



APLIKASI KEHADIRAN ONLINE

Disusun Oleh

• SYAHRIANI, TRI SANTOSO, ASTRILYANA, ARDIAN DWI PRABA,
IDAH YUNIASIH, SETIAJI, MARYANAH SAFITRI, ALBERT RIYANDI

JAKARTA 2024



KATA PENGANTAR

Dalam kesempatan yang berbahagia ini, kami dengan rendah hati ingin menyampaikan buku panduan (*Manual Book*) yang akan dibuatkan dalam Hak Kekayaan Intelektual (HKI) yang telah disusun dengan penuh dedikasi dan kerja keras. Buku panduan ini mencakup prosedur langkah-langkah dalam penggunaan **Aplikasi Kehadiran Online Sederhana**. Kami ingin mengucapkan terima kasih kepada semua pihak yang telah memberikan kontribusi, baik secara langsung maupun tidak langsung, dalam penyusunan buku panduan ini. Semoga Aplikasi ini bermanfaat bagi pembaca dan dapat menjadi sumber inspirasi untuk pengembangan lebih lanjut. Akhir kata, kami mengharapkan masukan dan saran yang konstruktif dari para pembaca guna perbaikan dan pengembangan Aplikasi ini di masa yang akan datang.

Jakarta , 13 April 2024

Tim Penyusun

DAFTAR ISI

COVER	i
KATA PENGANTAR	ii
A. PENDAHULUAN.....	1
B. SPESIFIKASI KOMPUTER.....	2
C. PROSEDUR PENGGUNAAN APLIKASI KEHADIRAN ONLINE SEDERHANA.....	3
D. SCRIPT APLIKASI KEHADIRAN ONLINE SEDERHANA.....	6

A. Pendahuluan

Presensi Kehadiran adalah suatu tindakan atau rutinitas yang dilakukan oleh seorang dosen(pengajar) untuk menunjukkan bahwa ada mahasiswa yang hadir dalam suatu kelas atau rapat(Mulia, 2020). Dahulu seorang pengajar melakukan presensi kehadiran dengan cara memanggil satu persatu nama murid(mahasiswa) dalam kelas yang diajarnya. Hal tersebut dirasa kurang efektif dalam permasalahan presensi kehadiran, dimana seorang pengajar harus menyiapkan waktu beberapa menit untuk melakukan presensi kehadiran murid(mahasiswa)nya, baik diawal mulai pengajaran atau pada akhir pengajaran.

Untuk mengurangi permasalahan yang ada didalam kelas saat jam kegiatan belajar mengajar, maka memerlukan terobosan baru untuk presensi kehadiran. Hal ini dimaksud, agar waktu yang terbuang sia-sia dapat diminalisir dan dapat digunakan sepenuhnya untuk kegiatan belajar mengajar. Sistem presensi mahasiswa ini bertujuan untuk mengefektifkan dan mempercepat proses presensi, serta dapat dimanfaatkan oleh pihak terkait (kampus atau sekolah), guna menilai tingkat partisipasi murid(mahasiswa). Informasi tersebut kedepannya akan dimanfaatkan sebagai komponen kelulusan murid(mahasiswa)(Anwar, Tiyas, Rosaidi, 2023).

Sebenarnya, sudah banyak sekali pengembangan teknologi yang dilakukan oleh banyak peneliti dalam pembangunan sistem kehadiran. Ada yang menggunakan sidik jari, website, pengenalan wajah, pengenalan retina mata dan lain sebagainya. Tetapi didalam pembahasan kali ini, Aplikasi Kehadiran Online Sederhana yang dibuat ini menggunakan website. Dimana aplikasi ini hanya berisi menu kehadiran mahasiswa saja tanpa adanya rekap kehadiran. Aplikasi sederhana ini hanya sampai sebatas kehadiran yang sudah dilakukan dapat tersimpan kedalam tabel absensi saja.

B. SPESIFIKASI KOMPUTER

1. **Spesifikasi computer**

A. Kebutuhan Pengajar atau

Server Hardware

- a. PC intel i3
- b. Windows atau Linux 64Bit
- c. Ram 4GB
- d. Hdd 250 GB

Software

- a. Web Browser
- b. PHP versi 7.4 atau lebih tinggi
- c. Mysql versi 5.7
- d. Xampp atau Apache

B. Kebutuhan Pengguna (murid) Hardware

- a. PC / Laptop
- b. Windows atau Linux atau Mac
- c. Ram 1 GB Minimal
- d. Hdd 250 GB

Software

- a. Web Browser

C. PROSEDUR PENGGUNAAN APLIKASI KEHADIRAN ONLINE SEDERHANA

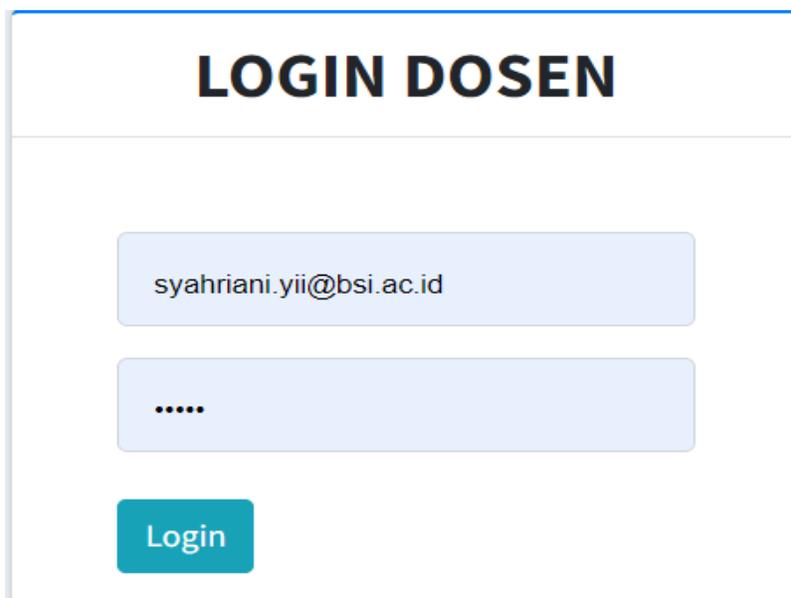
Pembuatan aplikasi ini menggunakan website dan bantuan bootstrap untuk frameworknya. Data-data yang sudah dibuat akan disimpan kedalam database server mysql dengan menggunakan bahasa php, html dan javasript. Pada apliasi sederhana ini, disediakan 1 hak akses saja yaitu diberikan kepada pengajar(dosen/guru) yang dapat melakukan proses kehadiran murid(mahasiswa)nya. Untuk menggunakan aplikasi ini pengajar (dosen/guru) harus melakukan login terlebih dahulu dengan memasukan *username* dan *password* standar yang diberikan.

Adapun tahapan-tahapan untuk menjalankan aplikasi kehadiran online sederhana ini adalah sebagai berikut:

A. Tampilan Program User

1. Halaman Login

Halaman ini berfungsi untuk memvalidasi username dan password dari pengajar(dosen/guru) yang telah diberikan hak izin akses, sesuai atau tidak dengan data yang berada pada database.



Gambar 1. Halaman Login Pengajar (Dosen/Guru)

2. Halaman Beranda

Halaman beranda untuk pengguna berfungsi sebagai antarmuka utama setelah login, memberikan akses cepat ke informasi dan fitur penting. Ini biasanya mencakup ringkasan aktivitas, notifikasi terbaru, dan akses cepat ke fitur inti seperti absensi harian. Halaman ini dirancang untuk memberikan gambaran umum tentang status pengguna, seperti absensi terbaru dan jadwal kerja, serta menyediakan navigasi mudah ke bagian lain dari aplikasi, seperti profil pengguna, laporan absensi, dan pengaturan akun.



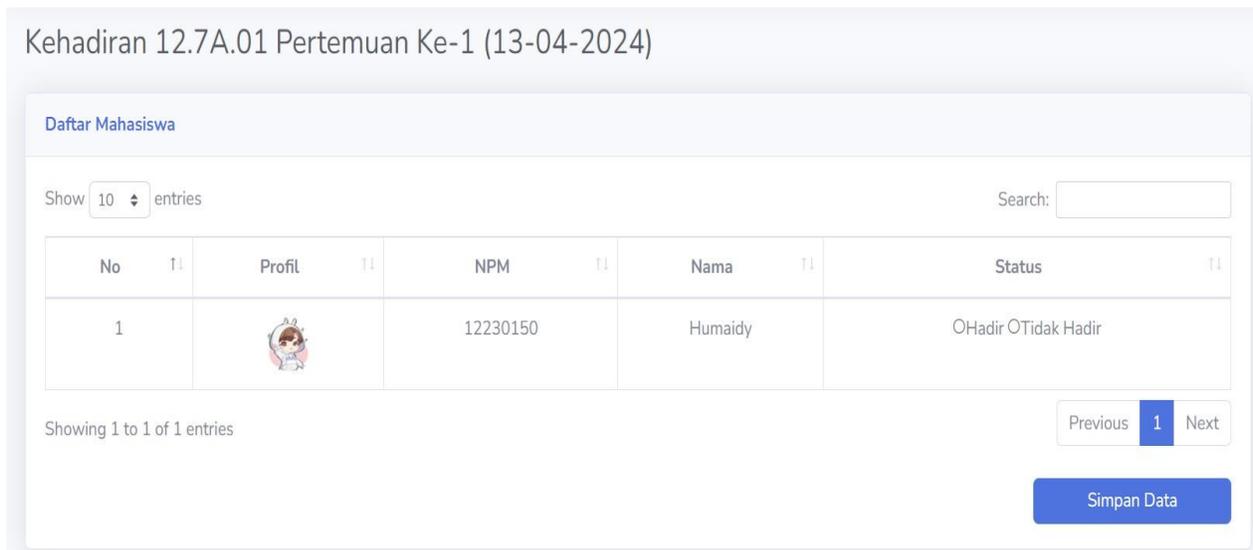
Gambar 2. Halaman Beranda

3. Halaman Kehadiran

Halaman ini berisi halaman yang akan digunakan untuk melakukan proses kehadiran, dan penginputannya disesuaikan dengan jumlah murid/mahasiswa yang hadir pada kelas saat jam pengajaran dimulai sampai selesai.



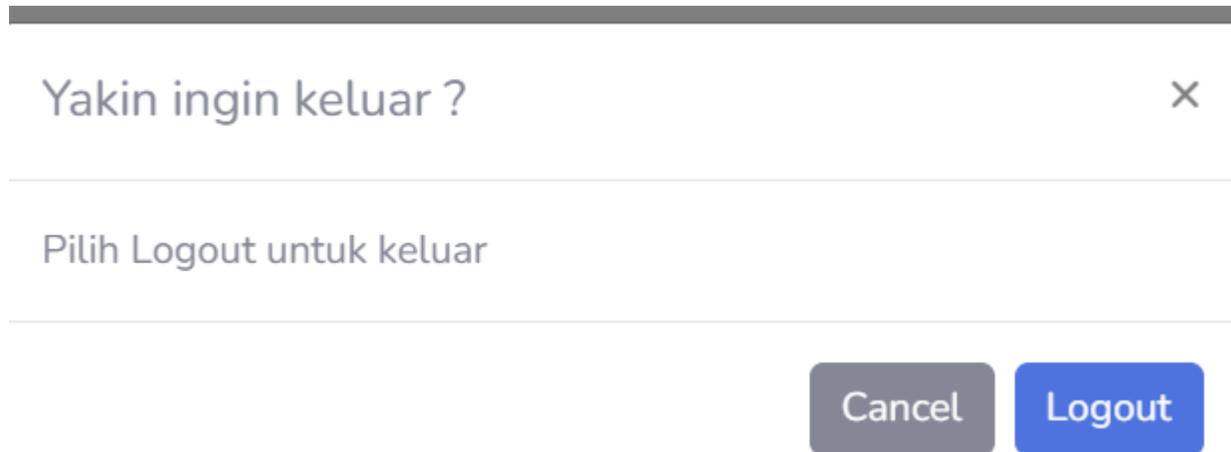
Gambar 3. Halaman Kehadiran Berupa Card Informasi



Gambar 4. Halaman Kehadiran

4. Logout

Halaman ini berisi perintah keluar dari aplikasi, bila sudah selesai digunakan dan kembali ke halaman awal yaitu login.

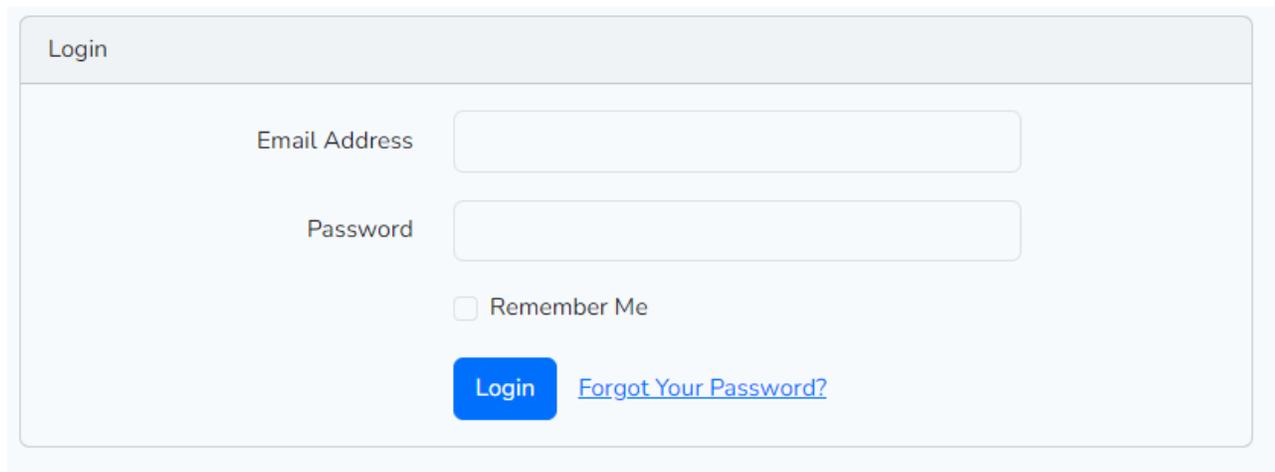


Gambar 4. Halaman Logout

B. Tampilan Program Admin

1. Halaman Login Admin

Halaman login untuk admin berfungsi sebagai pintu masuk yang aman ke dashboard admin, memungkinkan administrator mengautentikasi diri dengan memasukkan kredensial (username/email dan password). Ini memastikan hanya pengguna yang berwenang dapat mengakses dan mengelola konten serta fitur sistem backend secara efektif dan aman.



Gambar 5. Halaman Login admin

2. Halaman Dashboard Admin

Halaman dashboard admin berfungsi sebagai pusat kontrol utama bagi administrator, memberikan akses menyeluruh ke berbagai aspek manajemen sistem. Ini mencakup fitur utama seperti monitoring data absensi karyawan, mengelola pengguna (menambah, menghapus, atau mengedit informasi karyawan), dan melihat laporan absensi harian, mingguan, atau bulanan. Dashboard juga menampilkan ringkasan statistik penting seperti jumlah karyawan yang hadir, terlambat, atau absen. Dengan antarmuka yang intuitif dan informasi yang disajikan secara real-time, halaman ini membantu administrator dalam mengambil keputusan cepat, memastikan operasional berjalan lancar, dan meningkatkan efisiensi pengelolaan sumber daya manusia dalam perusahaan.



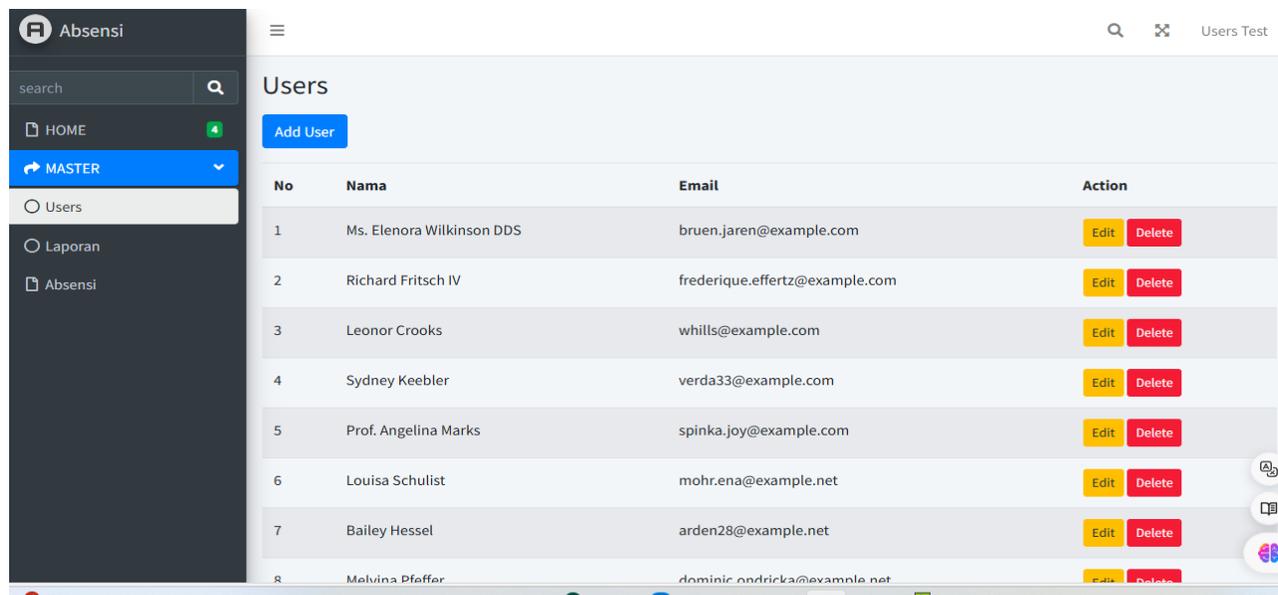
Gambar 6. Halaman Dashboard Admin

3. Halaman Master Users atau Pengguna

Halaman master user yang berisi fitur CRUD (Create, Read, Update, Delete) berfungsi sebagai alat utama bagi administrator untuk mengelola data pengguna dalam sistem. Berikut adalah fungsionalitas utama dari halaman ini:

- a. Create (Membuat): Administrator dapat menambahkan pengguna baru ke sistem dengan memasukkan informasi seperti nama, email, jabatan, dan detail lainnya. Formulir input yang disediakan memastikan data yang dimasukkan lengkap dan akurat.
- b. Read (Membaca): Halaman ini menampilkan daftar lengkap semua pengguna yang terdaftar dalam sistem. Informasi penting seperti nama, email, tanggal pendaftaran, dan status aktif ditampilkan dalam bentuk tabel, memudahkan administrator untuk memantau dan mengakses informasi pengguna secara cepat.
- c. Update (Memperbarui): Administrator dapat memperbarui informasi pengguna yang ada, seperti mengubah detail kontak, memperbarui jabatan, atau mereset kata sandi. Fitur ini memastikan data pengguna tetap terkini dan relevan.
- d. Delete (Menghapus): Administrator dapat menghapus pengguna dari sistem. Ini penting untuk menjaga database tetap bersih dan menghapus data pengguna yang tidak lagi relevan atau diperlukan.

Dengan fitur CRUD ini, halaman master user memungkinkan administrator untuk mengelola data pengguna dengan efisien, memastikan integritas data, dan meningkatkan pengelolaan sumber daya manusia dalam organisasi. Hal ini juga memungkinkan fleksibilitas dalam penanganan perubahan staf dan memastikan akses kontrol yang tepat dalam sistem.



Gambar 7. Halaman users atau pengguna

4. Halaman Laporan Absensi user

Halaman laporan absensi dalam aplikasi absensi sangat penting untuk memantau dan mengelola kehadiran karyawan secara efektif. Berikut adalah penjelasan tentang berbagai fungsionalitasnya:

a. Melihat Data Absensi Per Tanggal:

- Administrator dapat melihat data absensi untuk tanggal tertentu. Fitur ini menampilkan daftar karyawan yang hadir, waktu check-in dan check-out, serta status kehadiran (hadir, terlambat, atau absen) pada tanggal yang dipilih.
- Ini berguna untuk memantau kehadiran harian dan memastikan karyawan mengikuti jadwal kerja mereka.

b. Melihat Data Absensi Per Periode:

- Administrator dapat menarik laporan absensi untuk periode tertentu, misalnya mingguan, bulanan, atau kustom sesuai kebutuhan.
- Laporan ini membantu dalam analisis jangka panjang, seperti mengevaluasi pola kehadiran, mengidentifikasi tren ketidakhadiran, dan merencanakan tindakan korektif jika diperlukan.
- Fitur ini juga berguna untuk penyusunan laporan kinerja bulanan atau tahunan, serta untuk keperluan audit.

c. Melihat Absensi Per User:

- Administrator dapat menelusuri data absensi spesifik untuk setiap karyawan. Dengan memilih seorang pengguna, mereka dapat melihat riwayat absensi lengkap, termasuk semua tanggal hadir, jam check-in dan check-out, serta status kehadiran setiap hari.
- Fitur ini membantu dalam evaluasi kinerja individu, memberikan wawasan tentang kebiasaan kehadiran karyawan, dan memungkinkan diskusi berbasis data selama penilaian kinerja atau pertemuan one-on-one.

Dengan kombinasi dari fitur-fitur ini, laporan absensi memberikan alat yang komprehensif bagi administrator untuk memantau dan menganalisis kehadiran karyawan secara detail dan

menyeluruh. Hal ini memastikan transparansi, meningkatkan akuntabilitas, dan mendukung pengambilan keputusan yang berdasarkan data.

Laporan Absensi

show 10 entries Search:

No	Nama	Email	Tanggal	Check-In	Check-Out
1	Users Test	test@mail.com	2024-06-22	11:53:51	11:53:56
2	Users Test	test@mail.com	2024-06-22	11:53:54	
3	Leonor Crooks	whills@example.com	2014-11-07	00:22:37	
4	Ms. Elenora Wilkinson DDS	bruen.jaren@example.com	2004-09-07	19:02:56	14:40:40
5	Ms. Elenora Wilkinson DDS	bruen.jaren@example.com	1983-01-08	05:56:29	15:25:45
6	Leonor Crooks	whills@example.com	1981-09-05	03:18:34	
7	Prof. Angelina Marks	spinka.joy@example.com	2009-04-21	12:43:14	21:02:35
8	Carmelo Graham	corine.labadie@example.com	1985-05-15	01:55:18	
9	Prof. Angelina Marks	spinka.joy@example.com	1986-01-01	17:13:15	09:46:27

Gambar 8. Halaman Laporan Absensi

D. SCRIPT APLIKASI KEHADIRAN ONLINE SEDERHANA

1. Script Login.php

```
namespace App\Http\Controllers;

use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;

class AuthController extends Controller
{
    public function showLoginForm()
    {
        return view('auth.login');
    }

    public function login(Request $request)
    {
        $request->validate([
            'email' => 'required|email',
            'password' => 'required',
        ]);

        if (Auth::attempt($request->only('email', 'password'))) {
            return redirect()->intended('dashboard');
        }

        return back()->withErrors([
            'email' => 'The provided credentials do not match our records.',
        ]);
    }

    public function logout(Request $request)
    {
        Auth::logout();
        return redirect('/login');
    }
}
```

1. Script Absensi.php

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Models\Attendance;
use Auth;

class AttendanceController extends Controller
{
    public function index()
    {
        $attendances = Attendance::where('user_id', Auth::id()->get());
        return view('attendances.index', compact('attendances'));
    }

    public function store(Request $request)
    {
        $request->validate([
            'date' => 'required|date',
            'check_in' => 'required|date_format:H:i',
        ]);

        Attendance::create([
            'user_id' => Auth::id(),
            'date' => $request->date,
            'check_in' => $request->check_in,
            'check_out' => $request->check_out,
        ]);

        return redirect()->route('attendances.index');
    }

    public function checkin(Request $request)
    {
        $request->validate([
            'date' => 'required|date',
            //check_in' => 'required|date_format:H:i',
        ]);

        Attendance::create([
            'user_id' => Auth::id(),
            'date' => $request->date,
            'check_in' => date('H:i:s'),
            //check_out' => $request->check_out,
        ]);

        return redirect()->route('attendances.index');
```

```

}

public function checkout(Request $request)
{
    $request->validate([
        'date' => 'required|date',
    ]);

    $attendance = Attendance::where('user_id', Auth::id())
        ->where('date', $request->date)
        ->first();

    if ($attendance && is_null($attendance->check_out)) {
        $attendance->update([
            'check_out' => date('H:i:s'),
        ]);
    }

    return redirect()->route('attendances.index');
}
}

```