evaluasi pada lembaga kursus dan pelatihan.

3.2.1 PENELITIAN IMPLEMENTASI APLIKASI PRESENSI SISWA DAN TUTOR BERBASIS WEB DI LPK WIDI COURSE KUDUS

Listyorini dan Shaleh (2022) membuat sistem presensi yang digunakan oleh siswa dan tutor untuk melakukan presensi siswa dan tutor setiap kehadiran di Widi Course agar tidak terjadi hilang catatan dari siswa dan tutor yang membuat perhitungan presensi kurang efektif. Dengan demikian, diharapkan dapat membantu lembaga untuk dapat merekap presensi untuk digunakan juga sebagai perhitungan upah pengajarnya. Implementasi penelitian ini memiliki dua role, yaitu siswa dan tutor.

Siswa dan tutor dapat melakukan login untuk dapat masuk pada *dashboard user*. Pada halaman *dashboard user* dapat melihat langsung data-data maupun informasi siswa melalui grafik lingkaran. Di dalam *dashboard* kita dapat melihat menu master data, ambil *QR code*, *scan QR code*, data absensi dan setting. Data *user* siswa maupun tutor dapat dilihat pada master data. Rekap keseluruhan dari data siswa dan tutor dapat kita lakukan aksi lihat, edit dan hapus.

Selanjutnya pada halaman scan, nantinya *QR code* yang sudah dicetak dapat dilakukan *scan* pada halaman ini dengan menggunakan kamera perangkat *mobile* dan presensi dilakukan, pengguna cukup melihatkan kartu yang sudah dilengkapi dengan *QR code* ke kamera HP atau laptop admin. Pada halaman *history* presensi berisikan data setiap member yang telah melakukan presensi menggunakan *QR code*. Sedangkan laporan rekap presensi dapat mencetak laporan rekap presensi siswa maupun tentor sebagai laporan ke pemilik LPK *Widi Course*.

No	NAMA	STATUS	Total Kehadiran	Sakit	Total Ketidakhadiran
					Ijin Alpha
1	Zuha	SISWA	1	0	1 0
2	Faisal Wahyu	SISWA	2	1	0 0

Gambar 3.1 Halaman Laporan Presensi

3.2.2 PENELITIAN APLIKASI SISTEM INFORMASI PRESENSI PESERTA DIDIK PADA SMAK ST. AUGUSTINUS NGANJUK

Pada penelitian yang dilakukan Kesuma dan Kholifah (2019) yang membuat sistem informasi akademik untuk mencatat presensi siswa. Melalui sistem ini, diharapkan dapat membantu untuk mengetahui status kehadiran siswa, seperti hadir, terlambat, maupun tidak hadir. Sistem presensi dilengkapi dengan informasi berbasis SMS yang akan diinformasikan kepada orang tua dengan status terdiri dari keterangan : NIS, Nama, Jam Masuk, Telat, Sakit, Ijin, Alpha.

Fitur yang terdapat pada sistem informasi akademik meliputi login yang dapat dilakukan oleh admin, guru, dan siswa. Halaman home ini merupakan tampilan yang dapat diakses oleh guru. Pada halaman ini guru hanya bisa menginputkan centang kehadiran siswa untuk proses absensi di kelas. Semua data yang telah tersimpan akan masuk kedalam sistem. Halaman input siswa absen dapat diakses oleh siswa yang berfungsi untuk memasukkan data siswa yang tidak hadir, dengan keterangan ijin maupun sakit. Jika tidak memasukkan data tersebut maka siswa dianggap tidak hadir tanpa keterangan.

Setiap sekolah memiliki jam efektif belajar masing-masing. Oleh karena itu dibuat halaman input jam masuk sehingga admin perlu memasukkan jam masuk setiap harinya. Terdapat halaman data siswa berisikan NIS, nama siswa, kelas dan nomor handphone wali siswa yang dapat di filter dengan kelas. Halaman ini berfungsi untuk menambah, mengubah dan menghapus data-data tersebut. Fitur terakhir adalah halaman laporan yang berfungsi untuk mencetak laporan yang didapat di filter dengan bulan dan kelas. User dapat langsung mencetak laporan jika perangkat sudah terhubung dengan mesin cetak.

APLIKASI ABSENSI SMAK ST. AUGUSTINUS NGANJUK
Welcome, XII-IPS Logout

Absensi Siswa

NIS	Nama	Status
9971004722	Pegi Andiana	<input type="checkbox"/>
9971080921	Dewi Lola Lusiana	<input checked="" type="checkbox"/>
99771317983	Charlene Kweesley	<input checked="" type="checkbox"/>
9978180727	Roni Setiawan	<input checked="" type="checkbox"/>
9980828419	Yudi Indri Wenita Pratama	<input checked="" type="checkbox"/>
9980905587	Ryan Andik Saputra	<input checked="" type="checkbox"/>
9983636495	Amy Fransiska	<input checked="" type="checkbox"/>
9983636513	Nurpita Sari	<input checked="" type="checkbox"/>
998748482	Yohanes Yoyok Kriswanto	<input type="checkbox"/>

Simpan

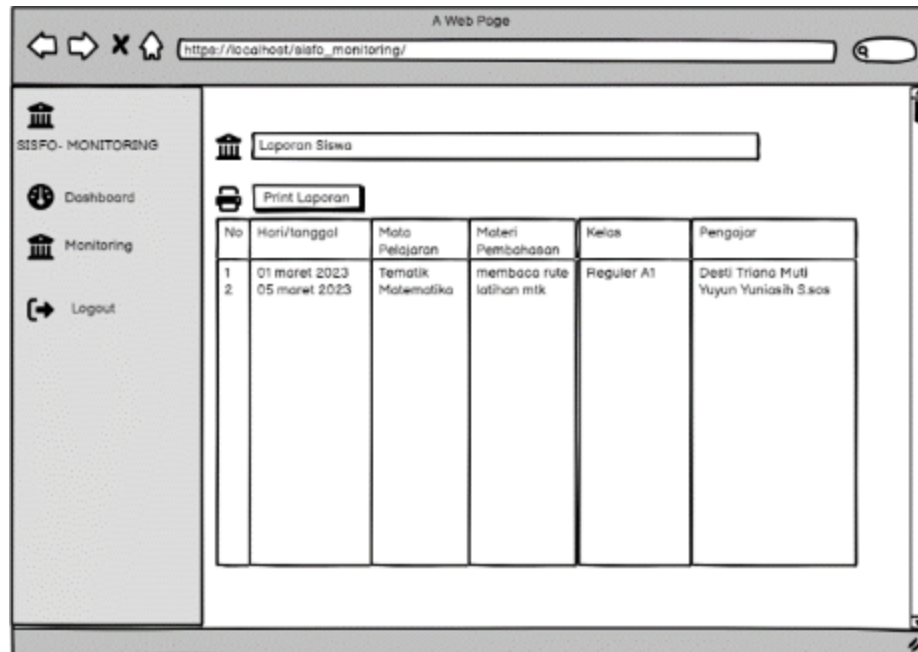
Gambar 3.2 Halaman Presensi Siswa

3.2.3 PENELITIAN FRAMEWORK CODEIGNITER UNTUK PENGEMBANGAN SISTEM INFORMASI MONITORING AGENDA KELAS DAN PRESENSI SISWA (LEMBAGA BIMBINGAN BELAJAR BRILLIANT LEARNING CENTER)

Pada penelitian Nistrina, Ghivari, dan Kurniawan (2023) membuat sistem yang dapat membantu Bimbel BLC dalam mengoptimalkan kegiatan monitoring agenda kelas dan presensi siswa. Hal tersebut ditujukan untuk merancang dan membangun sistem informasi monitoring yang dapat diakses oleh guru sebagai pemasukkan data agenda kelas dan presensi siswa. Selain itu juga dapat menyampaikan data agenda kelas dan presensi siswa yang akurat dan real time pada orang tua siswa karena menurut pihak bimbel, orang tua siswa berhak tahu bagaimana agenda kelas dan bagaimana status kehadiran anak mereka.

Perancangan sistem ini terdapat role admin dan orang tua. Tetapi penelitian akan berfokus pada fitur dan tampilan admin. Pada awal penggunaan sistem, admin harus melakukan login, sistem akan mengarahkan pada halaman *dashboard*. Pada halaman *dashboard* terdapat menu agenda kelas yang terdapat beberapa atribut diantaranya label halaman data kelas, tabel data kelas, tombol tambah, tombol edit, dan tombol hapus.

Pada halaman presensi siswa terdapat beberapa atribut diantaranya label halaman presensi siswa, tabel presensi siswa, tombol cetak presensi, tombol generate presensi dan tombol edit. Selanjutnya terdapat halaman laporan presensi siswa yang dapat diupload untuk akun orang tua yang terdapat beberapa atribut diantaranya label halaman laporan siswa, tabel laporan siswa dan tombol print laporan siswa.



Gambar 3.3 Halaman Presensi Siswa

Tabel 3.1 Perbandingan Sistem Sejenis

Sistem LKP Widi <i>Course</i>	Sistem SMAK St. Augustinus	Sistem Bimbel BLC
Terdapat role siswa dan tentor	Menggunakan role admin, guru, siswa	Role digunakan oleh admin dan orang tua
Login dapat dilakukan oleh seluruh role	Login hanya dapat dilakukan oleh admin saja	Login hanya dilakukan admin
Fitur <i>add, edit, delete</i> digunakan pada data dosen siswa dan tentor	Fitur <i>add, edit, delete</i> digunakan pada absen siswa, hari libur, data	Fitur <i>add, edit, delete</i> digunakan pada fitur agenda kelas dan

	siswa, admin, dan kelas	presensi kelas
Hasil laporan yang dikeluarkan adalah laporan nama maupun status kehadiran setiap kelas untuk periode bulanan	Hasil laporan yang dikeluarkan adalah laporan nama maupun status kehadiran setiap kelas untuk periode bulanan yang berupa file	Hasil laporan yang dikeluarkan adalah laporan waktu les, mata pelajaran dan materi yang dipelajari, kelas, maupun pengajar

3.3 ANALISIS MASALAH

Analisis masalah ditujukan untuk mengetahui permasalahan atau kesulitan yang saat ini terjadi. Permasalahan yang terjadi saat ini, yaitu:

- Proses presensi tidak dilakukan saat berakhir kelas, jadi seringkali terlambat dalam pengisian presensi.
- Proses laporan kepada orang tua seringkali terlambat karena menunggu presensi yang diisikan oleh pengajar.
- Admin harus melakukan input laporan secara manual dan dikirim pada masing-masing orang tua murid.

3.4 ANALISIS KEBUTUHAN SISTEM

Berdasarkan analisis yang dilakukan sebelumnya, maka sistem presensi dan evaluasi dibuat dengan harapan dapat membantu pihak LKP Messenger Academy dalam melakukan presensi maupun laporan. Sistem ini akan dibuat untuk tiga role, yaitu role untuk admin, pengajar, dan orang tua. Kegunaan setiap role adalah sebagai berikut:

1. Admin bertugas untuk mencatat setiap kelas yang berlangsung pada setiap minggu, memantau apakah pengisian presensi setiap pengajar, dan mengirim akses kepada orang tua untuk dapat melihat laporan pengajaran dalam bulan tersebut.
2. Pengajar bertugas mengisi presensi untuk setiap murid yang telah diajar.
3. Orang tua berfungsi untuk memantau perkembangan anak dalam mengikuti LKP.

Beberapa fitur yang dibutuhkan sebagai kebutuhan sistem antara lain:

1. Pengisian Data Murid

Fitur pengisian data ini dilakukan pada role admin. Admin mengisi data murid dengan detail lainnya yang sesuai dengan murid tersebut, seperti nama, kelas, pelatihan yang diikuti, pengajar yang mengajar, dan sebagainya yang digunakan untuk keperluan LKP.

2. Pengisian Data Kelas

Fitur pengisian data ini dilakukan pada role admin. Admin mengisi data kelas yang akan dilakukan dalam satu minggu kedepan. Admin harus memasukkan detail guru siapa saja yang akan mengajar murid, kapan kelas tersebut akan berlangsung, dan dimana kelas tersebut akan berlangsung.

3. Presensi

Fitur presensi merupakan fitur utama dalam sistem ini. Pengajar akan mengisi presensi dengan melengkapi materi apa saja yang dipelajari, penilaian maupun catatan yang diberikan kepada murid tersebut.

4. Laporan

Fitur laporan dapat diakses oleh orang tua untuk mengetahui waktu kelas, apa saja yang dipelajari, bagaimana proses belajar dengan melihat catatan pengajar, maupun penilaian pada murid dari segi keaktifan dan penangkapan materi.

BAB 4

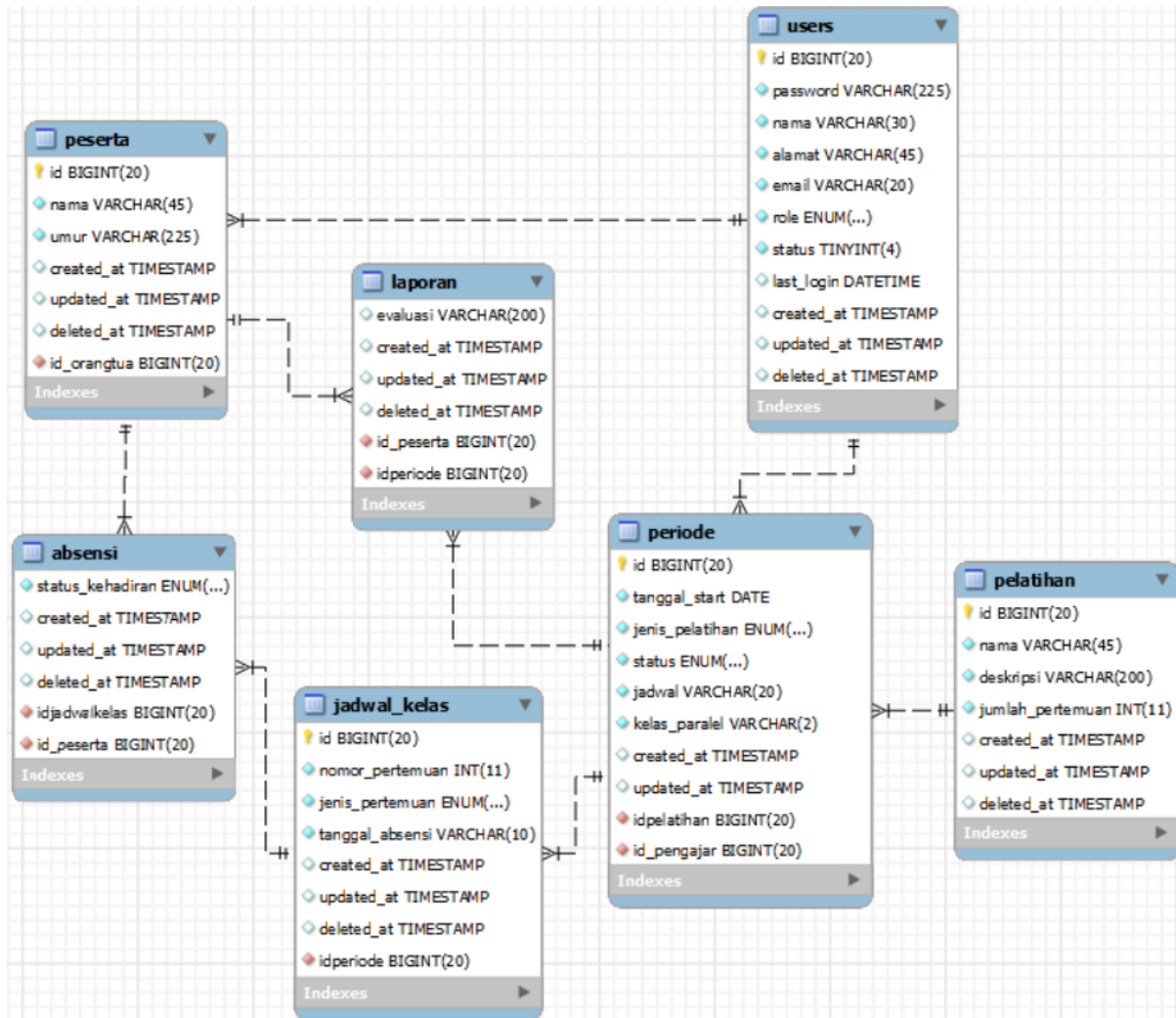
DESAIN SISTEM

Bab ini akan membahas mengenai hasil desain sistem informasi presensi dan evaluasi pada lembaga kursus Messenger Academy Surabaya yang dibuat berdasarkan hasil analisis sistem. Desain yang dibahas pada bab ini meliputi Entity Relationship Diagram (ERD), desain proses dan design User Interface.

4.1 DESAIN *ENTITY RELATIONSHIP DIAGRAM* (ERD)

Perancangan desain basis data pada sistem analisis tes gaya belajar dan keterampilan belajar akan menggunakan *Entity Relationship Diagram* (ERD). Entity Relationship Diagram pada sistem informasi presensi dan evaluasi pada lembaga kursus Messenger Academy Surabaya dari 5 entitas utama dan 1 entitas hasil relasi many-to-many dari entitas utama.

ERD ini dibuat menggunakan MySQL Workbench. Entitas utama meliputi entitas peserta, orang_tua, absensi, pengajar, dan pelatihan. Entitas yang terbentuk karena relasi many-to-many adalah laporan. Entitas laporan dihasilkan dari relasi antara entitas periode dan entitas peserta. ERD ini digunakan untuk menyimpan data absensi dari murid dan penilaiannya. Di sisi lain, ERD ini digunakan untuk menyimpan informasi kelas, pengajar, murid, dan orang tua murid dengan detail yang diperlukan untuk sistem ini.



**Gambar 4.1 ERD Sistem Informasi Presensi dan Evaluasi pada Lembaga Kursus
Messenger Academy Surabaya**

4.1.1 USERS

Tabel users menyimpan data pengguna dengan berbagai peran dalam sistem, seperti orang_tua, pengajar, dan admin. Setiap pengguna memiliki informasi seperti password, nama, alamat, email, role, status, serta timestamp untuk last_login, created_at, updated_at, dan deleted_at. Primary key id digunakan sebagai referensi di tabel peserta untuk menunjukkan hubungan orang tua dan anak, serta di tabel periode untuk menunjukkan pengajar yang

bertanggung jawab atas suatu periode pelatihan. Email pengguna juga harus unik, yang dijamin oleh unique index pada kolom email.

4.1.2 PESERTA

Tabel peserta menyimpan data peserta pelatihan, yang mencakup informasi seperti nama, umur, dan timestamp untuk `created_at`, `updated_at`, serta `deleted_at`. Setiap peserta terhubung dengan orang tua melalui foreign key `id_orangtua` yang merujuk ke kolom `id` di tabel `users`. Tabel peserta juga berhubungan dengan tabel absensi dan laporan di mana `id` peserta digunakan sebagai foreign key untuk mencatat kehadiran dan evaluasi peserta dalam pelatihan.

4.1.3 ABSENSI

Tabel absensi mencatat status kehadiran peserta untuk setiap jadwal kelas, dengan informasi seperti `status_kehadiran` (`hadir`, `alfa`, `sakit`, `ijin`), serta timestamp untuk `created_at`, `updated_at`, dan `deleted_at`. Tabel ini memiliki dua foreign key: `idjadwalkelas` yang merujuk ke kolom `id` di tabel `jadwal_kelas` untuk menunjukkan jadwal kelas tertentu, dan `id_peserta` yang merujuk ke kolom `id` di tabel `peserta` untuk menunjukkan peserta yang kehadirannya dicatat.

4.1.4 JADWAL KELAS

Tabel `jadwal_kelas` menyimpan jadwal pertemuan untuk setiap periode pelatihan, dengan informasi seperti `nomor_pertemuan`, `jenis_pertemuan` (`pengganti` atau `reguler`), `tanggal_absensi`, serta timestamp untuk `created_at`, `updated_at`, dan `deleted_at`. Setiap jadwal kelas terhubung dengan periode pelatihan melalui foreign key `idperiode` yang merujuk ke kolom `id` di tabel `periode`. Tabel ini juga berhubungan dengan tabel absensi di mana `id` jadwal kelas digunakan sebagai foreign key untuk mencatat kehadiran peserta dalam setiap pertemuan.

4.1.5 PERIODE

Tabel periode mencatat informasi tentang periode pelatihan, termasuk tanggal_start, jenis_pelatihan (kelompok atau private), status (berjalan atau selesai), jadwal, kelas_paralel, serta timestamp untuk created_at, updated_at, dan deleted_at. Setiap periode terkait dengan satu pelatihan melalui foreign key idpelatihan yang merujuk ke kolom id di tabel pelatihan, dan juga terkait dengan satu pengajar melalui foreign key id_pengajar yang merujuk ke kolom id di tabel users. Selain itu, periode pelatihan juga dihubungkan dengan tabel jadwal_kelas dan laporan melalui foreign key idperiode.

4.1.6 PELATIHAN

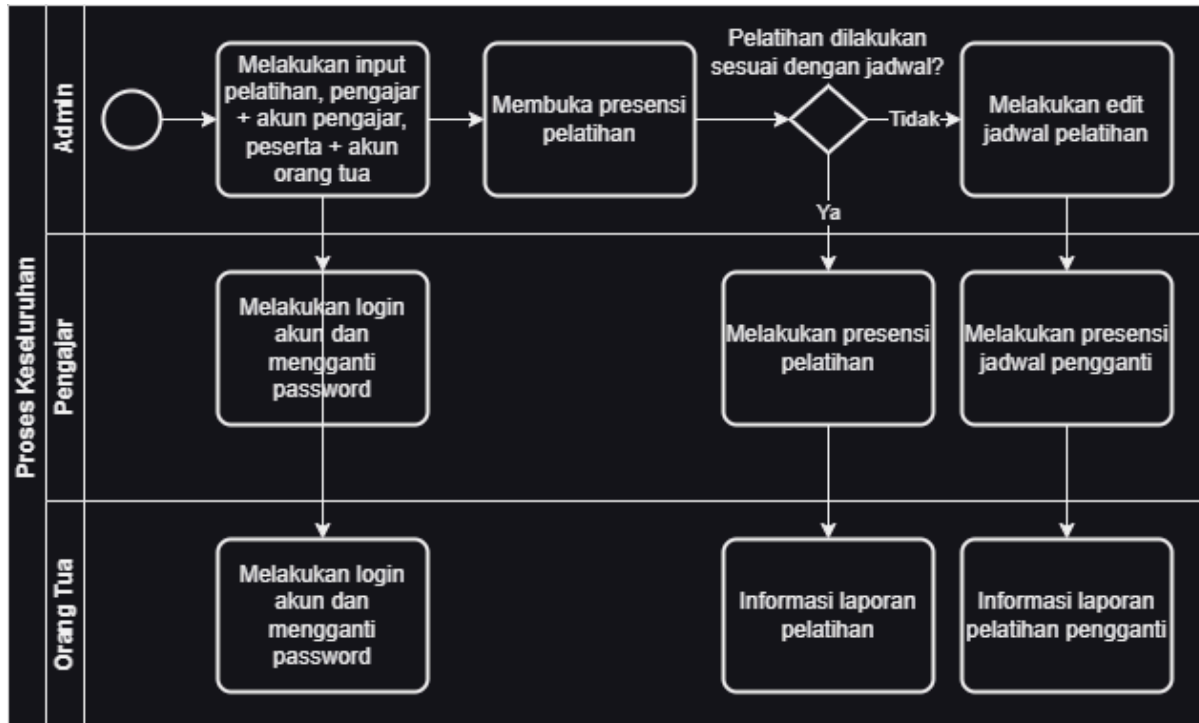
Tabel pelatihan menyimpan data pelatihan yang tersedia dalam sistem, dengan informasi seperti nama, deskripsi, jumlah_pertemuan, serta timestamp untuk created_at, updated_at, dan deleted_at. Setiap pelatihan memiliki primary key id, yang digunakan sebagai foreign key di tabel periode untuk mengidentifikasi pelatihan mana yang sedang berlangsung dalam periode tertentu. Tabel pelatihan berdiri sendiri tanpa hubungan langsung dengan tabel lain selain melalui tabel periode.

4.1.7 LAPORAN

Tabel laporan menyimpan evaluasi pelatihan untuk setiap peserta, dengan informasi seperti evaluasi, serta timestamp untuk created_at, updated_at, dan deleted_at. Setiap laporan terkait dengan satu peserta melalui foreign key id_peserta yang merujuk ke kolom id di tabel peserta, dan juga terkait dengan satu periode pelatihan melalui foreign key idperiode yang merujuk ke kolom id di tabel periode. Laporan ini membantu mengukur kinerja dan perkembangan peserta dalam periode pelatihan tertentu.

4.1 DESAIN *BUSINESS PROCESS MODEL AND NOTATION* (BPMN)

Perancangan desain proses pada Sistem Informasi Presensi dan Evaluasi akan menggunakan *Business Process Model and Notation* (BPMN). Pada BPMN berikut, dapat dilihat proses keseluruhan dan tugas dari beberapa role.

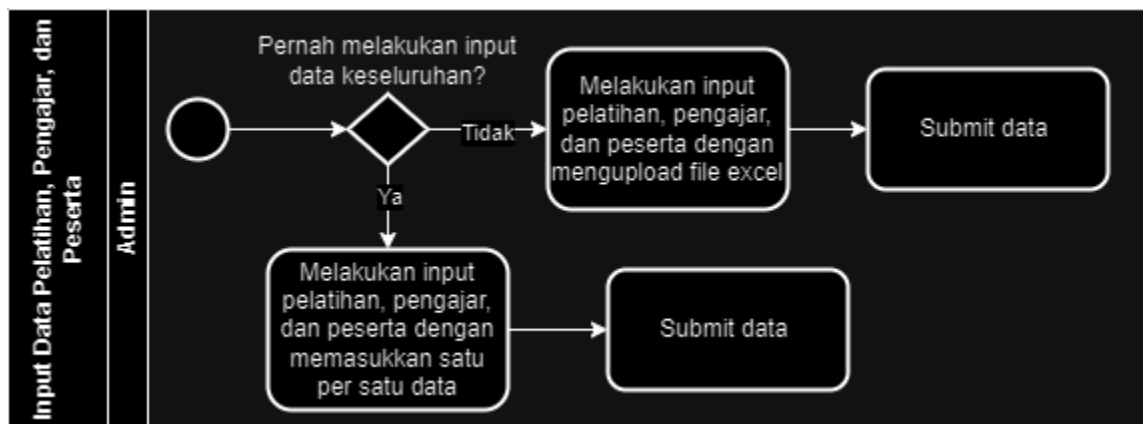


Gambar 4.2 BPMN Keseluruhan

4.2.1 PROSES INPUT DATA PELATIHAN, PENGAJAR, PESERTA, DAN ORANG TUA

Proses input data pelatihan akan dilakukan oleh admin. Apabila admin belum pernah melakukan input data, program menyediakan cara efektif dengan cara mengupload file csv yang berisi data pelatih, pengajar, dan peserta didalamnya. Setelah di upload, maka data akan langsung terinput pada database.

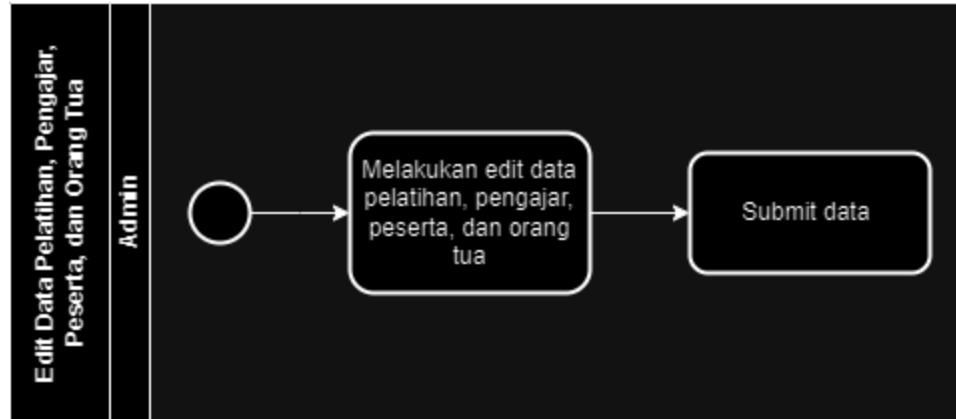
Apabila sewaktu-waktu admin perlu untuk melakukan input data satuan, maka program menyediakan fitur tersebut. Admin hanya perlu menekan tombol Add dan mengisi data satuan sesuai dengan inputan yang ada. Setelah seluruh data sudah terisi dengan benar, maka admin dapat langsung melakukan submit data.



Gambar 4.3 BPMN Input Data

4.2.2 PROSES EDIT DATA PELATIHAN, PENGAJAR, PESERTA, DAN ORANG TUA

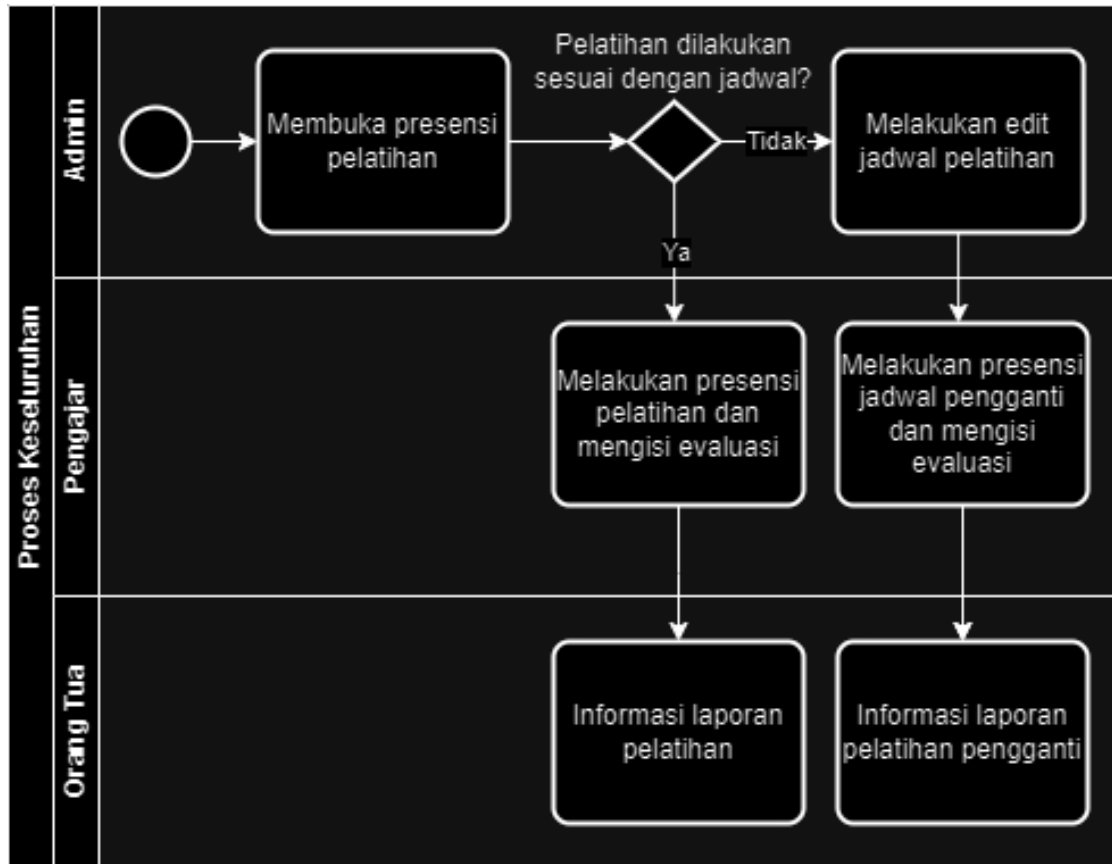
Proses edit data pelatihan akan dilakukan oleh admin. Apabila sewaktu-waktu admin perlu untuk melakukan edit data satuan, maka program menyediakan fitur tersebut. Admin hanya perlu menekan tombol edit dan melakukan edit data satuan sesuai dengan inputan yang ada. Setelah seluruh data sudah terisi dengan benar, maka admin dapat langsung melakukan submit data.



Gambar 4.4 BPMN Edit Data

4.2.3 PROSES PRESENSI DAN LAPORAN PELATIHAN

Proses presensi pelatihan akan dilakukan oleh pengajar setelah pelatihan sudah dibuka oleh admin. Admin wajib memasukkan jadwal pelatihan yang sudah ditentukan. Apabila pertemuan tersebut merupakan pertemuan pengganti, maka admin dapat merubah menjadi kelas pengganti. Setelah presensi dilakukan oleh pengajar, maka pada akun orang tua akan menampilkan data laporan pertemuan pada hari itu.

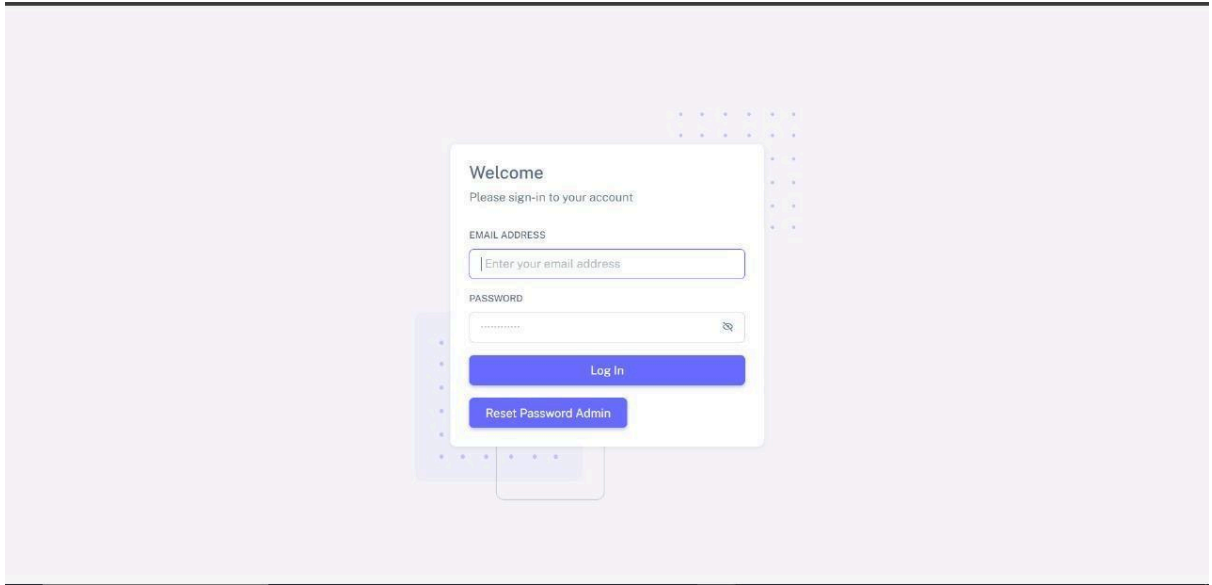


Gambar 4.5 BPMN Presensi dan Laporan

4.1 DESAIN *USER INTERFACE* (UI)

4.3.1 HALAMAN LOGIN

Halaman ini akan mengambil data alamat *email* dan *password* pengguna yang akan melakukan login. Kemudian akan dilakukan pengecekan ke dalam *database*. Pengguna dapat menekan ikon mata untuk melihat *password* yang sedang diketik. Halaman ini merupakan halaman yang pertama dibuka saat pengguna membuka aplikasi.



Gambar 4.6 Halaman Login

4.3.2 HALAMAN DAFTAR PENGAJAR

Halaman daftar pengajar ini terdapat pada pengguna dengan hak akses admin. Halaman ini menampilkan semua pengajar yang ada dalam sistem. Admin dapat menambahkan data pengajar baru dengan menekan tombol *add* atau *upload CSV*. Selain itu admin juga bisa melakukan pengeditan terhadap data pengajar.

11 Juli 2024 pukul 17:59:53

Upload CSV +Add

pengajar Jeromee is already updated

Show 10 entries Search:

NOMOR	NAMA LENGKAP	ALAMAT	EMAIL	LAST LOGIN	EDIT
1	Yantu	Jl. A. Yani No. 2, Surabaya	yantu@email.com		Edit
2	Yante	Jl. Pahlawan No. 42, Yogyakarta	yante@email.com		Edit
3	aaa@a	jalan ...	a@a		Edit
4	Jeromee	Suite 775	Jerome67@gmail.com		Edit
5	Suzanne	Suite 910	Suzanne.OReilly41@hotmail.com		Edit
6	Wilson	Suite 180	Wilson_Balistreri@gmail.com		Edit
7	Kristy	Suite 532	Kristy54@yahoo.com		Edit
8	Kenny	Suite 516	Kenny_Gutmann23@hotmail.com		Edit
9	Irene	Suite 224	Irene_Raynor-Nader58@hotmail.com		Edit
10	Amanda	Apt. 196	Amanda_Bogisich35@hotmail.com		Edit

Show 1 to 10 of 12 entries Previous 1 2 Next

Gambar 4.7 Halaman Daftar Pengajar

4.3.3 HALAMAN TAMBAH PENGAJAR

Halaman ini berguna untuk menambahkan data pengajar sesuai dengan *input* yang diberikan. Halaman ini hanya terdapat pada hak akses admin dan dikeluarkan saat menekan tombol *Add* pada halaman list pengajar. Setelah admin memasukkan semua datanya, maka data akan dimasukkan ke dalam database. Semua data pada halaman ini harus diisi sebelum dilakukan *submit*.

The screenshot shows a web application interface for 'Messenger 3D ACADEMY'. On the left is a sidebar menu with the following items: 'Daftar Pengajar' (highlighted), 'Daftar Peserta', 'Daftar Orang Tua', 'Daftar Pelatihan', 'Periode', and 'Logout'. The main content area is titled 'Add New Pengajar' and contains a form with the following fields: 'Nama', 'Alamat', 'Email', and 'Password'. Each field has a corresponding input box. Below the 'Password' field is a blue 'Submit' button. The top right corner of the page displays the date and time: '11 Juli 2024 pukul 18.01.14'.

Gambar 4.8 Halaman Tambah Pengajar

4.3.4 HALAMAN EDIT PENGAJAR

Halaman ini digunakan untuk melakukan perubahan pada data pengajar. Saat masuk ke dalam halaman ini, data lama sudah terisi pada input di halaman edit pengajar. Kemudian data tersebut akan dibawa ke dalam database untuk diubah nilainya. Semua data input perlu diisi dalam halaman ini sebelum melakukan proses submit.

Gambar 4.9 Halaman Edit Pengajar

4.3.5 HALAMAN DAFTAR PESERTA

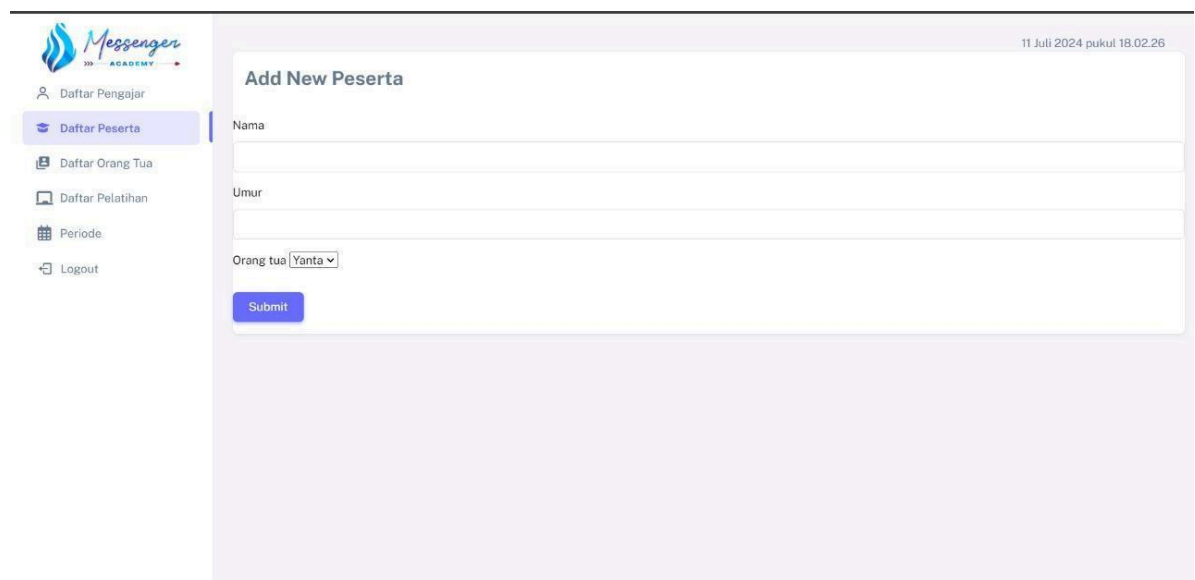
Halaman daftar peserta ini terdapat pada pengguna dengan hak akses admin. Halaman ini menampilkan semua peserta dengan nama orang tuanya yang ada dalam sistem. Admin dapat menambahkan data peserta baru dengan menekan tombol *add* atau *upload CSV*. Selain itu admin juga bisa melakukan pengeditan terhadap data peserta.

#	NAMA LENGKAP	UMUR	ORANG TUA	EDIT
1	Anak Yanta 1	14	Yanta	Edit
2	Anak Yanta 2	12	Yanta	Edit
3	Anak Yanta 3	10	Yanta	Edit
4	Anak Yanti 1	13	Yanti	Edit
5	Anak Yanti 2	11	Yanti	Edit
6	cc	12	Yanti	Edit
7	cccc	16	Yanti	Edit

Gambar 4.10 Halaman Daftar Pengajar

4.3.6 HALAMAN TAMBAH PESERTA

Halaman ini berguna untuk menambahkan data peserta sesuai dengan *input* yang diberikan dan pilihan orang tuanya. Halaman ini hanya terdapat pada hak akses admin dan dikeluarkan saat menekan tombol *Add* pada halaman list peserta. Setelah admin memasukkan semua datanya, maka data akan dimasukkan ke dalam database. data nama dan umur pada halaman ini harus diisi sebelum dilakukan *submit*.



The screenshot displays the 'Add New Peserta' form within the Messenger Academy interface. On the left, a sidebar menu includes options like 'Daftar Pengajar', 'Daftar Peserta' (which is highlighted), 'Daftar Orang Tua', 'Daftar Pelatihan', 'Periode', and 'Logout'. The main content area features a form with the following fields: 'Nama' (Name), 'Umur' (Age), and 'Orang tua' (Parent) with a dropdown menu currently showing 'Yanta'. A blue 'Submit' button is located at the bottom of the form. The top right corner of the page shows the date and time: '11 Juli 2024 pukul 18.02.26'.

Gambar 4.11 Halaman Tambah Peserta

4.3.7 HALAMAN EDIT PESERTA

Halaman ini digunakan untuk melakukan perubahan pada data peserta. Saat masuk ke dalam halaman ini, data lama sudah terisi pada input di halaman edit peserta. Kemudian data tersebut akan dibawa ke dalam database untuk diubah nilainya. Semua data input perlu diisi dalam halaman ini sebelum melakukan proses submit.

Messenger 300 ACADEMY

11 Juli 2024 pukul 18.03.38

Daftar Pengajar

Daftar Peserta

Daftar Orang Tua

Daftar Pelatihan

Periode

Logout

Edit Data Peserta

*Untuk data orang tua bisa diedit pada bagian list orang tua

Nama

Anak Yanta 1

Umur

14

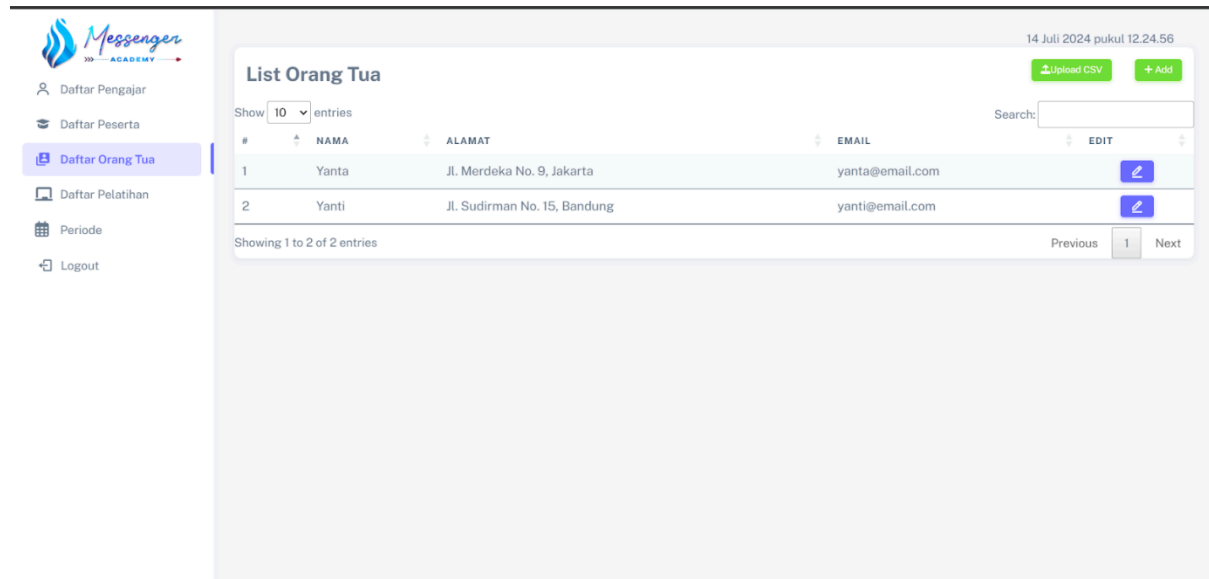
Orang tua: Yanta

Submit

Gambar 4.12 Halaman Edit Peserta

4.3.8 HALAMAN DAFTAR ORANG TUA

Halaman daftar orang tua ini terdapat pada pengguna dengan hak akses admin. Halaman ini menampilkan semua orang tua yang ada dalam sistem. Admin dapat menambahkan data orang tua baru dengan menekan tombol *add* atau *upload CSV*. Selain itu admin juga bisa melakukan pengeditan terhadap data orang tua.



Gambar 4.13 Halaman Daftar Orang Tua

4.3.9 HALAMAN TAMBAH ORANG TUA

Halaman ini berguna untuk menambahkan data orang tua sesuai dengan *input* yang diberikan. Halaman ini hanya terdapat pada hak akses admin dan dikeluarkan saat menekan tombol *Add* pada halaman list orang tua. Setelah admin memasukkan semua datanya, maka data akan dimasukkan ke dalam database. Semua data pada halaman ini harus diisi sebelum dilakukan *submit*.



The screenshot shows a web application interface for 'Messenger Academy'. On the left is a sidebar menu with options: 'Daftar Pengajar', 'Daftar Peserta', 'Daftar Orang Tua' (highlighted), 'Daftar Pelatihan', 'Periode', and 'Logout'. The top right corner displays the date and time: '11 Juli 2024 pukul 18.04.05'. The main content area is titled 'Add New Orang Tua' and contains a form with the following fields: 'Nama', 'Alamat', 'Email', and 'Password'. Each field has a corresponding input box. Below the 'Password' field is a blue 'Submit' button.

Gambar 4.14 Halaman Tambah Orang Tua

4.3.10 HALAMAN EDIT ORANG TUA

Halaman ini digunakan untuk melakukan perubahan pada data orang tua. Saat masuk ke dalam halaman ini, data lama sudah terisi pada input di halaman edit orang tua. Kemudian data tersebut akan dibawa ke dalam database untuk diubah nilainya. Semua data input perlu diisi dalam halaman ini sebelum melakukan proses submit.

Messenger ACADEMY

11 Juli 2024 pukul 18.09.01

Daftar Pengajar

Daftar Peserta

Daftar Orang Tua

Daftar Pelatihan

Periode

Logout

Edit Data Orang Tua

Nama

Yanta

Alamat


Jl. Merdeka No. 9, Jakarta

Submit

Gambar 4.15 Halaman Edit Orang Tua

4.3.11 HALAMAN PELATIHAN

Halaman daftar pelatihan ini terdapat pada pengguna dengan hak akses admin. Halaman ini menampilkan semua pelatihan yang ada dalam sistem. Admin dapat menambahkan data pelatihan baru dengan menekan tombol *add* atau *upload CSV*. Selain itu admin juga bisa melakukan pengeditan terhadap data pelatihan.



- Daftar Pengajar
- Daftar Peserta
- Daftar Orang Tua
- Daftar Pelatihan**
- Periode
- Logout






14 Juli 2024 pukul 12.26.23

Upload CSV

+ Add

Show 10 entries

Search:

#	NAMA	DESKRIPSI	JUMLAH PERTEMUAN	EDIT
1	Akuntansi	Akuntansi adalah ilmu yang mempelajari tentang pencatatan, pengol	6	
2	Dasar Pemrograman	Dasar Pemrograman adalah ilmu yang mempelajari tentang cara menggunakan bahasa pemrograman untuk menciptakan program komputer yang dapat bekerja dan berperilaku yang tepat.	8	
3	Psikologi Dewasa	Psikologi Dewasa adalah ilmu yang menggunakan teori dan metodologi yang berfokus pada perilaku, kebebasan, dan pengaruh psikologis di dalam	4	
4	Hukum Pidana	Hukum Pidana adalah ilmu yang menggunakan teori dan metodologi yang berfokus pada penyelesaian masalah-masalah yang terjadi di dalam perkawinan, keluarga, dan negara.	9	
5	Bisnis Keuangan	Bisnis Keuangan adalah ilmu yang menggunakan teori dan metodologi yang berfokus pada pencatatan, pengolahan, dan analisis ke	10	

Showing 1 to 5 of 5 entries

Previous

1

Next

Gambar 4.16 Halaman Pelatihan

4.3.12 HALAMAN TAMBAH PELATIHAN

Halaman ini berguna untuk menambahkan data pelatihan sesuai dengan *input* yang diberikan. Halaman ini hanya terdapat pada hak akses admin dan dikeluarkan saat menekan tombol *Add* pada halaman list pelatihan. Setelah admin memasukkan semua datanya, maka data akan dimasukkan ke dalam database. Semua data pada halaman ini harus diisi sebelum dilakukan *submit*

The screenshot shows a web interface for 'Messenger ACADEMY'. On the left is a sidebar with navigation links: 'Daftar Pengajar', 'Daftar Peserta', 'Daftar Orang Tua', 'Daftar Pelatihan' (highlighted), 'Periode', and 'Logout'. The main content area is titled 'Add New Pelatihan' and contains a form with the following fields: 'Nama' (text input), 'Deskripsi' (text input), and 'Jumlah Pertemuan' (text input). A blue 'Submit' button is located below the 'Jumlah Pertemuan' field. The top right corner of the page displays the date and time: '11 Juli 2024 pukul 18.09.26'.

Gambar 4.17 Halaman Tambah Pelatihan

4.3.13 HALAMAN EDIT PELATIHAN

Halaman ini digunakan untuk melakukan perubahan pada data pelatihan. Saat masuk ke dalam halaman ini, data lama sudah terisi pada input di halaman edit pelatihan. Kemudian data tersebut akan dibawa ke dalam database untuk diubah nilainya. Semua data input perlu diisi dalam halaman ini sebelum melakukan proses submit.

Messenger ACADEMY

11 Juli 2024 pukul 18.09.52

Daftar Pelatihan

Edit Pelatihan

Nama
Akuntansi

Deskripsi
Akuntansi adalah ilmu yang mempelajari tentang pencatatan, pengol...

Jumlah Pertemuan 6

Submit

Gambar 4.18 Halaman Edit Pelatihan

4.3.14 HALAMAN DAFTAR PERIODE


Halaman daftar periode pelatihan ini terdapat pada pengguna dengan hak akses admin. Halaman ini menampilkan semua periode pelatihan yang ada dalam sistem. Admin dapat menambahkan data pelatihan baru dengan menekan tombol *add* atau *upload CSV*. Selain itu admin juga bisa melakukan pengeditan terhadap data periode pelatihan. Pada halaman ini, admin dapat membuka absensi untuk setiap anggota kelasnya dan juga melakukan proses melihat anggota kelas.

#	NAMA PELATIHAN	NAMA PENGAJAR	TANGGAL START	JENIS PELATIHAN	KELAS PARALEL	STATUS	JADWAL PELATIHAN	EDIT	BUKA ABSENSI	LIHAT ANGGOTA
1	Akuntansi	Yantu	2023-09-01	private	A	berjalan	Senin,09.00-12.00		Buka Absensi	Lihat Anggota
2	Dasar Pemrograman	Yantu	2023-09-08	private	A	berjalan	Senin,13.00-15.00		Buka Absensi	Lihat Anggota
3	Dasar Pemrograman	Yantu	2023-09-15	private	B	berjalan	Selasa,09.00-12.00		Buka Absensi	Lihat Anggota
4	Psikologi Dewasa	Yante	2023-09-22	kelompok	A	berjalan	Selasa,13.00-15.00		Buka Absensi	Lihat Anggota
5	Hukum Pidana	Yante	2023-09-29	kelompok	A	berjalan	Rabu,09.00-12.00		Buka Absensi	Lihat Anggota
6	Bisnis Keuangan	Yante	2024-07-12	kelompok	A	berjalan	Senin,12.00-14.00		Buka Absensi	Lihat Anggota
7	Bisnis Keuangan	Yante	2024-07-12	kelompok	A	berjalan	Rabu,08.00-10.00		Buka Absensi	Lihat Anggota
8	Bisnis Keuangan	Yante	2024-07-12	kelompok	A	berjalan	Jumat,20.00-23.00		Buka Absensi	Lihat Anggota

Gambar 4.19 Halaman Daftar Periode

4.3.15 HALAMAN TAMBAH PERIODE

Halaman ini berguna untuk menambahkan data periode pelatihan sesuai dengan *input* yang diberikan. Halaman ini hanya terdapat pada hak akses admin dan dikeluarkan saat menekan tombol *Add* pada halaman list periode pelatihan. Setelah admin memasukkan semua datanya, maka data akan dimasukkan ke dalam database. Semua input data pada halaman ini harus diisi sebelum dilakukan *submit*



Daftar Pengajar

Daftar Peserta

Daftar Orang Tua

Daftar Pelatihan

Periode

Logout

11 Juli 2024 pukul 18.10.21

Add New Periode

Tanggal Start

dd/mm/yyyy

Jenis Pelatihan

Kelompok

Kelas Paralel

Pengajar

Yantu

Pelatihan

Akuntansi

Jadwal Pertemuan

Hari

Pilih Hari Pertemuan

Waktu

00.00

-


00.00

Submit

Gambar 4.20 Halaman Tambah Periode

4.3.16 HALAMAN EDIT PERIODE

Halaman ini digunakan untuk melakukan perubahan pada data periode pelatihan. Saat masuk ke dalam halaman ini, data lama sudah terisi pada input di halaman edit periode pelatihan. Kemudian data tersebut akan dibawa ke dalam database untuk diubah nilainya. Semua data input perlu diisi dalam halaman ini sebelum melakukan proses submit.



Daftar Pengajar

Daftar Peserta

Daftar Orang Tua

Daftar Pelatihan

Periode

Logout

11 Juli 2024 pukul 18.10.44

Edit Periode

Tanggal Start

08/09/2023

Jenis Pelatihan

Kelompok

Kelas Paralel

A

Pengajar

Yantu

Pelatihan

Dasar Pemrograman

Jadwal Pertemuan

Hari

Pilih Hari Pertemuan

Waktu

00.00

-

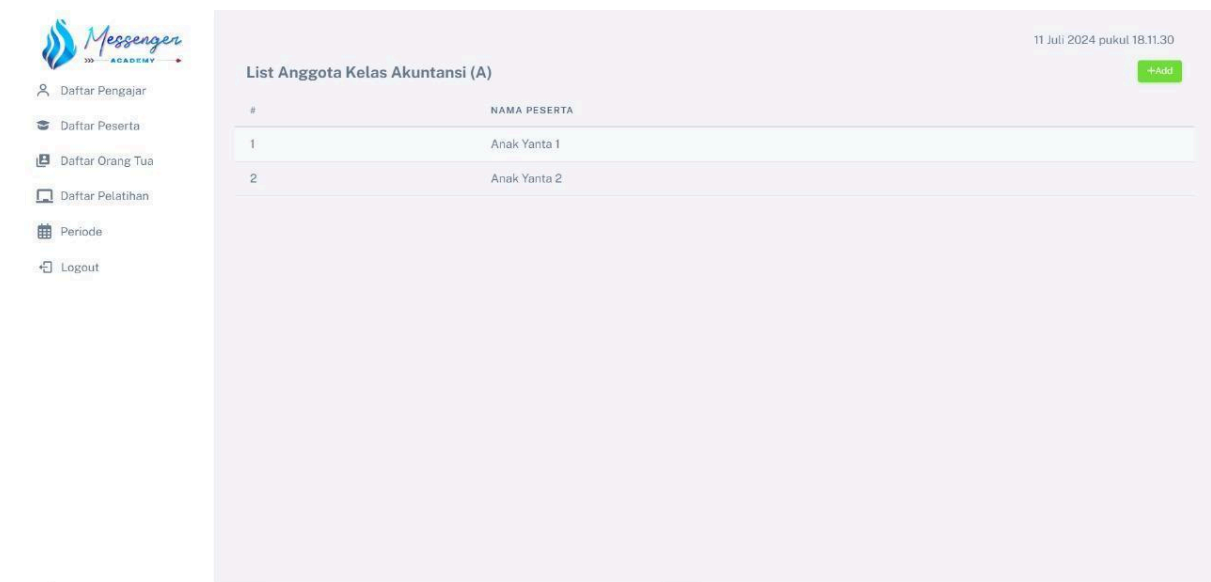
00.00

Submit

Gambar 4.21 Halaman Edit Periode

4.3.17 HALAMAN LIHAT ANGGOTA

Halaman ini digunakan untuk melihat setiap anggota yang tergabung dalam satu periode pelatihan. Halaman ini dapat melakukan penambahan data anggota kelas. Penambahan data anggota kelas ini dapat dilakukan dengan menekan tombol *Add*. Tombol ini akan menampilkan modal yang digunakan untuk memilih peserta yang akan dimasukkan ke dalam data anggota kelas.

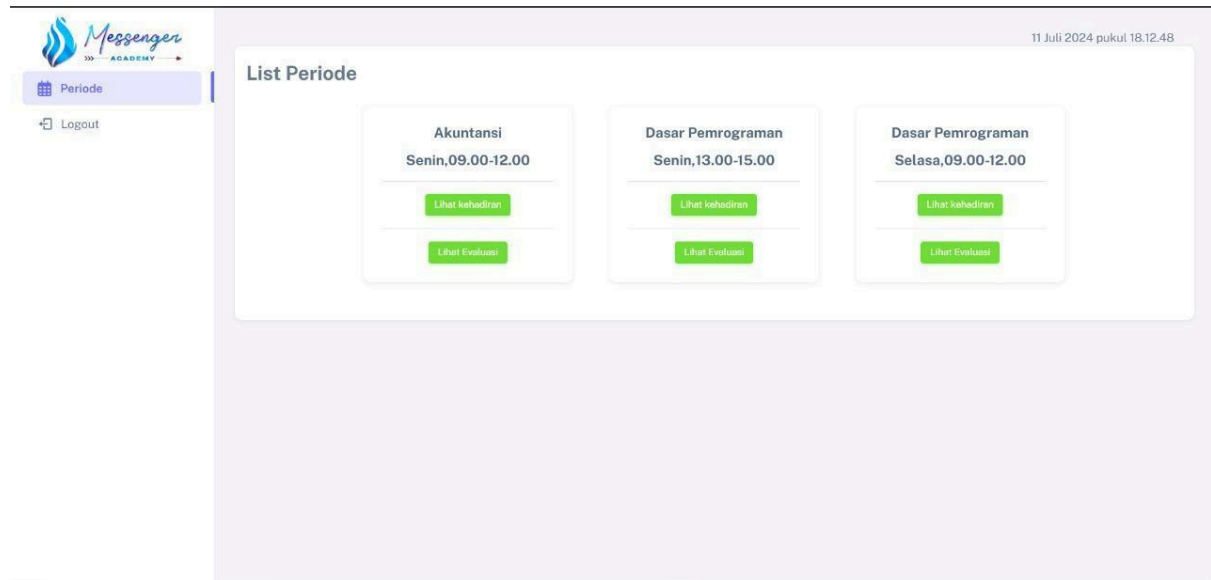


Gambar 4.22 Halaman Lihat Anggota

4.3.18 HALAMAN LIST PERIODE

Halaman ini digunakan untuk menampilkan data list periode pelatihan. Halaman ini merupakan list periode pelatihan untuk pengajar. Data yang ditampilkan disini hanya data periode pelatihan yang diajarkan oleh pengajar yang sedang mengakses halaman ini. Kemudian

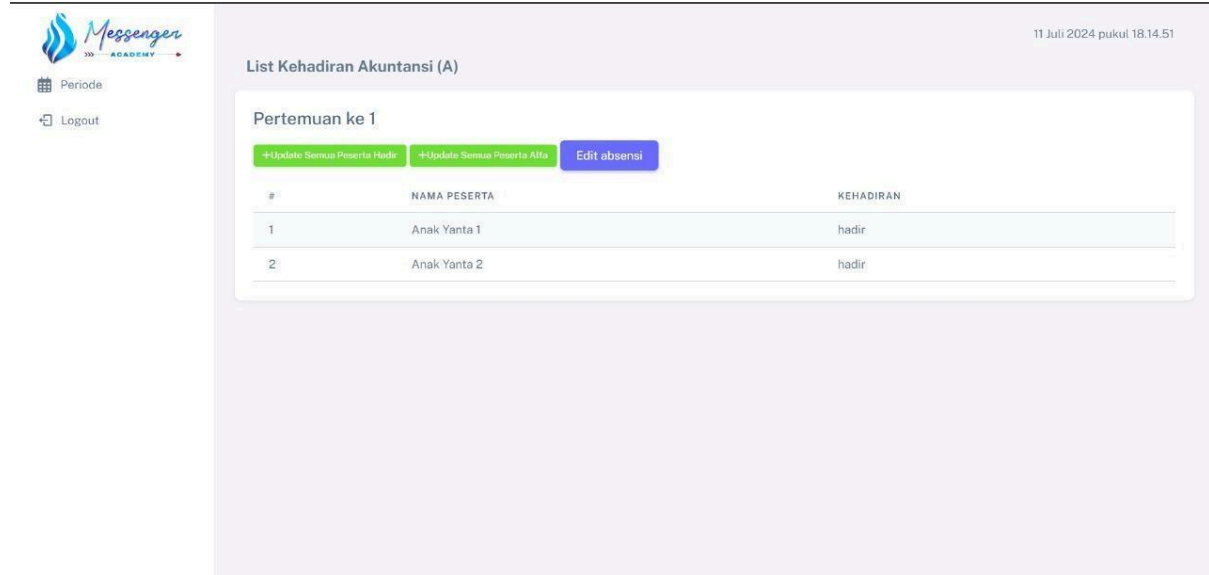
pengajar dapat melihat absensi dan melihat laporan yang pada kedua halaman tersebut nantinya dapat dilakukan perubahan data.



Gambar 4.23 Halaman List Periode

4.3.19 HALAMAN LIST KEHADIRAN

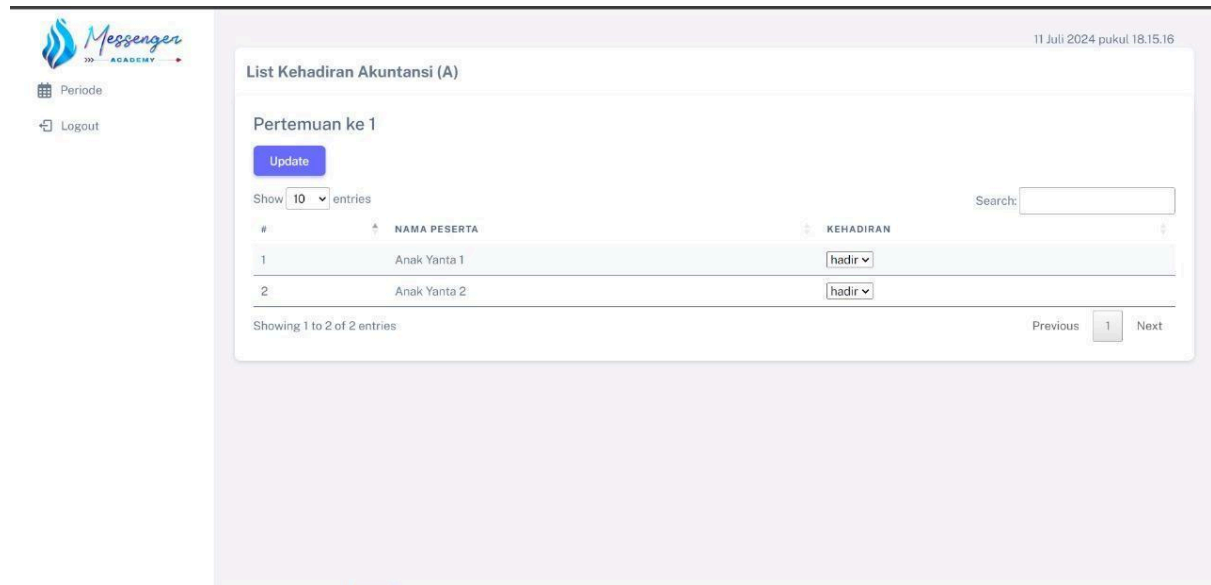
Halaman ini merupakan halaman yang dibuka melalui halaman list periode pelatihan dengan hak akses pengajar. Halaman ini menampilkan semua data absensi dalam setiap jadwal kelas. Halaman ini juga dapat digunakan untuk mengubah data semua peserta menjadi alfa semua ataupun hadir semua. Kemudian apabila ingin mengubah data peserta satu persatu dapat menekan tombol Edit Absensi.



Gambar 4.24 Halaman List Kehadiran

4.3.20 HALAMAN EDIT KEHADIRAN

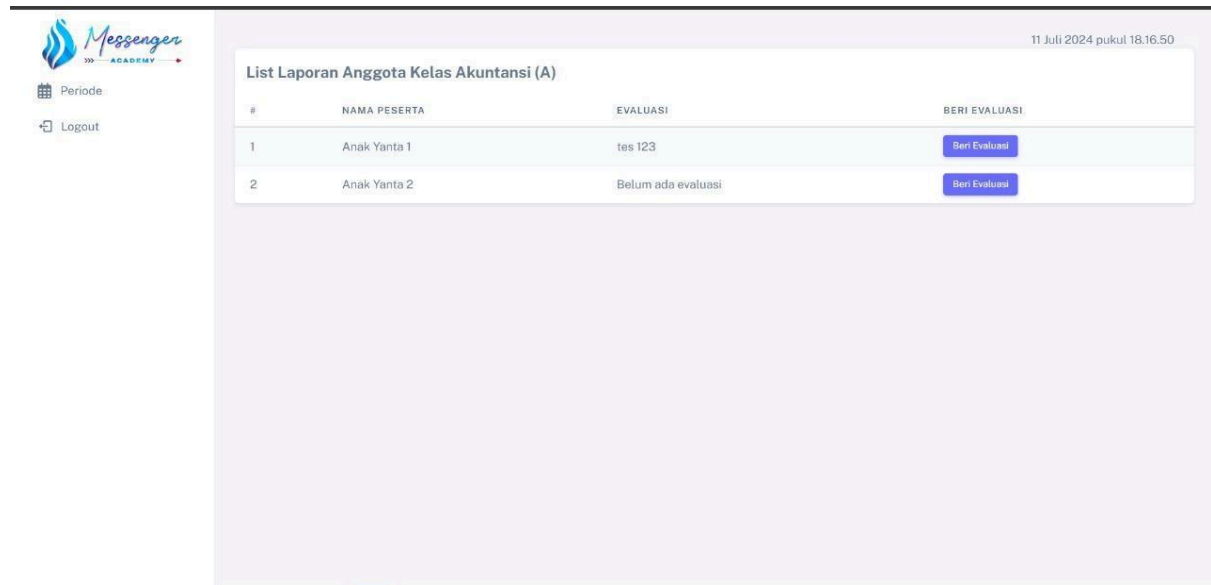
Halaman ini digunakan untuk melakukan perubahan pada data kehadiran peserta pada masing-masing pesertanya. Pengguna bisa mengubah data kehadiran dengan menekan *combo box* pada kolom kehadiran dan memilih nilai yang benar. Setelah semua data sudah diubah, maka perlu ditekan tombol update, maka semua data akan diubah dengan data yang baru. Untuk memudahkan pengisian disarankan mengupdate semua data menjadi hadir terlebih dahulu pada halaman sebelumnya dan setelah itu dilakukan proses pengeditan setiap anggota yang tidak hadir dengan beberapa alasan (sakit, ijin, dan alfa).



Gambar 4.24 Halaman Edit Kehadiran

4.3.21 HALAMAN EVALUASI

Halaman ini merupakan halaman yang dibuka melalui halaman list periode pelatihan dengan hak akses pengajar. Halaman ini menampilkan semua data laporan dalam setiap periode kelas. Halaman ini menunjukkan data evaluasi dari setiap anggota kelas jika sudah diubah, jika belum maka muncul belum ada evaluasi. Kemudian apabila ingin mengubah data peserta satu persatu dapat menekan tombol Edit Evaluasi.



Gambar 4.25 Halaman Evaluasi

4.3.22 HALAMAN EDIT EVALUASI

Halaman ini digunakan untuk melakukan perubahan pada data eval. Saat masuk ke dalam halaman ini, data lama sudah terisi pada input di halaman edit periode pelatihan. Kemudian data tersebut akan dibawa ke dalam database untuk diubah nilainya. Semua data input perlu diisi dalam halaman ini sebelum melakukan proses submit.

Messenger 100 ACADEMY

11 Juli 2024 pukul 18.17.09

Edit Evaluasi

Pelatihan: Akuntansi (A)

Nama: Anak Yanta1

Evaluasi

tes 123

Submit


Periode

Logout

Gambar 4.26 Halaman Edit Evaluasi

4.3.23 HALAMAN LIHAT PERIODE

Halaman ini digunakan untuk melihat periode pelatihan yang diikuti oleh anak dari orang tua (peserta) yang orang tuanya adalah pemilik akun yang telah login. Halaman ini dimiliki oleh hak akses orang tua. Pada setiap datanya terdapat tombol untuk melihat kehadiran dan laporannya. Periode dibagi menjadi periode pelatihan yang sedang berjalan maupun sudah selesai.



Periode
 Logout

11 Juli 2024 pukul 17:48:26

Data Pelatihan Anak Yanta 1

Berjalan

NAMA PELATIHAN	JADWAL	LIHAT KEHADIRAN	LIHAT EVALUASI
Akuntansi	Senin,09.00-12.00	Lihat Kehadiran	Lihat Evaluasi
Dasar Pemrograman	Senin,13.00-15.00	Lihat Kehadiran	Lihat Evaluasi

Selesai

NAMA PELATIHAN	JADWAL	LIHAT KEHADIRAN	LIHAT EVALUASI
----------------	--------	-----------------	----------------

Data Pelatihan Anak Yanta 2

Berjalan

NAMA PELATIHAN	JADWAL	LIHAT KEHADIRAN	LIHAT EVALUASI
Dasar Pemrograman	Senin,13.00-15.00	Lihat Kehadiran	Lihat Evaluasi
Dasar Pemrograman	Selasa,09.00-12.00	Lihat Kehadiran	Lihat Evaluasi
Psikologi Dewasa	Selasa,13.00-15.00	Lihat Kehadiran	Lihat Evaluasi

Selesai

NAMA PELATIHAN	JADWAL	LIHAT KEHADIRAN	LIHAT EVALUASI
CC	Selasa,10.00-13.00	Lihat Kehadiran	Lihat Evaluasi

Gambar 4.27 Halaman Lihat Periode

4.3.24 HALAMAN LIHAT KEHADIRAN PESERTA

Halaman ini menampilkan status kehadiran suatu peserta dalam suatu periode pelatihan yang diambil dari tombol lihat kehadiran pada halaman lihat periode dari hak akses orang tua diatas. Disini ditunjukkan nomor pertemuan dan status kehadirannya dalam periode pelatihan tersebut. Untuk menjelaskan lebih detail, maka ditambahkan informasi nama peserta dan nama pelatihan, serta kelas paralel dari periode pelatihan tersebut pada judul halamannya. Halaman ini dimiliki oleh hak akses orang tua.



Gambar 4.28 Halaman Lihat Kehadiran Peserta

4.3.25 HALAMAN EVALUASI PESERTA

Halaman ini menampilkan status kehadiran suatu peserta dalam suatu periode pelatihan yang diambil dari tombol lihat kehadiran pada halaman lihat periode dari hak akses orang tua diatas. Disini ditunjukkan data evaluasi dari periode pelatihan tersebut. Untuk menjelaskan lebih detail, maka ditambahkan informasi nama peserta dan nama pelatihan, serta kelas paralel dari periode pelatihan tersebut pada judul halamannya. Halaman ini dimiliki oleh hak akses orang tua.



Gambar 4.29 Halaman Evaluasi Peserta

BAB 5

IMPLEMENTASI

Bab ini akan membahas mengenai implementasi hasil desain sistem informasi presensi dan evaluasi pada lembaga kursus Messenger Academy Surabaya yang sudah didesain pada Bab ini. Implementasi ini meliputi program yang menerapkan hasil dari desain yang sudah dibuat tersebut dari penggunaan database dalam proses aplikasi sebagai implementasi dari *Entity Relationship Diagram (ERD)*.

5.1 IMPLEMENTASI ENTITY RELATIONSHIP DIAGRAM (ERD)

5.1.1 TABEL USERS

Tabel ini berisi nama setiap user dari beberapa hak akses yang memerlukan proses autentikasi seperti admin, pengajar, dan orang tua. Untuk admin hanya dapat dilakukan proses login, sedangkan pengajar dan orang tua dapat dilakukan proses *read*, *create*, dan *update*. Berikut merupakan potongan kode untuk melakukan setiap prosesnya.

Read:

Pengajar:

```
public function index()
{
    $pengajar = DB::table('users')
        ->select('users.*')
        ->where('role', 'pengajar')
```

```

->get();

return view('pengajar.index', compact('pengajar'));
}

```

Orang tua:

```

public function index()
{
    $orangtua = DB::table('users')
        ->select('users.*')
        ->where('role', orang_tua)
        ->get();

    return view('pengajar.index', compact(orangtua));
}

```

Pada potongan kode diatas, juga berlaku bagi hak akses orang tua dimana pada bagian *where*, pengajar dapat digantikan dengan orang tua. Kode ini berguna untuk mengambil semua data yang ada pada tabel *users* sesuai dengan hak akses (*role*) yang diperlukan untuk dicari. Setelah semua data diambil, maka akan dimasukkan ke dalam variabel bernama “pengajar” atau “orang tua: jika hak akses yang dicari adalah orang tua. Setelah itu data akan dikirimkan ke dalam website untuk ditampilkan datanya.

Create:

Pengajar:

```
public function store(Request $request)
{
    $validator = Validator::make($request->all(), [
        'password' => 'required|min:8',
    ], [
        'password.required' => 'Password harus diisi.',
        'password.min' => 'Password minimal harus terdiri dari 8 karakter.',
    ]);

    if ($validator->fails()) {
        return redirect()->back()->withErrors($validator)->withInput();
    }

    $user = new Users();

    $user->password = Hash::make($request->get('password'));

    $user->nama = $request->get('nama');
    $user->alamat = $request->get('alamat');
    $user->email = $request->get('email');
    $user->role = "pengajar";
```

```

$user->status = "1";

$user->created_at = now("Asia/Bangkok");

$user->updated_at = now("Asia/Bangkok");

$user->save();

return redirect()->route('pengajar.index')->with('status', 'New pengajar ' .
$user->nama . ' is already inserted');
}

```

Orang tua:

```

public function store(Request $request)
{
    $validator = Validator::make($request->all(), [
        'password' => 'required|min:8',
    ], [
        'password.required' => 'Password harus diisi.',
        'password.min' => 'Password minimal harus terdiri dari 8 karakter.',
    ]);

    if ($validator->fails()) {
        return redirect()->back()->withErrors($validator)->withInput();
    }
}

```



```

$users = new Users();

$users->nama = $request->get('nama');

$users->alamat = $request->get('alamat');

$users->email = $request->get('email');

$users->password = $request->get('password');

$users->role = "orang_tua";

$users->status = 1;

$users->created_at = now("Asia/Bangkok");

$users->updated_at = now("Asia/Bangkok");

$users->save();

return redirect()->route('orangtua.index')->with('status', 'Orangtua ' . $users->nama
. ' berhasil ditambahkan');

}

```

Kedua fungsi ini sama-sama melakukan proses *create* ke dalam database. Data yang diambil dari *request* dari halaman *create* dari tabel terkait. Program akan membuat *user* baru untuk dimasukkan ke dalam *database*. *User* baru itu akan menerima semua data yang diperlukan ke dalam *database* dan kemudian disimpan kedalam *database*. Setelah proses simpan selesai, maka akan dikembalikan ke halaman *index* dengan status yang dikeluarkan.

Update:

Pengajar:

```

public function update(Request $request, $id)
{
    $user = Users::find($id);

    $user->nama = $request->get('nama');
    $user->alamat = $request->get('alamat');
    $user->save();

    return redirect()->route('pengajar.index')->with('status', 'pengajar ' . $user->nama . '
is already updated');
}

```

Orang tua:

```

public function update(Request $request, $id)
{
    $users = Users::where('id', $id)->first();

    $users->nama = $request->get('nama');
    $users->alamat = $request->get('alamat');
    $users->save();

    return redirect()->route('orangtua.index')->with('status', 'Orangtua ' . $users->nama
. ' berhasil diupdate');
}

```

```
}
```

Fungsi ini mengimplementasikan perubahan data ke dalam database dimana tabel users data akan di update dimana id sesuai dengan yang dipilih. Kemudian, nama dan alamat kedua role tersebut akan diupdate berdasarkan data yang diambil dari form yang mengeluarkan *request* update. Setelah itu data disimpan ke dalam database users sesuai dengan rolenya. Kemudian request untuk memanggil read yang sudah dijelaskan diatas.

5.1.2 TABEL PESERTA

Tabel peserta merupakan tabel yang aksesnya menunjukkan data-data dari setiap peserta dan relasinya dengan tabel orang tua. Berikut adalah potongan kode untuk proses *read*, *create*, dan *update*.

Read

```
public function index($id_orangtua = null)
{

    $peserta = DB::table('peserta')
    ->join('users', 'users.id','=', 'peserta.id_orangtua')
    ->select('peserta.*', 'users.nama as namaorangtua')
    ->get();

    return view('peserta.index', compact('peserta'));
}
```

Read ini mengambil seluruh data pada peserta dan juga data nama orang tua yang ada pada tabel users dengan role orang tua. Penggabungan tabel terdapat ketentuan dimana id_orangtua pada tabel peserta sama dengan id pada tabel users tersebut. Hal ini digunakan untuk memperoleh data orang tua dari setiap peserta tersebut. Kemudian data peserta dimasukkan ke dalam view yang untuk ditampilkan datanya.

Create

```
public function store(Request $request)
{

    $peserta = new Peserta();

    $peserta->nama = $request->get('nama');

    $peserta->umur = $request->get('umur');

    $peserta->created_at = now("Asia/Bangkok");

    $peserta->updated_at = now("Asia/Bangkok");

    $peserta->id_orangtua = $request->get('id_orangtua');

    $peserta->save();

    return redirect()->route('peserta.index')->with('message', 'Peserta ' . $peserta->nama
    . ' berhasil ditambahkan');

}
```

Fungsi ini melakukan proses *create* ke dalam tabel peserta pada database. Data yang diambil dari *request* dari halaman *create* dari tabel peserta dengan id orang tua yang diambil akan membuat peserta ini terikat pada relasi one-to-many dimana 1 peserta hanya dapat memiliki 1 orang tua dan 1 orang tua dapat memiliki banyak anak. Program akan membuat *peserta* baru untuk dimasukkan ke dalam *database*. *Peserta* baru itu akan menerima semua data yang diperlukan ke dalam *database* dan kemudian disimpan kedalam *database*. Setelah proses simpan selesai, maka akan dikembalikan ke halaman *index* dengan status yang dikeluarkan.

Update

```
public function update(Request $request, $id)
{
    $peserta = Peserta::find($id);

    $peserta->nama = $request->get('nama');
    $peserta->umur = $request->get('umur');
    $peserta->id_orangtua = $request->get('id_orangtua');
    $peserta->updated_at = now("Asia/Bangkok");
    $peserta->save();

    return redirect()->route('peserta.index')->with('message', 'Peserta ' .
    $peserta->nama . ' berhasil diupdate');
}
```

Fungsi ini melakukan proses *update* ke dalam tabel peserta pada database dengan id yang diambil melalui *request*. Kemudian program akan mencari id peserta yang ingin dilakukan proses edit. Setelah itu, program akan mengambil semua request yang ada dan seperti sebelumnya, id orang tua juga akan diambil. Setelah proses *update* selesai, maka akan dikembalikan ke halaman *index* dengan status yang dikeluarkan.

5.1.3 TABEL LAPORAN

Tabel laporan merupakan tabel yang menghubungkan keikutsertaan peserta dengan suatu periode pelatihan dan evaluasinya. Pada halaman admin, laporan ini digunakan untuk melihat data peserta-peserta yang ada di setiap kelasnya dan menambahkan peserta ke dalam suatu kelas. Namun pada halaman pengajar, laporan ini juga digunakan untuk melihat data peserta-peserta yang ada di setiap kelas yang diajar oleh pengajar tersebut, namun pengajar juga dapat memberikan evaluasi ke setiap anggota kelasnya. Sedangkan untuk orang tua, digunakan untuk melihat evaluasi anak-anaknya (peserta) dari setiap pelatihan yang diikutinya. Berikut adalah potongan kode untuk proses *read*, *create* dan *update*.

Read

```
public function daftarpeserta($idperiode)
{
    if (Auth::user()->role == "admin") {
        $anggota_kelas = DB::table('laporan')
            ->join('periode', 'laporan.idperiode', '=', 'periode.id')
            ->join('peserta', 'laporan.id_peserta', '=', 'peserta.id')
```

```

->join('pelatihan', 'periode.idpelatihan', '=', 'pelatihan.id')

->select('laporan.*', 'pelatihan.nama as namapelatihan', 'periode.kelas_paralel
as kelasparalel',

'periode.id as idperiode', 'peserta.nama as namapeserta')

->where('periode.id'='',$idperiode)

->get();

//dd($anggota_kelas);

$peserta = DB::table("peserta")

->whereNotIn('id', function($query) use ($idperiode) {

$query->select('id_peserta')

->from('laporan')

->where('idperiode', $idperiode);

})

->select('id as idpeserta', 'nama as namapeserta')

->get();

return view('laporan.index', ['anggota_kelas'=>$anggota_kelas,
'peserta'=>$peserta]);

}

else if (Auth::user()->role == "pengajar") {

$anggota_kelas = DB::table('laporan')

->join('periode', 'laporan.idperiode', '=', 'periode.id')

```

```

->join('peserta', 'laporan.id_peserta', '=', 'peserta.id')

->join('pelatihan', 'periode.idpelatihan', '=', 'pelatihan.id')

->select('laporan.*', 'pelatihan.nama as namapelatihan', 'periode.kelas_paralel
as kelasparalel',

        'periode.id as idperiode', 'peserta.nama as namapeserta', 'peserta.id as
idpeserta')

->where('periode.id', '=', $idperiode)

->get();

return view('laporan.index', ['anggota_kelas'=>$anggota_kelas]);
}else if (Auth::user()->role == "orang_tua") {

$evaluasi = DB::table('laporan')

->join('periode', 'laporan.idperiode', '=', 'periode.id')

->join('peserta', 'laporan.id_peserta', '=', 'peserta.id')

->join('pelatihan', 'periode.idpelatihan', '=', 'pelatihan.id')

->select('laporan.*', 'pelatihan.nama as namapelatihan', 'periode.kelas_paralel
as kelasparalel',

        'periode.id as idperiode', 'peserta.nama as namapeserta', 'peserta.id as
idpeserta', 'laporan.evaluasi as eval')

->where('periode.id', '=', $idperiode)

->where('peserta.id_orangtua', '=', Auth::id())

->get();

return view('laporan.index', ['evaluasi'=>$evaluasi]);

}

```



```
}
```

Fungsi ini berguna untuk mengambil data laporan setiap anggota kelas dari suatu periode,. Tabel periode merupakan tabel yang menyatakan satu periode dari suatu pelatihan. Untuk setiap hak akses, terdapat ketentuan yang berbeda-beda untuk data yang diambil. Untuk hak akses admin, diambil semua anggota kelas beserta nama pelatihan, nama peserta, dan kelas paralel dan juga dicari peserta yang belum ada di periode itu. Untuk hak akses pengajar, yang dicari adalah hampir sama seperti hak akses admin namun tidak diambil peserta yang belum ada di periode itu. Sedangkan untuk hak akses orang tua terdapat tambahan satu parameter *where* yaitu untuk mencari hanya id peserta yang memiliki id orang tua sama dengan data id login saat ini. Kemudian ditampilkan ke dalam halaman lainnya.

Create

```
public function store(Request $request)
```

```
{
```

```
    DB::statement("INSERT INTO laporan (id_peserta, idperiode, evaluasi) VALUES  
(?,?,?)",
```

```
[
```

```
    $request->get('id_peserta'),
```

```
    $request->get('idperiode'),
```

```
    "
```

```
]);
```

```
return redirect()->route('laporan.daftarpeserta', [
```

```

        'id'=>$request->get('idperiode'),
        'message' => 'Data berhasil ditambahkan'
    );
}

```

Fungsi ini digunakan untuk menambahkan data laporan baru ke dalam database. Fungsi ini dalam sistem digunakan untuk menambah data peserta baru ke dalam suatu periode. Yang diambil dari fungsi ini adalah data id periode dan id peserta. Evaluasi dibiarkan berupa string kosong untuk nantinya di update oleh pengajar saat pengajar menuliskan evaluasi kepada peserta dalam suatu periode. Setelah data itu dimasukkan, maka akan dikembalikan ke list laporan atau dalam kasus admin list anggota.

Update

```

public function updateevaluasi(Request $request)
{
    $new_evaluasi = $request->get('evaluasi');
    $id_peserta = $request->get('idpeserta');
    $idperiode = $request->get('idperiode');

    DB::statement("UPDATE laporan SET evaluasi = ? WHERE id_peserta = ? AND
idperiode = ?", [
        $new_evaluasi, $id_peserta, $idperiode]);
    return redirect()->route('laporan.daftarpeserta',$idperiode);
}

```

Fungsi ini digunakan oleh pengajar untuk mengubah evaluasi dari setiap peserta. Data yang digunakan dalam fungsi ini adalah evaluasi, id peserta, dan id periode. Kemudian dilakukan proses update dimana id periode dan id peserta sesuai dengan data yang digunakan. Setelah itu dikembalikan ke dalam fungsi read

5.1.4 TABEL PERIODE

Tabel periode merupakan tabel yang menyatakan satu periode dari suatu pelatihan. Tabel ini memiliki relasi one-to-many dengan tabel pelatihan, laporan dan jadwal kelas. Tabel ini nantinya akan digunakan untuk menyimpan jadwal reguler setiap pelatihan dan jadwal kelas yang digunakan untuk proses absensi. Berikut adalah potongan kode untuk proses *read*, *create*, dan *update*.

Read

```
public function index()
{
    if(Auth::user()->role == 'admin'){
        $periode = DB::table('periode')
            ->join('pelatihan','periode.idpelatihan', '=', 'pelatihan.id')
            ->join('users','periode.id_pengajar', '=', 'users.id')
            ->select('periode.*', 'pelatihan.nama as namapelatihan',
                'pelatihan.jumlah_pertemuan as jumlahpertemuan',
                'users.nama as namapengajar')
            ->where('periode.status','=','berjalan')
```

```

->get();

return view('periode.index', ["periode" => $periode]);
}

elseif(Auth::user()->role == 'pengajar'){

    $periode = DB::table('periode')

    ->join('pelatihan','periode.idpelatihan', '=' , 'pelatihan.id')

    ->join('users','periode.id_pengajar', '=' , 'users.id')

        ->select('periode.*', 'pelatihan.nama as namapelatihan',
'pelatihan.jumlah_pertemuan as jumlahpertemuan',

        'users.nama as namapengajar')

    ->where('periode.status','=','berjalan')

    ->where('periode.id_pengajar','=',Auth::id())

    ->get();

    return view('periode.index', ["periode" => $periode]);
}

elseif(Auth::user()->role == 'orang_tua'){

    $peserta = DB::table('peserta')

    ->select('*')

    ->where('id_orangtua','=', Auth::id())

    ->get();


    $periodepeserta = [];

    foreach($peserta as $key=>$value){

```

```

$periodejalan = DB::table('peserta')

->join('laporan','laporan.id_peserta', '=', 'peserta.id')

->join('periode','laporan.idperiode', '=', 'periode.id')

->join('pelatihan','periode.idpelatihan', '=', 'pelatihan.id')

->select('periode.*' , 'pelatihan.nama as namapelatihan',
'pelatihan.jumlah_pertemuan as jumlahpertemuan',
'peserta.nama as namapeserta', 'peserta.id as idpeserta', 'periode.id as
idperiode')

->where('periode.status','=','berjalan')

->where('peserta.id','=',$value->id)

->get();

$periodeselesai = DB::table('peserta')

->join('laporan','laporan.id_peserta', '=', 'peserta.id')

->join('periode','laporan.idperiode', '=', 'periode.id')

->join('pelatihan','periode.idpelatihan', '=', 'pelatihan.id')

->select('periode.*' , 'pelatihan.nama as namapelatihan',
'pelatihan.jumlah_pertemuan as jumlahpertemuan',
'peserta.nama as namapeserta', 'peserta.id as idpeserta', 'periode.id as
idperiode')

->where('periode.status','=','selesai')

->where('peserta.id','=',$value->id)

->get();

```

```

array_push($periodepeserta,
['periodejalan'=>$periodejalan,'periodeselesai'=>$periodeselesai]);

    }

    return view('periode.index', ['peserta'=>$peserta, "periode" => $periodepeserta]);
}
}

```

Fungsi read ini digunakan untuk mengambil data periode pelatihan dari database. Data yang diambil berbeda untuk setiap role. Admin melihat semua periode pelatihan. Pengajar melihat periode pelatihan yang diajarkan. Sedangkan orang tua dapat mengambil data untuk melihat data periode pelatihan yang diikuti anaknya saja..

Create

```

public function store(Request $request)
{
    for ($i = 0; $i<=$request->get('jumlahpertemuan');$i++) {
        $data = [
            'hari'=>$request->get('hari_pertemuan_'.$i),
            'waktu_awal'=>$request->get('waktu_awal_pertemuan_'.$i),
            'waktu_akhir'=>$request->get('waktu_akhir_pertemuan_'.$i),
        ];
    }
}

```

```

$periode = new Periode();

$periode->tanggal_start = $request->get('tanggal_start');

$periode->jenis_pelatihan = $request->get('jenis_pelatihan');

$periode->status = "berjalan";

$periode->jadwal = $data['hari'].', '.$data['waktu_awal'].', '.$data['waktu_akhir'];

$periode->kelas_paralel = $request->get('kelas_paralel');


$periode->created_at = now("Asia/Bangkok");

$periode->updated_at = now("Asia/Bangkok");


$periode->id_pengajar = $request->get('id_pengajar');

$periode->idpelatihan = $request->get('idpelatihan');

$periode->save();

}


return redirect()->route('periode.index')->with('message', 'Periode berhasil
ditambahkan'. $request->get('jumlahpertemuan'));

}

```

Tabel periode merupakan tabel yang menyatakan satu periode dari suatu pelatihan. Tabel periode merupakan tabel yang menyatakan satu periode dari suatu pelatihan.

Update

```
public function update(Request $request, $id)
{
    $periode = Periode::find($id);

    $periode->status = "berjalan";

    $periode->jadwal =

    $request->get('hari_pertemuan').', '.$request->get('waktu_awal_pertemuan').'-'.
        $request->get('waktu_akhir_pertemuan');

    $periode->updated_at = now("Asia/Bangkok");

    $periode->save();

    return redirect()->route('periode.index')->with('status', 'Periode berhasil diupdate');
}
```

Tabel periode merupakan tabel yang menyatakan satu periode dari suatu pelatihan. Tabel periode merupakan tabel yang menyatakan satu periode dari suatu pelatihan.

5.1.5 TABEL PELATIHAN

Tabel pelatihan adalah tabel yang digunakan untuk melihat pelatihan yang ada. Pelatihan ini dapat digunakan untuk beberapa periode dengan cara mengambil id dari pelatihan ini. Berikut adalah potongan kode untuk proses *read*, *create*, dan *update*.

Read:


```

public function index()
{

    $pelatihan = DB::table('pelatihan')
        ->select('*')
        ->get();

    return view('pelatihan.index', ["pelatihan" => $pelatihan]);
}

```

Proses *read* pada tabel ini memiliki berguna untuk menampilkan semua data pelatihan. Proses *read* pada pelatihan ini akan mengambil data nama, deskripsi, dan jumlah pertemuan pada pelatihan. Data pelatihan ini hanya dimiliki oleh hak akses admin. Data pelatihan ini akan dibawa ke dalam halaman list peserta pada admin.

Create

```

public function store(Request $request)
{
    $pelatihan = new Pelatihan();
    $pelatihan->nama = $request->get('nama');
    $pelatihan->deskripsi = $request->get('deskripsi');
    $pelatihan->jumlah_pertemuan = $request->get('jumlah_pertemuan');

    $pelatihan->created_at = now("Asia/Bangkok");
}

```

```

$pelatihan->updated_at = now("Asia/Bangkok");

$pelatihan->save();

return redirect()->route('pelatihan.index')->with('status', 'Pelatihan ' .
$pelatihan->nama . ' berhasil ditambahkan');

}

```

Fungsi ini digunakan untuk menambahkan data pelatihan me dalam database. Input yang diambil dari fungsi ini adalah nama, deskripsi, dan jumlah pertemuan dalam pelatihan itu. Kemudian data tersebut akan dimasukkan ke dalam database. Setelah dimasukkan, maka akan dikembalikan ke halaman list pelatihan.

Update:

```

public function update(Request $request, $id)
{
    $pelatihan = Pelatihan::find($id);

    $pelatihan->nama = $request->get('nama');

    $pelatihan->deskripsi = $request->get('deskripsi');

    $pelatihan->jumlah_pertemuan = $request->get('jumlah_pertemuan');

    $pelatihan->updated_at = now("Asia/Bangkok");

    $pelatihan->save();
}

```

```

        return redirect()->route('pelatihan.index')->with('status', 'Pelatihan ' .
$pelatihan->nama . ' berhasil diupdate');
    }

```

Update digunakan untuk mengubah data pelatihan yang sudah ada. Pertama sistem akan mencari id pelatihan yang sesuai dengan *request*, kemudian akan mengupdate data sesuai dengan request dari pengguna. Setelah itu akan dikembalikan ke halaman list peserta.

5.1.6 TABEL JADWAL KELAS

Tabel jadwal kelas merupakan tempat menyimpan semua jadwal kelas yang dibuka. Tabel ini merupakan tempat menyimpan absensi-absensi yang pernah dibuka sebelumnya. Berikut adalah potongan kode untuk proses *create*.

Create

```

public function store(Request $request)
{
    $curr_date = Carbon::now()->locale('id')->toDateString();

    $jadwal_kelas = new JadwalKelas();
    $jadwal_kelas->jenis_pertemuan = $request->post("jenis_pertemuan");
    $jadwal_kelas->idperiode = $request->post("idperiode");
    $jadwal_kelas->nomor_pertemuan = $request->post("nomor_pertemuan");

    $cek_tanggal = DB::table('jadwal_kelas')

```

```
->select('*')  
  
->where("idperiode", "=", $jadwal_kelas->idperiode)  
  
->where("tanggal_absensi", "=", $curr_date)  
  
->get();
```

```
if ($cek_tanggal->count() == 0) { //kalau hari ini belum buka presensi
```

```
    $pengikut_kelas = DB::table('laporan')  
  
    ->select("*")  
  
    ->where("idperiode", "=", $jadwal_kelas->idperiode)  
  
    ->get();  
  
    $jadwal = new JadwalKelas();  
  
    $jadwal->nomor_pertemuan = $jadwal_kelas->nomor_pertemuan;  
  
    $jadwal->jenis_pertemuan = $jadwal_kelas->jenis_pertemuan;  
  
    $jadwal->tanggal_absensi = $curr_date;  
  
    $jadwal->idperiode = $jadwal_kelas->idperiode;  
  
    $jadwal->save();  
  
    $idjadwal = $jadwal->id;  
  
    foreach ($pengikut_kelas as $pengikut) {  
  
        $absensi = new Absensi();  
  
        $absensi->status_kehadiran = "alfa";  
  
        $absensi->idjadwalkelas = $idjadwal;  
  
        $absensi->id_peserta = $pengikut->id_peserta;
```

```

        $absensi->save();
    }

    $message = "Absensi berhasil dibuka";

    return redirect()->route("periode.index")->with("message", $message);
}

else{

    return redirect()->route("periode.index")->with("error", "Anda sudah membuka
presensi hari ini");

}

}

```

Fungsi ini berguna untuk menambahkan data jadwal kelas baru. Pertama akan mengambil semua data dari *request* ditambah dengan data tanggal hari ini. Setelah itu akan dilakukan pengecekan adanya jadwal kelas dari periode yang sama dalam hari ini. Jika ada, maka batalkan proses *create* (buka absensi), jika belum ada, maka pakai semua data yang sudah diambil dan masukkan ke dalam tabel absensi. Kemudian masukkan absensi untuk setiap anggota kelas yang diambil dari tabel laporan dengan status kehadiran alfa. Setelah itu akan dikembalikan ke tabel list periode kelas.

5.1.7 TABEL ABSENSI

Tabel absensi merupakan tabel yang menyimpan data presensi dari setiap peserta dari setiap jadwal kelas yang dibuka. Tabel ini akan menyimpan status kehadiran dari setiap peserta. Berikut adalah potongan kode untuk proses *read*, dan *update*

Read:

```
public function lihat_absensi($id, $idpeserta = null){
    if(Auth::user()->role == 'pengajar'){
        $peserta = DB::table('absensi')
            ->join('jadwal_kelas','jadwal_kelas.id','=', 'absensi.idjadwalkelas')
            ->join('periode','jadwal_kelas.idperiode','=', 'periode.id')
            ->join('pelatihan','pelatihan.id','=', 'periode.idpelatihan')
            ->join('peserta','absensi.id_peserta','=', 'peserta.id')
            ->select('absensi.*', 'pelatihan.nama as namapelatihan', 'periode.kelas_paralel as
kelasparalel',
                'periode.id as idperiode', 'peserta.nama as namapeserta', 'peserta.id as idpeserta')
            ->where('periode.id','=', $id)
            ->get();

        $jadwalkelas = DB::table('jadwal_kelas')
            ->select('jadwal_kelas.*')
            ->where('idperiode','=', $id)
            ->get();

        return view ('absensi.index',["peserta"=>$peserta,'jadwalkelas'=>$jadwalkelas]);
    }
}
```

```

    }

    elseif(Auth::user()->role == 'orang_tua'){

        $peserta = DB::table('absensi')

        ->join('jadwal_kelas','jadwal_kelas.id','=', 'absensi.idjadwalkelas')

        ->join('periode','jadwal_kelas.idperiode','=', 'periode.id')

        ->join('pelatihan','pelatihan.id','=', 'periode.idpelatihan')

        ->join('peserta','absensi.id_peserta','=', 'peserta.id')

        ->select('absensi.*', 'pelatihan.nama as namapelatihan', 'periode.kelas_paralel as
kelasparalel',

        'periode.id as idperiode', 'peserta.nama as namapeserta', 'peserta.id as idpeserta',

        'jadwal_kelas.nomor_pertemuan as nomorpertemuan')

        ->where('idperiode','=',$id)

        ->where('peserta.id', '=', $idpeserta)

        ->get();

        return view ('absensi.index',['peserta'=>$peserta]);

    }

}

```

Fungsi ini digunakan untuk mengambil data absensi dari peserta. Data yang diambil merupakan data absensi peserta pada id periode tertentu. Pada hak akses pengajar, juga diambil data jadwal kelas untuk mengetahui semua jadwal yang sudah pernah dibuka dalam periode pelatihan tersebut. Pada bagian orang tua ditambahkan ketentuan dimana data yang dikeluarkan adalah milik peserta merupakan anak dari orang tua tersebut. Kemudian akan diarahkan ke halaman list kehadiran peserta.

Update

```
public function updatestatuskehadiran(Request $request)
{
    $peserta = $request->peserta;

    foreach ($peserta as $p) {
        $idpeserta = $p['idpeserta'];
        $status_kehadiran = $p['status_kehadiran'];

        DB::table('absensi')
            ->where('id_peserta', $idpeserta)
            ->where('idjadwalkelas', $request->get('idjadwalkelas'))
            ->update(['status_kehadiran' => $status_kehadiran]);
    }

    // return a success response
    return redirect()->back()->with('status', 'Status kehadiran berhasil diperbarui.');
```

Fungsi ini berguna untuk melakukan proses update kepada semua peserta yang ada pada kelas itu dengan data kehadiran yang dimasukkan kepada setiap peserta. Pertama akan diambil id peserta dan status kehadiran yang ditentukan kepada masing-masing anggota kelas. Kemudian

akan dilakukan proses *update* kepada semua data yang ada pada absensi. Setelah itu akan dikembalikan halaman dimana data ini dikirimkan.

BAB 6

UJI COBA DAN EVALUASI

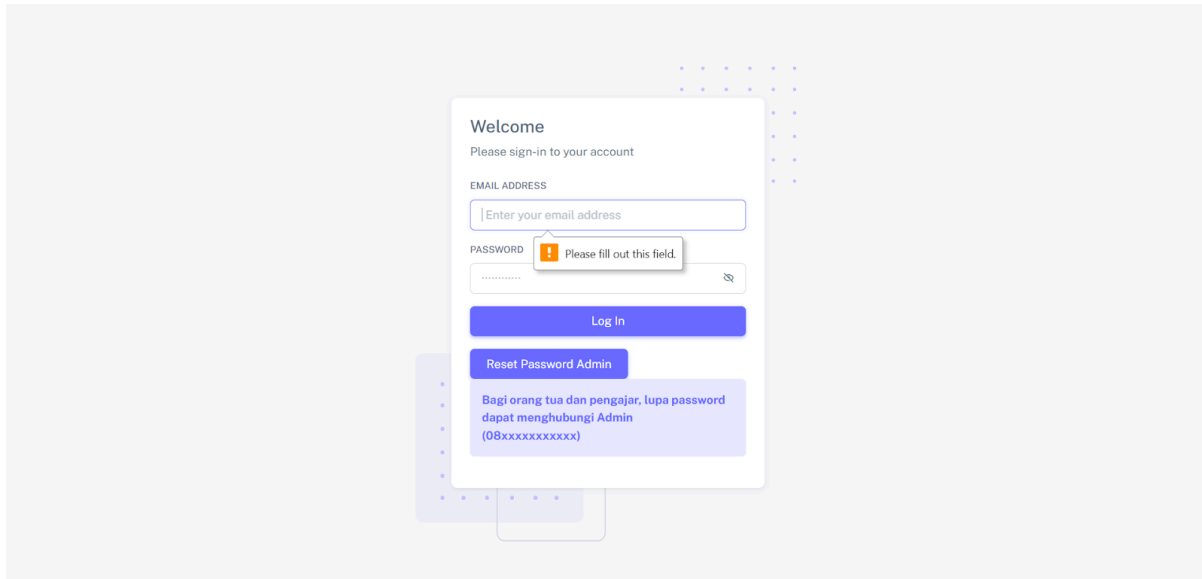
Bab ini akan membahas mengenai uji coba hasil implementasi sistem informasi presensi dan evaluasi pada lembaga kursus Messenger Academy Surabaya. Uji coba yang dilakukan meliputi verifikasi dan validasi. Verifikasi dilakukan untuk memastikan program terhindar dari error maupun bug pada aplikasi. Validasi dilakukan untuk memastikan program dibuat sesuai kebutuhan Lembaga Kursus dan Pelatihan Messenger Academy.

6.1 VERIFIKASI

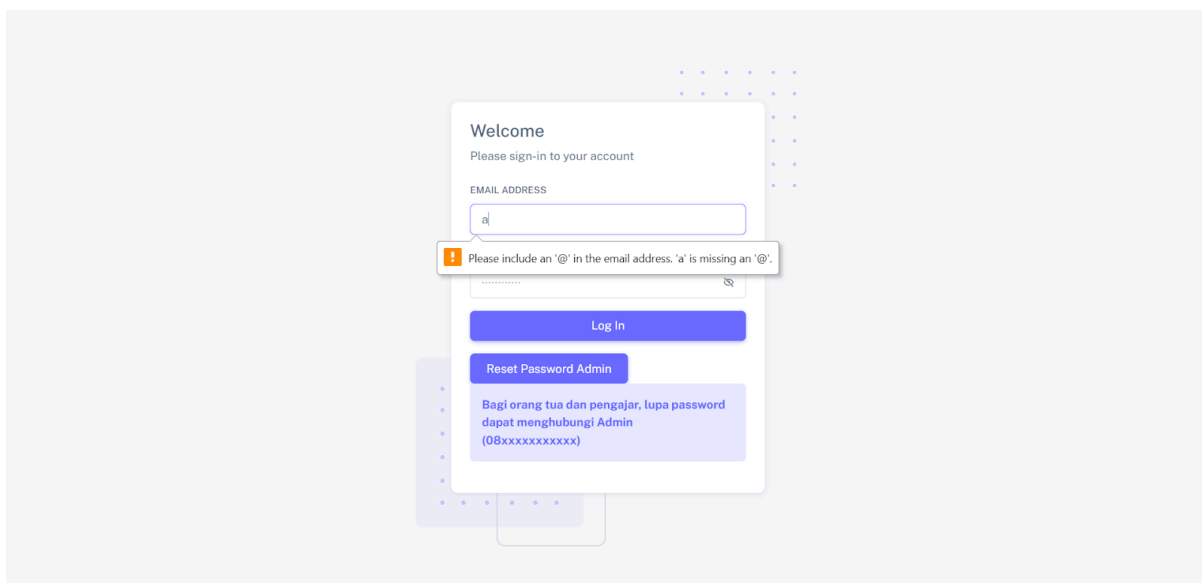
Pada tahap ini akan dilakukan verifikasi untuk memastikan program terbebas dari error maupun bug. Beberapa hal yang perlu diverifikasi pada sistem presensi dan evaluasi yaitu proses login, proses tambah data pengajar, peserta, dan orang tua.

6.1.1 VERIFIKASI PROSES LOGIN

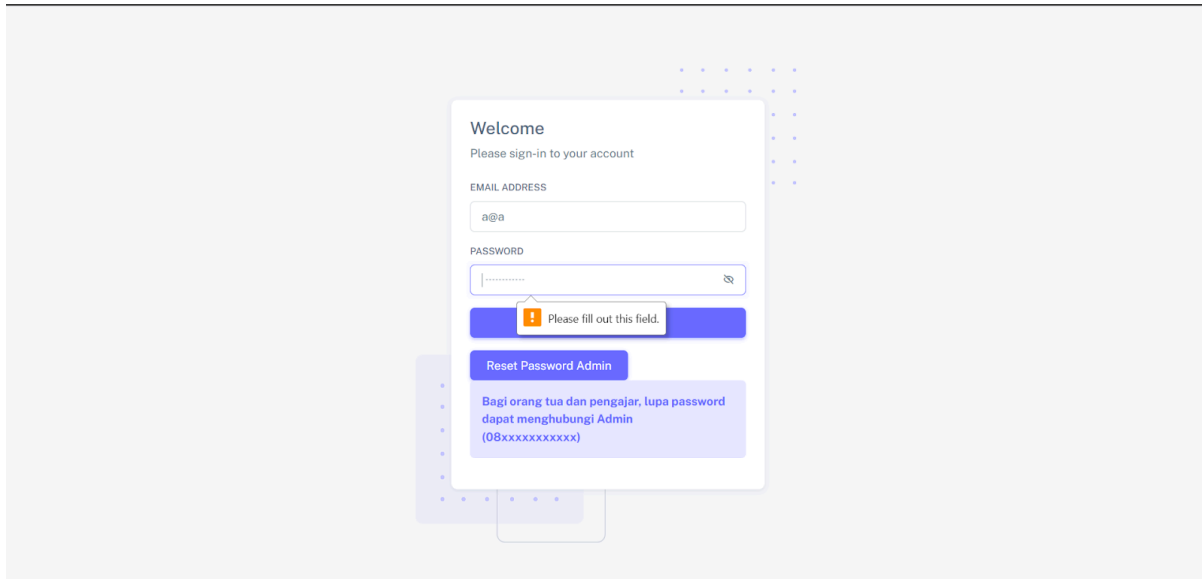
Pada proses login terdapat pesan pemberitahuan apabila email atau password tidak diisi atau tidak tepat pengisiannya sehingga pengguna dapat memastikan ketepatan dan kebenaran email dan password yang diinputkan. Apabila akun tidak terdeteksi, maka pengguna tidak dapat masuk pada sistem.



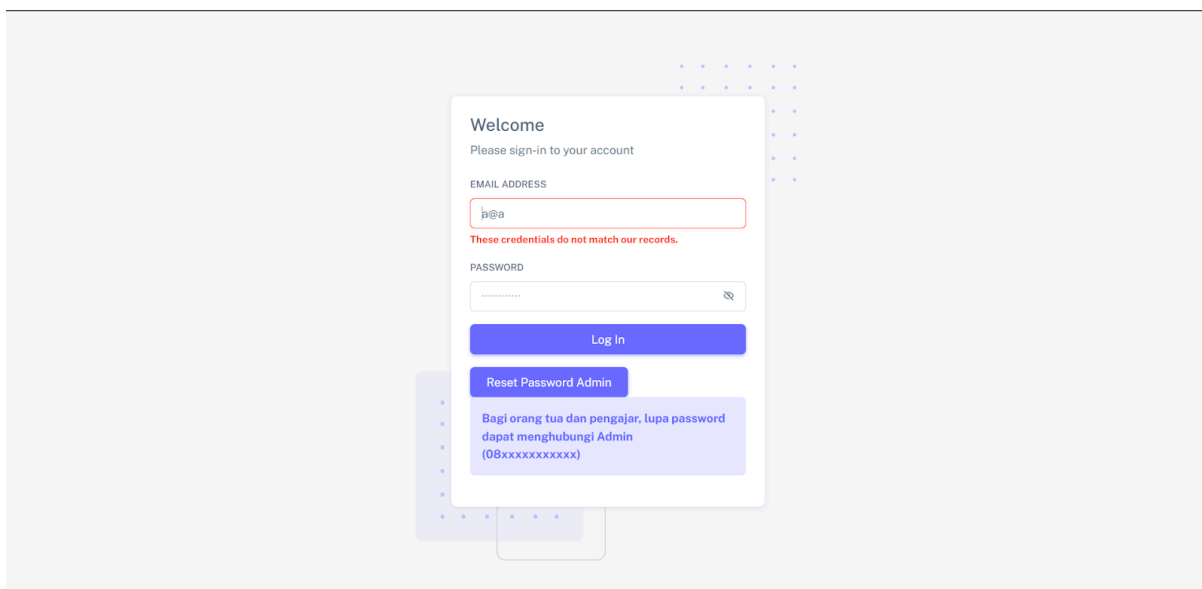
Gambar 6.1 Halaman Login Pengecekan Email Kosong



Gambar 6.2 Halaman Login Pengecekan Email



Gambar 6.3 Halaman Login Pengecekan Password



Gambar 6.4 Halaman Login Pengecekan Data

6.1.2 VERIFIKASI PROSES TAMBAH DATA PENGAJAR

Pada proses tambah data pengajar, terdapat pengecekan apabila pengisian tidak terisi. Pengguna akan diberitahu kembali untuk mengisi data sesuai dengan input yang disediakan. Hal tersebut meminimalisir kosongnya data yang dimasukkan.

Messenger 300+ ACADEMY

15 Juli 2024 pukul 14.17.41

Daftar Pengajar

Daftar Peserta

Daftar Orang Tua

Daftar Pelatihan

Periode

Logout

Add New Pengajar

Nama

Alamat

Please fill out this field.

Email

Password

Submit

Gambar 6.5 Halaman Tambah Data Pengajar

6.1.3 VERIFIKASI PROSES TAMBAH DATA PESERTA

Pada proses tambah data peserta, terdapat pengecekan apabila pengisian tidak terisi. Pengguna akan diberitahu kembali untuk mengisi data sesuai dengan input yang disediakan. Hal tersebut meminimalisir kosongnya data yang dimasukkan.

Messenger 300+ ACADEMY

15 Juli 2024 pukul 16.50.18

Daftar Pengajar

Daftar Peserta

Daftar Orang Tua

Daftar Pelatihan

Periode

Logout

Add New Peserta

Nama

Umur

Please fill out this field.

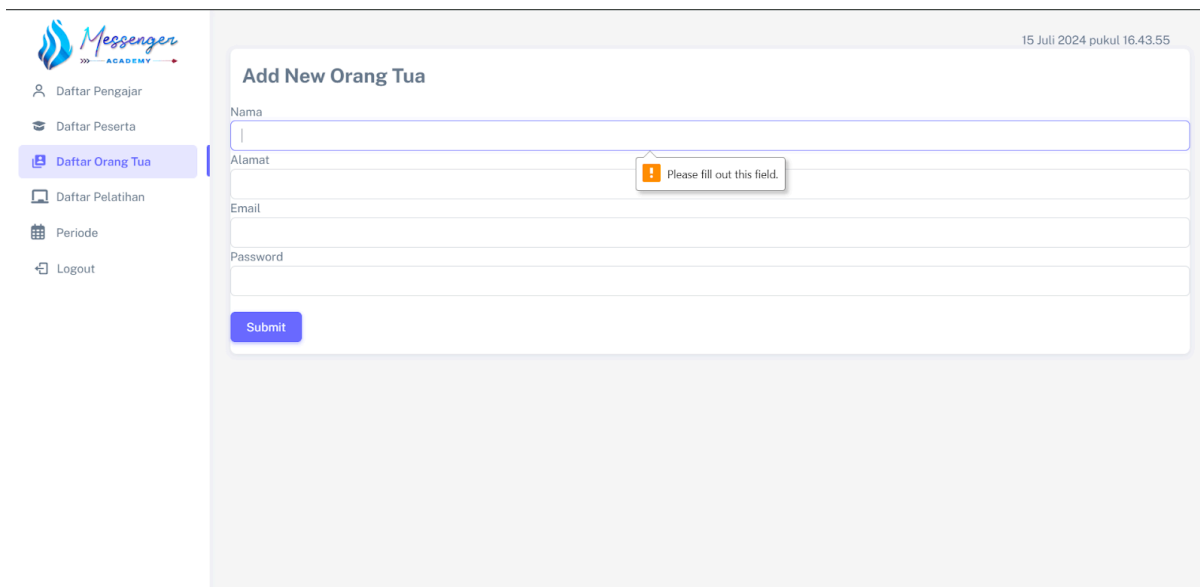
Orang tua Yanta

Submit

Gambar 6.6 Halaman Tambah Data Peserta

6.1.4 VERIFIKASI PROSES TAMBAH DATA ORANG TUA

Pada proses tambah data orang tua, terdapat pengecekan apabila pengisian tidak terisi. Pengguna akan diberitahu kembali untuk mengisi data sesuai dengan input yang disediakan. Hal tersebut meminimalisir kosongnya data yang dimasukkan.



The screenshot displays a web interface for adding a new parent. On the left is a sidebar with navigation links: 'Daftar Pengajar', 'Daftar Peserta', 'Daftar Orang Tua' (highlighted), 'Daftar Pelatihan', 'Periode', and 'Logout'. The main content area is titled 'Add New Orang Tua' and contains four input fields: 'Nama', 'Alamat', 'Email', and 'Password'. The 'Alamat' field is currently empty and has a red border with a tooltip message 'Please fill out this field.' pointing to it. A blue 'Submit' button is located below the fields. The top right corner shows the date and time: '15 Juli 2024 pukul 16.43.55'.

Gambar 6.7 Halaman Data Orang Tua

6.1.5 VERIFIKASI PROSES TAMBAH DATA PELATIHAN

Pada proses tambah data pelatihan, terdapat pengecekan apabila pengisian tidak terisi. Pengguna akan diberitahu kembali untuk mengisi data sesuai dengan input yang disediakan. Hal tersebut meminimalisir kosongnya data yang dimasukkan.

Messenger 300 ACADEMY

15 Juli 2024 pukul 17:18.04

Add New Pelatihan

Nama

Deskripsi Please fill out this field.

Jumlah Pertemuan

Submit

Daftar Pengajar
Daftar Peserta
Daftar Orang Tua
Daftar Pelatihan
Periode
Logout

Gambar 6.8 Halaman Tambah Data Pelatihan

6.1.6 VERIFIKASI PROSES TAMBAH DATA PERIODE PELATIHAN

Pada proses tambah data periode pelatihan, terdapat pengecekan apabila pengisian tidak terisi. Pengguna akan diberitahu kembali untuk mengisi data sesuai dengan input yang disediakan. Hal tersebut meminimalisir kosongnya data yang dimasukkan.

Messenger 300 ACADEMY

15 Juli 2024 pukul 17:31.13

Add New Periode

Tanggal Start dd/mm/yyyy

Jenis Kela Please fill out this field.

Pengajar Yantu

Pelatihan Akuntansi

Jadwal Pertemuan

Hari Pilih Hari Pertemuan

Waktu

Tambah Jadwal **Clear jadwal**

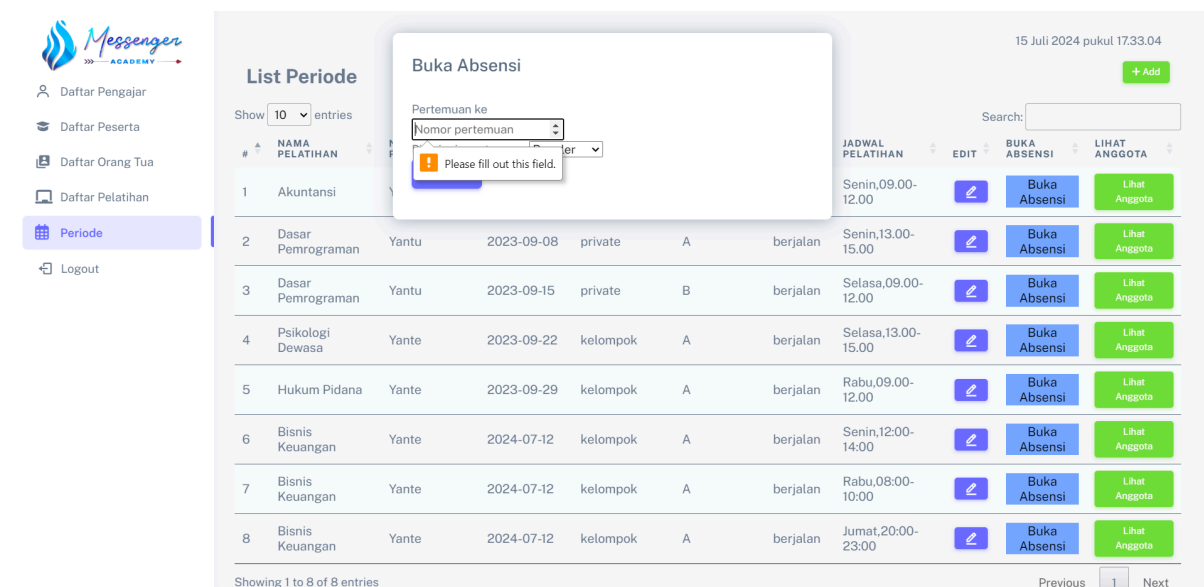
Submit

Daftar Pengajar
Daftar Peserta
Daftar Orang Tua
Daftar Pelatihan
Periode
Logout

Gambar 6.9 Halaman Tambah Data Periode Pelatihan

6.1.7 VERIFIKASI PROSES BUKA ABSENSI

Pada proses buka absensi, admin harus mengisi pertemuan yang sedang berjalan. Selain itu admin harus mengisi pertemuan yang sedang berjalan merupakan pertemuan reguler atau pertemuan pengganti. Terdapat pesan error apabila pertemuan tidak terisi oleh admin.



Gambar 6.10 Halaman Buka Presensi

6.2 VALIDASI

Validasi bertujuan untuk memastikan bahwa aplikasi yang dibuat mudah digunakan dan sesuai dengan kebutuhan LKP Messenger Academy. Validasi akan dilakukan dengan melakukan demonstrasi sistem dan mempersilahkan pengguna, seperti admin dan pengajar untuk mencoba aplikasi. Setelah itu, akan dilakukan wawancara untuk meminta pendapat pengguna terhadap sistem presensi dan evaluasi

Berdasarkan hasil wawancara yang telah dilakukan terhadap admin, pengajar, dan dipantau oleh pengelola LKP. Wawancara dilakukan dengan mendemonstrasikan program dan menanyakan pada masing-masing role apakah sesuai dengan kebutuhan.

Pada role admin merasa sudah cukup dengan dapat memasukkan setiap data dan terdapat fitur edit untuk dapat mengganti data dan dapat melakukan laporan presensi dengan tidak perlu membuat laporan setiap bulannya. Role pengajar pun dimudahkan karena tidak perlu untuk mengisi presensi secara manual dan mengisi data yang terus menerus. Laporan yang dihasilkan pun tidak akan terlambat karena semua dilakukan by system.

Sistem yang telah dibuat mendapatkan respon yang cukup baik dan membantu proses presensi dan pengiriman laporan kepada orang tua. Jadi dapat disimpulkan bahwa sistem yang telah dibuat cukup bagus dan mudah digunakan.

BAB 7

KESIMPULAN DAN SARAN

Pada bab ini akan membahas kesimpulan dari pembuatan sistem ini dan saran yang digunakan untuk pengembangan aplikasi di masa yang akan datang. Kesimpulan dan saran diperoleh berdasarkan seluruh langkah-langkah dan kegiatan yang dilakukan selama proses pembuatan sistem presensi dan evaluasi.

7.1 KESIMPULAN

Kesimpulan dari pembuatan sistem presensi dan evaluasi yang ditujukan pada LKP Messenger Academy adalah sebagai berikut:

1. Sistem yang dibuat sudah dapat melakukan presensi regular maupun pengganti dengan disertai evaluasi hingga dapat mengeluarkan laporan evaluasi yang dapat dibaca langsung oleh orang tua.
2. Sistem yang dapat diakses admin dapat memiliki akses penuh atas data pengajar, peserta, orang tua, pelatihan, periode pelatihan, dan buka presensi.
3. Sistem yang diakses oleh pengajar hanya dapat digunakan untuk melakukan presensi dan memberikan evaluasi terhadap peserta.
4. Sistem yang diakses oleh orang tua hanya dapat digunakan untuk melihat hasil laporan pertemuan murid.

7.2 SARAN

Validasi bertujuan untuk memastikan bahwa aplikasi yang dibuat mudah digunakan dan sesuai dengan kebutuhan LKP Messenger Academy. Validasi akan dilakukan dengan melakukan demonstrasi sistem dan mempersilahkan pengguna, seperti admin dan pengajar untuk mencoba aplikasi. Setelah itu, akan dilakukan wawancara untuk meminta pendapat pengguna terhadap sistem presensi dan evaluasi

Berdasarkan hasil wawancara yang telah dilakukan terhadap admin, pengajar, dan dipantau oleh pengelola LKP, sistem yang telah dibuat mendapatkan respon yang cukup baik dan membantu proses presensi dan pengiriman laporan kepada orang tua. Jadi dapat disimpulkan bahwa sistem yang telah dibuat cukup bagus dan mudah digunakan.

DAFTAR PUSTAKA

Merdekawati, A., Rahayu, L. K., & Martini, M. (2022). Sistem Pengolahan Nilai Siswa Secara Online (E-Raport) Menggunakan Waterfall. *JTIK (Jurnal Teknik Informatika Kaputama)*, 6(2), 495–504.

Munthe, B., Arifin, A., Nugroho, B. S., & Fitriani, E. (2021). Online student attendance system using Android. *Journal of Physics: Conference Series*, 1933(1), 012048.

Satria, M. B., & Ardiansyah, H. (2023). Analisis dan Perancangan Sistem Raport Digital Metode Waterfall. *Journal on Education*, 5(2), 5143–5151.

Wantoro, A. (2016). Pengembangan Sistem Presensi Dan Kedisiplinan Dosen Terhadap Biaya Operasional Perguruan Tinggi. *Jurnal Teknoinfo*, 10(1), 1–5.