

# INTEGRASI ODOO 16 DALAM PENGEMBANGAN APLIKASI ABSENSI MAHASISWA (STUDI KASUS: UNIVERSITAS SINGAPERBANGSA KARAWANG)

Adrian Wibisono<sup>1</sup>, Didi Juardi<sup>2</sup>, Asep Jamaludin<sup>3</sup>

<sup>1,2,3</sup>Universitas Singaperbangsa Karawang; Jl. HS.Ronggo Waluyo, Puseurjaya, Telukjambe Timur Karawang, Indonesia; (0267) 641177

Received: 10 Juli 2024

Accepted: 31 Juli 2024

Published: 7 Agustus 2024

## Keywords:

Absensi, Odoo 16, Website, Usability Testing

## Correspondent Email:

adrianwibisono03@gmail.com

**Abstrak.** Dalam era perkembangan teknologi informasi, integrasi Odoo menjadi solusi inovatif untuk mengatasi tantangan dalam sistem absensi di institusi pendidikan. Universitas membutuhkan sistem manajemen yang efisien untuk mendukung kegiatan administratif dan akademis. Odoo, sebuah sistem Enterprise Resource Planning (ERP), menawarkan solusi terintegrasi untuk berbagai aspek manajemen, termasuk absensi mahasiswa. Pemilihan Odoo 16 dalam pengembangan aplikasi absensi mahasiswa bertujuan untuk meningkatkan efektivitas sistem, pengelolaan data, dan pengalaman administrasi kehadiran mahasiswa. Evaluasi dilakukan menggunakan metode analisis usability dengan lima indikator: Learnability, Memorability, Efficiency, Errors, dan Satisfaction. Penelitian ini bertujuan untuk mengintegrasikan Odoo 16 dalam pengembangan aplikasi absensi mahasiswa, mengatasi permasalahan efisiensi dan akurasi yang ada. Dampak positif dari penelitian ini adalah peningkatan pengelolaan administrasi universitas dan mendukung tujuan pendidikan dengan teknologi terkini. Dengan demikian, penelitian ini tidak hanya memecahkan masalah manajemen absensi mahasiswa, tetapi juga menghadapi tuntutan teknologi di dunia pendidikan.

**Abstract.** In the era of information technology development, Odoo integration is an innovative solution to overcome challenges in attendance systems in educational institutions. Universities need efficient management systems to support administrative and academic activities. Odoo, an Enterprise Resource Planning (ERP) system, offers integrated solutions for various aspects of management, including student attendance. The choice of Odoo 16 in developing student attendance applications aims to improve system effectiveness, data management and student attendance administration experience. Evaluation was carried out using the usability analysis method with five indicators: Learnability, Memorability, Efficiency, Errors, and Satisfaction. This research aims to integrate Odoo 16 in the development of student attendance applications, overcoming existing efficiency and accuracy problems. The positive impact of this research is improving university administration management and supporting educational goals with the latest technology. Thus, this research not only solves the problem of student attendance management, but also faces the demands of technology in the world of education.

## 1. PENDAHULUAN

Dalam era perkembangan teknologi informasi, integrasi Odoo menjadi sebuah tema yang menarik, menawarkan solusi inovatif untuk mengatasi tantangan yang dihadapi dalam sistem absensi yang sudah ada. Universitas sebagai lembaga pendidikan memiliki kebutuhan akan sistem manajemen yang efisien untuk mendukung kegiatan administratif dan akademis. Odoo adalah sebuah sistem *Enterprise Resource Planning* (ERP) yang menyediakan solusi terintegrasi untuk berbagai aspek manajemen, mulai dari akuntansi dan manufaktur hingga penjualan dan sumber daya [1]. Pemilihan Odoo sebagai pusat integrasi dalam pengembangan aplikasi absensi mahasiswa bukan sekadar pilihan teknologi, melainkan langkah strategis untuk meningkatkan efektivitas sistem, meningkatkan pengelolaan data, dan memberikan pengalaman yang lebih baik dalam administrasi kehadiran mahasiswa di lingkungan akademis.

Oleh karena itu, pemahaman mendalam tentang fitur dan potensi Odoo menjadi kunci utama dalam merancang solusi yang inovatif dan efisien. Dalam konteks ini, Odoo 16 dipilih karena keunggulan dan fitur-fiturnya yang dapat memberikan solusi yang komprehensif. Keputusan ini didukung oleh kemampuan Odoo 16 untuk menyediakan platform terintegrasi yang dapat digunakan untuk mengelola berbagai aspek, termasuk absensi mahasiswa. Odoo 16 menggunakan sistem manajemen konten (CMS) Odoo dan ditulis dalam bahasa pemrograman Python, dengan PostgreSQL sebagai basis datanya. Untuk aspek front-end, Odoo 16 menggunakan framework UI Bootstrap serta berbagai pustaka JavaScript seperti Moment.js, Select2, jQuery UI, dan jQuery. Odoo juga mengintegrasikan berbagai skrip font dari Google Font API dan menggunakan Popper serta Less untuk fungsionalitas tambahan. Sistem ini berjalan di server Nginx dan mendukung HTTP/2, menunjukkan kemampuan untuk menangani beban lalu lintas tinggi dengan performa optimal. Oleh karena itu, penelitian ini bertujuan untuk mengintegrasikan Odoo 16

dalam pengembangan aplikasi absensi mahasiswa, sebagai langkah proaktif untuk mengatasi permasalahan tersebut. Penelitian ini memiliki dampak positif yang signifikan terhadap pengelolaan administrasi universitas, mendukung upaya pencapaian tujuan pendidikan dengan menggunakan teknologi terkini.

## 2. TINJAUAN PUSTAKA

### 2.1. *Enterprise Resource Planning* (ERP)

Perencanaan Sumber Daya Perusahaan (ERP) adalah sistem informasi yang membantu perusahaan merencanakan dan mengelola sumber dayanya melalui berbagai aplikasi program yang terintegrasi dan multimodul. Tujuannya adalah untuk membantu berbagai fungsi perusahaan, meningkatkan efisiensi dan efektivitas operasional, dan memberikan layanan pelanggan terbaik. Diharapkan bahwa ini akan meningkatkan nilai dan menguntungkan semua pemangku kepentingan Perusahaan [2].

Sistem informasi ERP memainkan peran penting dalam mengintegrasikan dan mengotomatisasi proses bisnis perusahaan jasa atau manufaktur, ini mencakup semua aspek operasional, produksi, dan distribusi Perusahaan [3].

### 2.2. Odoo

Sebelumnya dikenal sebagai TinyERP dan OpenERP, sekarang diberi nama Odoo, merupakan sebuah platform perencanaan sumber daya perusahaan. Odoo S.A., yang berbasis di Belgia, mengembangkan kode kerangka kerja objek dan modul inti ERP. Persatuan global dan mitra bersertifikat di berbagai belahan dunia menyediakan aplikasi spesifik, dukungan, dan layanan [4].

Sistem ERP Odoo terdiri dari tiga komponen utama:

- Server database PostgreSQL, yang mencakup semua database dan elemen data utama dari sistem ERP Odoo.
- Server aplikasi Odoo, yang mencakup semua logika perusahaan.
- Server web, termasuk aplikasi yang disebut web open object client,

membantu koneksi ke Odoo melalui aplikasi penelusuran web standar dan tidak diperlukan jika terhubung melalui klien GTK (Graphic Toolkit). [5]

### 2.3. Absensi

Menurut kamus besar Bahasa Indonesia, kata absen merujuk pada keadaan seseorang yang tidak hadir di suatu instansi. Sementara itu, istilah absensi biasanya digunakan untuk mencatat atau menandai waktu kehadiran seseorang dalam dokumen yang disiapkan dengan benar, bertujuan sebagai referensi dalam pengambilan keputusan di dalam lingkup penelitian [6]

### 2.4. PostgreSQL

PostgreSQL adalah pilihan yang bagus untuk penyimpanan aplikasi sistem informasi karena mendukung banyak platform dan berlisensi bebas. Selain itu, PostgreSQL bertujuan sebagai database server yang andal, dan memiliki banyak fitur pendukung yang membuatnya menjadi pilihan yang bagus untuk pengguna database. Department Computer Science University of California Berkeley mengembangkan PostgreSQL. Sifatnya yang open source memungkinkan pengembangan sesuai dengan kebutuhan [7].

### 2.5. Pengujian Perangkat Lunak

Jaminan kualitas perangkat lunak merujuk pada pola yang sistematis dan terencana dalam semua tahapan yang dilaksanakan dimana bertujuan untuk memastikan bahwa suatu aplikasi telah memenuhi kebutuhan teknis dan aktivitas yang telah direncanakan untuk mengevaluasi aplikasi yang telah diciptakan [8].

#### A. Black Box Testing

Pengujian black box adalah suatu metode dalam pengujian perangkat lunak yang digunakan untuk mengevaluasi fungsionalitas aplikasi. Pada dasarnya, fokus utama dari pengujian ini adalah input yang diberikan pada aplikasi dan hasil yang diharapkan untuk setiap nilai input. Teknik pengujian ini didasarkan pada persyaratan dan spesifikasi perangkat lunak, di mana penguji tidak mengetahui detail kerja internal dari item yang diuji [9].

#### B. Evaluasi Penggunaan

*Usability Testing* merupakan salah satu kategori metode dalam evaluasi usability yang digunakan untuk mengevaluasi sebuah produk dengan mengujinya langsung pada pengguna [10]. *Usability Testing* dapat memberikan pemahaman mendalam terhadap sejauh mana pendapat pengguna akhir terhadap produk yang digunakan. *Usability Testing* adalah penilaian formal terhadap seberapa efektif, efisien, dan memuaskan interaksi pengguna dengan suatu produk atau sistem [11].

#### C. Instrumen Penelitian

Skala Likert digunakan sebagai metode pengukuran untuk menilai pendapat seseorang melalui kuesioner guna mengetahui sikap terhadap objek tertentu. Dalam skala ini, setiap jawaban kuesioner diberikan skor antara 1 hingga 5, di mana skor 1 menunjukkan pendapat "sangat tidak setuju" dan skor 5 menunjukkan pendapat "sangat setuju" [12].

### 2.6. Diagram UML

*Unified Modeling Language* (UML) adalah alat yang digunakan untuk merancang pengembangan perangkat lunak dengan pendekatan berbasis objek. UML juga menetapkan standar untuk membuat blueprint sistem, termasuk konsep proses bisnis, skema database, penulisan kelas-kelas dalam bahasa pemrograman tertentu, dan komponen sistem perangkat lunak yang diperlukan [13].

#### A. Use Case Diagram

Diagram *Use Case* adalah gambaran dari fungsionalitas yang diinginkan dari suatu sistem dan mencerminkan interaksi antara aktor dan sistem. Dalam *Use Case*, terdapat pemeran atau aktor yang merupakan representasi entitas manusia atau sistem yang melaksanakan tugas di dalam sistem [14]

#### B. Class Diagram

*Class Diagram* merupakan visualisasi dari struktur dan karakteristik class, package, dan objek yang memiliki hubungan antar satu sama lain, seperti pewarisan, asosiasi, dan elemen-elemen lainnya [15].

### 3. METODE PENELITIAN

Metodologi penelitian yang digunakan adalah SDLC (*Software Development Life Cycle*) dengan model proses *Agile*. Model proses ini digunakan karena sistem yang dikembangkan memerlukan fleksibilitas tinggi dan responsif terhadap perubahan kebutuhan yang mungkin muncul selama pengembangan.

## 4 HASIL DAN PEMBAHASAN

### 4.1 Hasil

Hasil penelitian integrasi Odoo 16 dalam pengembangan sistem absensi Mahasiswa di Universitas Singaperbangsa Karawang yang dirancang merupakan website ERP, dengan menggunakan metode *Software Development Life Cycle (SDLC)* model *Agile*. Dengan tahapan perencanaan, implementasi, testing, dokumentasi, peluncuran dan pemeliharaan.

### 4.2 Pembahasan

#### 4.2.1 Perencanaan

Tahap pertama penelitian ini adalah dengan observasi lalu menggambarkan kebutuhan pengguna menggunakan diagram UML *use case* yang kemudian dapat mengidentifikasi kebutuhan *software* dari pengguna. Pada tahap ini juga dilakukan penetapan tujuan sistem informasi, baik jangka pendek dan jangka panjang.

#### a. Analisis Kebutuhan

Pada tahap ini, peneliti melakukan observasi terhadap website E-Campus dan menggunakan literatur berupa penelitian sebelumnya yang terkait dengan sistem informasi. Hasil dari observasi tersebut memberikan beberapa fitur yang dapat diintegrasikan kedalam Odoo 16 diantaranya:

1. *Upload* gambar atau dokumen pada halaman absensi (untuk dosen).
2. Riwayat absensi mahasiswa yang sudah absen
3. Keterangan ketepatan waktu pada absensi mahasiswa

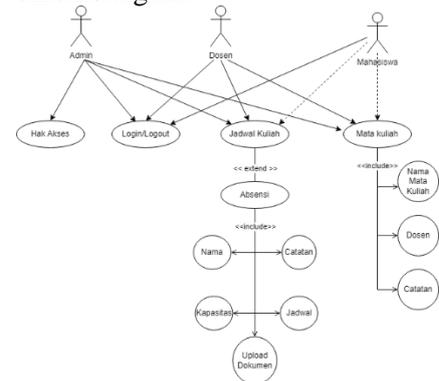
Adapun fitur yang memerlukan keterampilan implementasi yang lebih efektif seperti fitur pelacak lokasi (*Google Maps*) dan *Image Capture* mahasiswa tidak akan digunakan pada website ini.

#### b. Visualisasi Kebutuhan

Rancangan kebutuhan website digambar menggunakan UML (*Unified Modeling Language*). Diagram yang digunakan meliputi *Use case diagram* dan *class diagram*.

#### 1. Use Case Diagram

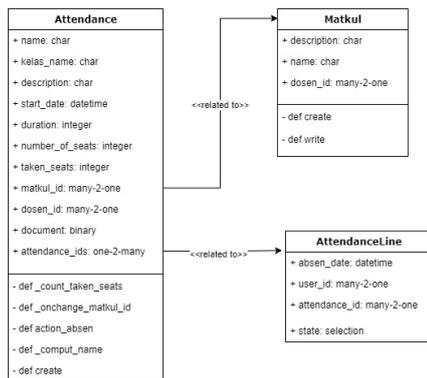
*Use case diagram* digunakan untuk mengilustrasikan bagaimana interaksi antara aktor dengan sistem yang sedang dibangun. Dengan kata lain, diagram ini memvisualisasikan fungsi-fungsi yang ada dalam sistem dan siapa saja yang dapat mengakses fungsi-fungsi tersebut. Gambar diagram 4.1 di bawah ini menunjukkan *use case diagram* dari website absensi berbasis Odoo 16 yang dikembangkan.



Gambar 1. *Use Case Diagram*

#### 2. Class Diagram

Diagram kelas menggambarkan struktur sistem dengan mendefinisikan kelas-kelas yang akan digunakan dalam pembangunan sistem. Kelas memiliki atribut dan metode, atau operasi. Atribut adalah variabel-variabel yang dimiliki oleh suatu kelas, namun pada pengembangan Odoo lebih umum disebut sebagai *field*, sementara metode atau operasi adalah fungsi-fungsi yang dimiliki oleh suatu kelas. Pembuatan diagram kelas bertujuan untuk memastikan sinkronisasi antara dokumentasi perancangan dan perangkat lunak.



Gambar 2. Class Diagram

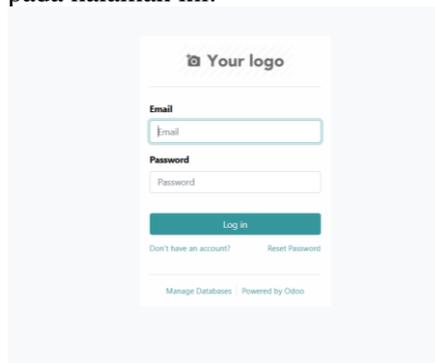
### 4.2.2 Implementasi

Pada tahap implementasi peneliti menggunakan desain dari visualisasi kebutuhan ke pembuatan program modul kustom Odoo dengan menggunakan bahasa pemrograman Python dan XML pada Visual Studio Code. Berikut adalah implementasi Sistem yang dikembangkan.

#### a. Implementasi Sistem

##### 1. Halaman Login

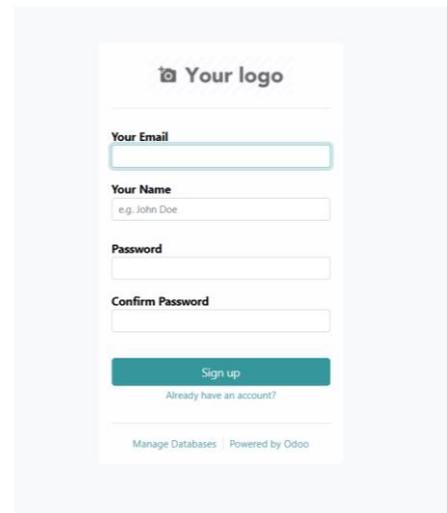
Pada halaman *login*, terdapat form untuk mengisi email dan password dimana pengguna dapat melakukan input data pada form tersebut. Pengguna juga dapat memilih menu “Don’t have an account” dan “reset password” pada halaman ini.



Gambar 3. Tampilan Halaman Login

##### 2. Halaman Sign-in

Ketika pengguna ingin membuat akun, pengguna perlu menekan teks “Don’t have an account” di halaman login. Pengguna akan diarahkan ke halaman pembuatan akun yang terlihat pada gambar:

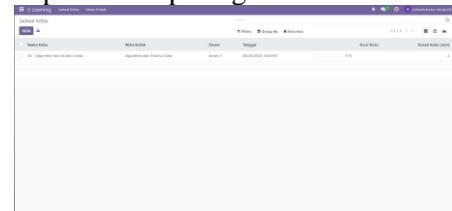


Gambar 4. Tampilan Halaman Sign-in

Pada halaman sign-in/pendaftaran di gambar diatas terdapat form untuk membuat akun. Dengan mengisi email, nama pengguna, password dan konfirmasi ulang password, pengguna dapat membuat akun setelah menekan tombol “Sign in” berdasarkan data yang dimasukkan.

##### 3. Halaman Jadwal Kelas (List)

Halaman jadwal kelas merupakan halaman utama pada aplikasi E-Learning, dimana pengguna akan melihat halaman ini sebagai halaman pertama setelah masuk ke website. Halaman jadwal kelas dengan tampilan tipe *List* dapat dilihat pada gambar:



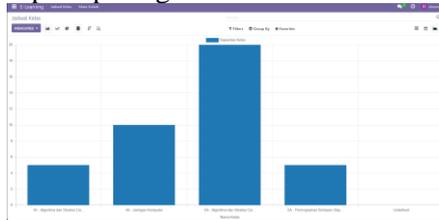
Gambar 5. Tampilan Halaman Jadwal Kelas (Tree)

Pada halaman jadwal kelas terdapat 2 menu *item* pada navbar dan tombol “New” untuk membuat record baru, tombol ini hanya muncul jika pengguna memiliki hak akses untuk membuat *record*. Pada header juga terdapat *search bar* dan terdapat ikon tampilan

yang tersedia pada bagian kanan. Di halaman ini tampil *record* yang telah dibuat dengan tampilan *list* atau daftar.

4. Halaman Jadwal Kelas (Graph)

Pengguna dapat mengganti tampilan *record* yang ingin ditampilkan pada halaman jadwal kelas dengan menekan tombol ikon *Graph* yang ada di bagian kanan atas. Tampilan grafik pertama yang muncul adalah grafik batang seperti pada gambar:

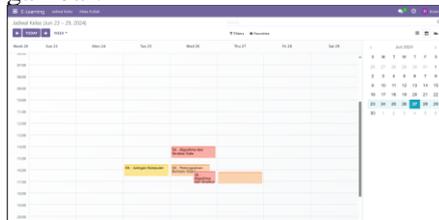


Gambar 6. Tampilan Halaman Jadwal Kelas (*Graph*)

Pada halaman jadwal kelas dengan tampilan grafik, data ditampilkan berdasarkan *field* yang dipilih dengan indikator keterangan tiap graph berupa *field* nama kelas. Pengguna juga dapat mengganti tampilan grafik yang tampil dengan pilihan grafik garis dan grafik lingkaran.

5. Halaman Jadwal Kelas (Calendar)

Pada halaman jadwal kelas pengguna dapat memilih untuk menampilkan *record* dalam tampilan kalender yang secara bawaan menampilkan jadwal kelas pada minggu tersebut seperti pada gambar:



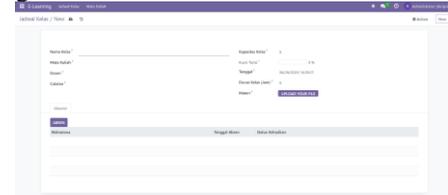
Gambar 7. Tampilan Halaman Jadwal Kelas (*Calendar*)

Pada halaman ini *record* yang sudah dibuat muncul bentuk tampilan kalender yang bisa diatur berdasarkan hari, minggu, bulan atau tahun. Pengguna yang memiliki hak akses juga dapat

menggeser jadwal dengan menggeser *record* jadwalnya.

6. Halaman Absensi (User Dosen)

Halaman ketika pengguna dengan hak akses “Dosen” membuka sebuah *record* dimana pengguna dapat mengisi atau mengubah nilai pada formulir halaman tersebut seperti pada gambar:

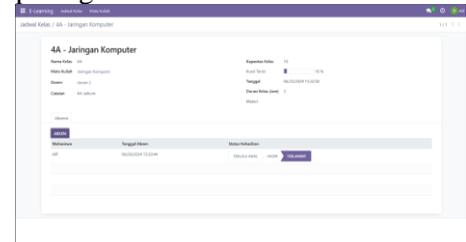


Gambar 8. Tampilan Halaman Absensi (*User Dosen*)

Pada halaman absensi untuk pengguna dengan hak akses “Dosen” pengguna dapat membuat sebuah *record* jadwal kelas baru dengan mengisi formulir yang tersedia. Pengguna dapat mengunggah berkas pada *field* Materi. Pada halaman ini *field* dosen akan secara otomatis terisi berdasarkan *field* mata kuliah jika memiliki nilai pada *field* dosen pengampu.

7. Halaman Absensi (User Mahasiswa)

Pengguna dengan hak akses mahasiswa dapat melakukan absensi dengan membuka *record* yang akan menampilkan halaman pada gambar:



Gambar 9. Tampilan Halaman Absensi (*User Mahasiswa*)

Pada halaman absensi untuk user Mahasiswa dapat menekan tombol “Absen” untuk mencatat kehadiran sesuai waktunya, status kehadiran akan menyesuaikan secara otomatis dengan waktu pengguna melakukan absen. Ketika

pengguna melakukan absensi, *field* kursi terisi akan berubah secara otomatis berdasarkan kapasitas kelas dan pengguna yang telah melakukan absensi.

8. Halaman Mata Kuliah

Pengguna dapat menampilkan halaman mata kuliah dengan menekan salah satu *menu item* pada navbar yang akan menampilkan halaman seperti di gambar:



Gambar 10. Tampilan Halaman List Mata Kuliah

Pada halaman mata kuliah terdapat tombol “New” untuk membuat record baru, tombol ini hanya muncul jika pengguna memiliki hak akses untuk membuat *record*. Pada header juga terdapat *search bar* dan tab filter untuk menyaring pencarian.

9. Halaman Record Mata Kuliah

Halaman *record* mata kuliah tampil ketika pengguna membuka atau membuat *record* baru dengan dengan tampilan seperti pada gambar:

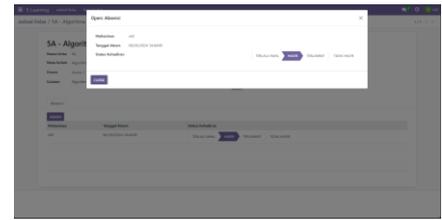


Gambar 11. Tampilan Halaman Mata Kuliah

Pada halaman mata kuliah, pengguna dapat melihat detail *record* mata kuliah. Pengguna dengan hak akses “Dosen” mengedit dapat mengubah keterangan yang ada pada record tersebut.

10. Pop-up Wizard riwayat absensi

Ketika pengguna tanpa akses edit menekan *record* riwayat absensi maka akan muncul tampilan pop-up pada gambar:



Gambar 12. Tampilan Pop-up Wizard Riwayat Absensi

Tampilan Pop-up *wizard* muncul untuk memberikan informasi terkait suatu record didalam suatu halaman secara detail, pada halaman ini *wizard* menampilkan data *record* absensi mahasiswa.

4.2.3 Testing

Black box testing adalah metode pengujian yang dilakukan dengan mengamati hasil eksekusi melalui data uji dan memeriksa fungsionalitas perangkat lunak. Pengujian ini hanya mengevaluasi tampilan antarmuka.

1. Pengujian halaman Mata Kuliah

Data pengujian yang dilakukan untuk halaman Mata Kuliah oleh pengguna “Dosen” dan “Mahasiswa” ditampilkan pada tabel di bawah.

Tabel 1. Pengujian di Halaman Mata Kuliah

Aktor	Pengujian	Hasil yang diharapkan	Hasil
Dosen	Membuat <i>record</i> baru	Tombol “New” muncul dan menampilkan form pembuatan mata kuliah	Sesuai Harapan
Dosen	Mengisi form mata kuliah	Semua <i>field</i> dapat diisi	Sesuai Harapan
Mahasiswa	Tidak dapat membuat <i>record</i>	Tombol “New” tidak ada	Sesuai Harapan
Mahasiswa	Menampilkan daftar <i>record</i>	Membuka halaman mata kuliah pada <i>menuitem</i>	Sesuai Harapan
Mahasiswa	Menampilkan isi <i>record</i>	Membuka halaman form Mata Kuliah	Sesuai Harapan

Berdasarkan data pada tabel pengujian, seluruh pengujian untuk

halaman mata kuliah sesuai dengan harapan.

2. Pengujian Halaman Jadwal Kelas

Data pengujian yang dilakukan untuk halaman Jadwal Kelas oleh pengguna “Dosen” dan “Mahasiswa” dilampirkan pada tabel di bawah.

Tabel 2. Pengujian di Halaman Jadwal Kelas

Aktor	Pengujian	Hasil yang diharapkan	Hasil
Dosen	Membuat <i>record</i> baru	Tombol “New” muncul dan menampilkan form pembuatan Jadwal Kelas	Sesuai Harapan
Dosen	Mengisi form jadwal kelas	Semua <i>field</i> dapat diisi	Sesuai Harapan
Dosen	Mengupload dokumen di form jadwal kelas	<i>Field</i> materi pada form Jadwal Kelas dapat mengunggah berkas	Sesuai Harapan
Dosen	Mengganti jadwal pada <i>view</i> kalender dengan menggeser <i>record</i>	<i>Field</i> tanggal berhasil berubah sesuai tanggal	Sesuai Harapan
Mahasiswa	Menampilkan isi <i>record</i>	Membuka halaman form Jadwal Kelas	Sesuai Harapan
Mahasiswa	Tidak dapat membuat <i>record</i>	Tombol “New” tidak ada	Sesuai Harapan
Mahasiswa	Tidak dapat menggeser jadwal pada <i>view</i> <i>calendar</i>	Muncul peringatan tentang batas hak akses ‘mahasiswa’	Sesuai Harapan
Dosen, Mahasiswa	Menampilkan <i>view</i> <i>graph</i>	Tampilan grafik muncul dan berubah sesuai tipe grafik yang dipilih	Sesuai Harapan
Dosen, Mahasiswa	Menampilkan data tertentu pada <i>view</i> <i>graph</i>	Tampilan grafik berubah berdasarkan <i>field</i> yang dipilih	Sesuai Harapan

Berdasarkan data pada tabel pengujian, seluruh pengujian untuk halaman jadwal kelas berhasil memenuhi harapan.

3. Pengujian Absensi Mahasiswa

Data pengujian yang dilakukan untuk melakukan absensi oleh pengguna “mahasiswa” dilampirkan pada tabel di bawah.

Tabel 3. Pengujian untuk absensi

Aktor	Pengujian	Hasil yang diharapkan	Hasil
Mahasiswa	Tombol “absensi” berfungsi	Tombol dapat ditekan dan mencatat data mahasiswa ke model AttendanceLine	Sesuai Harapan
Mahasiswa	Status kehadiran “Terlalu awal”	Status kehadiran “Terlalu awal” terpilih ketika tombol absen ditekan sebelum jadwal kelas	Sesuai Harapan
Mahasiswa	Status kehadiran “Hadir”	Status kehadiran “Hadir” terpilih ketika tombol absen ditekan saat jadwal kelas	Sesuai Harapan
Mahasiswa	Status kehadiran “Terlambat”	Status kehadiran “Terlambat” terpilih ketika tombol absen ditekan setelah jadwal kelas	Sesuai Harapan

Berdasarkan data pada tabel diatas, seluruh pengujian oleh pengguna mahasiswa untuk melakukan absensi berhasil memenuhi harapan.

4. Pengujian Login dan sign-in

Data pengujian yang dilakukan untuk melakukan *login* dan *sign-in* oleh pengguna “mahasiswa” dilampirkan pada table di bawah.

Tabel 4. Pengujian *Login*

Aktor	Pengujian	Hasil yang diharapkan	Hasil
User	Form sign-in dapat diisi	Pengguna dapat mendaftarkan akun dan langsung login	Sesuai Harapan
User	Form login dapat diisi	Pengguna dapat masuk dengan akun yang didaftarkan	Sesuai Harapan

Berdasarkan data pada tabel diatas, pengujian untuk menggunakan fitur bawaan reset *password* tidak sesuai

harapan karena permasalahan teknis berupa server *outgoing* yang tidak diterapkan

#### 4.2.4 Dokumentasi

Selama proses pengembangan, panduan pengguna telah dibuat dan disimpan di *Google Drive* untuk memudahkan akses oleh pengguna. Panduan pengguna ini mencakup langkah-langkah menggunakan Website Odoo 16 dengan aplikasi absensi.

Sistem versi kontrol Git juga telah digunakan untuk melacak perubahan kode. Beberapa branch telah dibuat di GitHub untuk mengorganisasi pekerjaan pengembangan. Berikut adalah beberapa *branch* yang dibuat:

- main: *Branch* utama yang berisi versi stabil dari proyek.
- development: *Branch* yang digunakan untuk pengembangan fitur baru dan perbaikan bug sebelum diintegrasikan ke branch utama

#### 4.2.5 Peluncuran

Setelah selesai melakukan pengembangan dan pengujian sistem di lingkungan lokal, langkah berikutnya adalah meluncurkan sistem ke server *live* agar dapat diakses oleh pengguna sesungguhnya. Pengguna dapat memberikan pendapat dan masukan setelah menggunakan *website* melalui kuesioner yang disebar.

##### a. Live Server

Dalam penelitian ini, server *live* menggunakan instance EC2 (*Elastic Compute Cloud*) Amazon Web Service yang diinstal dengan AMI (*Amazon Machine Image*) Bitnami untuk Odoo 16. Pemilihan EC2 AWS dilakukan karena keandalannya dalam menyediakan infrastruktur *cloud* yang terstruktur dan aman.

Proses instalasi dan konfigurasi dilakukan sebagai berikut:

- Membuat *Instance* EC2: *Instance* EC2 dengan spesifikasi yang sesuai.
- Menggunakan AMI Bitnami: AMI Bitnami untuk Odoo 16 digunakan untuk mempermudah instalasi dan konfigurasi Odoo di *instance* EC2.
- Konfigurasi Jaringan: Pengaturan keamanan dan jaringan dilakukan untuk memastikan sistem dapat diakses secara aman dari internet.
- Migrasi Data: Data dari lingkungan pengembangan diimpor ke server *live*

untuk memastikan kelangsungan operasi.

- Pengujian Akhir: Pengujian dilakukan di server *live* untuk memastikan semua fitur berfungsi dengan baik sesuai dengan pengembangan.

Setelah peluncuran server selesai, pengguna dapat mengakses *website* dengan menggunakan alamat IP publik <http://13.211.188.98>.

##### b. Kuisisioner

Untuk mengevaluasi tingkat kegunaan (*usability*) dari sistem yang telah dikembangkan, dilakukan pengumpulan data melalui kuesioner yang didistribusikan kepada pengguna sistem. Kuesioner ini didasarkan pada pernyataan dari *usability testing*, yang bertujuan untuk mengukur berbagai aspek kegunaan sistem, termasuk kemudahan penggunaan, efisiensi, kepuasan pengguna, dan keandalan. Adapun indikator pada *Usability Testing* dapat dilihat pada table berikut ini.

**Tabel 5.** Indikator Usability Testing

No	Indikator	Keterangan
1	<i>Learnability</i>	Untuk mengevaluasi seberapa mudah pengguna dapat mempelajari <i>website</i> untuk menyelesaikan tugas.
2	<i>Memorability</i>	Untuk mengukur sejauh mana pengguna dapat mengingat cara menggunakan <i>website</i> setelah periode waktu tertentu.
3	<i>Efficiency</i>	Untuk menilai seberapa efisien pengguna dalam menyelesaikan tugas pada <i>website</i> .
4	<i>Errors</i>	Untuk mengukur jumlah kesalahan yang terjadi dan bagaimana pengguna mengatasinya.
5	<i>Satisfaction</i>	Untuk menggambarkan tingkat kepuasan pengguna dalam menggunakan <i>website</i> .

Pertanyaan-pertanyaan dalam kuesioner berbasis *usability testing* yang disebar dapat dilihat pada tabel berikut.

**Tabel 6.** Daftar Pertanyaan

Indikator	Kode	Pertanyaan
Learnability (A)	(A1)	Website Absensi Berbasis Odoo 16 dapat dipelajari dengan mudah
	(A2)	Saya mampu dengan mudah memahami dan mengerti alur dari navigasi yang ada pada Website Absensi Berbasis Odoo 16
	(A3)	Tanpa instruksi tertulis atau <i>Manual Book</i> , saya dapat menemukan dan menggunakan fitur absensi di website Absensi Berbasis Odoo 16
Memorability (B)	(B1)	Saya dengan mudah mengetahui dan mengingat arah navigasi dan fitur pada Website Absensi Berbasis Odoo 16
	(B2)	Saya dapat dengan mudah mengingat dan menggunakan fitur absensi Website Absensi Berbasis Odoo 16
	(B3)	Saya merasa mudah menggunakan Website Absensi Berbasis Odoo 16 kapan saja.
Efficiency (C)	(C1)	Saya mampu melakukan akses menu absensi pada Website Absensi Berbasis Odoo 16 dengan cepat
	(C2)	Saya mampu mengakses dan menggunakan fitur absensi pada Website Absensi Berbasis Odoo 16 dengan efisien.
	(C3)	Saya mampu langsung menemukan informasi yang saya ingin cari dari awal membuka pada Website Absensi Berbasis Odoo 16
Errors (D)	(D1)	Saya tidak menemukan error disaat absen menggunakan Website Absensi Berbasis Odoo 16
	(D2)	Pada aplikasi E-Learning di website, Saya tidak menemukan menu yang error atau tidak sesuai dengan fungsinya
	(D3)	Saya dapat menemukan fitur dan menu yang saya cari pada

Indikator	Kode	Pertanyaan
Satisfaction (E)		Website Absensi Berbasis Odoo 16
	(E1)	Saya merasa nyaman menggunakan Website Absensi Berbasis Odoo 16
	(E2)	Saya senang dengan design antarmuka yang ada pada Website Absensi Berbasis Odoo 16 secara keseluruhan
	(E3)	Paduan warna dan tata letak konten nyaman untuk dilihat

Responden diminta untuk memberikan penilaian berdasarkan Skala Likert. Berikut adalah table-table yang menampilkan Skala Likert dan parameter bobot nilai untuk tiap-tiap indikator.

**Tabel 7.** Tabel Skala Likert

Kode	Pilihan Jawaban	Skor/Nilai
ST	SANGAT SETUJU	5
S	SETUJU	4
N	NETRAL	3
TS	TIDAK SETUJU	2
STS	SANGAT TIDAK SETUJU	1

**Tabel 8.** Parameter Bobot Nilai

Bobot Nilai	Keterangan
80% – 100%	SANGAT SETUJU
60% – 80%	SETUJU
40% – 60%	NETRAL
20% – 40%	TIDAK SETUJU
0% – 20%	SANGAT TIDAK SETUJU

Berdasarkan Skala Likert yang peneliti gunakan dalam penelitian ini, langkah yang harus dilakukan adalah mengalikan setiap jawaban dengan bobot nilai yang telah ditentukan lalu jumlahkan. Kemudian tentukan nilai indeks minimal dan nilai indeks maksimal atau kita singkat menjadi nilai X dan Y.

Setelah mendapat nilai X dan Y maka yang harus dilakukan adalah mencari total skor dari masing-masing pertanyaan pada setiap indikator *usability* dengan menggunakan rumus indeks untuk menentukan hasil persentase nilai indikator. Rumusnya dapat dilihat pada gambar di bawah:

$$\text{Total Skor} / Y \times 100$$

**Gambar 13.** Rumus Hasil Presentase Nilai Skala Likert (Sumber: Sugiyono, 2007)

Untuk menghitung hasil akhir dapat dilakukan dengan menjumlahkan seluruh nilai dari indikator lalu dibagi dengan kelima indikator usability tersebut. Berikut rumusnya pada gambar:

$$N = \frac{\text{HNI}}{\text{JI}}$$

**Gambar 14.** Rumusan Nilai Akhir  
Keterangan:

N = Nilai Akhir

HNI = Hasil Nilai Indikator

JI = Jumlah Indikator

#### c. Evaluasi User

Berdasarkan hasil dari pertanyaan kuesioner yang dijawab oleh 40 responden, maka dapat dilakukan perhitungan dari seluruh jawaban pertanyaan dari responden.

Berdasarkan data yang diperoleh, dapat diketahui bahwa nilai total untuk indikator *Learnability* adalah 524, *Memorability* bernilai total 522, *Efficiency* bernilai total 524, *Errors* bernilai total 523 dan *Satisfaction* bernilai total 507 dimana nilai maksimal untuk indikator adalah 600. Dengan menggunakan rumus hasil persentase indikator dapat diketahui persentase masing-masing indikator sebagai berikut:

- *Learnability* =  $(524/600) \times 100 = 87.3\%$
- *Memorability* =  $(522/600) \times 100 = 87\%$
- *Efficiency* =  $(524/600) \times 100 = 87.3\%$
- *Errors* =  $(523/600) \times 100 = 87.1\%$
- *Satisfaction* =  $(507/600) \times 100 = 84.5\%$

Setelah didapatkan nilai hasil rata-rata persentase dari setiap indikator usability

*testing* yang kemudian ditotalkan dan dihitung menggunakan rumus nilai akhir yang merupakan gabungan semua indikator maka didapatkan hasil berikut:

$$\text{Nilai Akhir} = \text{Hasil Nilai Indikator} / \text{Jumlah Indikator} = (87.3\% + 87\% + 87.3\% + 87.1\% + 84.5\%) / 5 = 433.2\% / 5 = 86.64\%$$

Jadi dapat disimpulkan dari hasil akhir pencapaian aplikasi absensi Odoo 16 dengan nilai 86.64 % yang berarti aplikasi absensi sangat baik dalam artian mudah dalam hal menggunakannya sebagai alat absensi berbasis Odoo 16

#### 4.2.6 Pemeliharaan

Tahap pemeliharaan merupakan bagian penting dari siklus hidup sistem yang bertujuan untuk memastikan bahwa sistem tetap berfungsi dengan baik dan memenuhi kebutuhan pengguna. Aktivitas pemeliharaan meliputi pemantauan kinerja sistem, penanganan bug, pembaruan fitur, dan memberikan dukungan teknis kepada pengguna.

## 5 KESIMPULAN

Berdasarkan hasil dan analisis pembahasan yang telah dilakukan maka dapat diambil kesimpulan bahwa:

1. Penelitian ini merancang modul aplikasi absensi Odoo 16 menggunakan metode metodologi SDLC (Software Development Life Cycle) dengan model *Agile*. Model *Agile* dipilih karena memberikan fleksibilitas yang tinggi dan memungkinkan penyesuaian cepat terhadap perubahan kebutuhan dan umpan balik pengguna.
2. Aplikasi absensi berbasis Odoo dapat diimplementasikan dengan menggunakan bahasa pemrograman *Python* untuk struktur logika Odoo dan XML untuk mendefinisikan struktur data.
3. Hasil dari evaluasi pengguna dengan menggunakan *usability testing* dengan 5 indikator yaitu 1) *Learnability* mendapatkan nilai 87.3%, 2) *Memorability* mendapatkan nilai 87%, 3) *Efficiency*

mendapatkan nilai 87.3%, 4) *Errors* mendapatkan nilai 87.1%, 5) *Satisfaction* mendapatkan nilai 84,5%. Dengan hasil akhir dari pengembangan aplikasi absensi Odoo 16 berhasil mendapatkan nilai 86.64% yang berarti fungsi aplikasi absensi berhasil dijalankan oleh pengguna dengan baik.

Adapun saran yang dapat peneliti sampaikan sebagai berikut :

1. Pengembangan aplikasi dapat diterapkan kepada mahasiswa/i untuk mempermudah pendataan administrasi mahasiswa.
2. Pada penelitian selanjutnya diharapkan dapat meningkatkan kualitas tampilan yang lebih baik dengan menggunakan tema yang lebih menarik.
3. Pada penelitian selanjutnya diharapkan dapat meningkatkan kualitas fungsi yang lebih efektif dengan menambahkan fitur pengambilan gambar atau pencatatan Lokasi.

### UCAPAN TERIMA KASIH

Penulis menyampaikan terima kasih kepada kedua orang tua yang selalu memberikan *support* dan doa restu dalam setiap langkah penelitian ini, serta kepada dosen pembimbing yang telah membantu dalam penelitian yang dilakukan.

### Daftar Pustaka

- [1] Kadek Adhitya Dwi Pratama, Gusti Made Arya Sasmita, Ni Made Ika Marini Mandenni (2021). Implementation of Laboratory Services and Online Consultation at Taksu Laboratory Clinic by using ODOO ERP.
- [2] Demilda, Yasa Ewa, Ary Arvianto, and Zainal Fanani Rosyada (2022). "Implementasi Software ODOO Dengan Menggunakan Modul Accounting, Inventory, Purchase, Dan Point Of Sales Pada Toko Al Hikmah Mart (Ah Mart) Di Bogor Jawa Barat." *Industrial Engineering Online Journal* 11.4.
- [3] Ristyawati, N. K. Y., Purnawan, I. K. A., & Sasmita, G. M. A. (2020). "The Implementation of Enterprise Resource Planning ERP on Sales Management Module using Odoo 11. *International Journal of Trend in Scientific Research and Development*," 4(4), 1-4.
- [4] Devkota, A. (2016). Open ERP Odoo guidebook for small and medium enterprises. 34. <https://www.theseus.fi/handle/10024/119606>.
- [5] Andini, A. F., Irzal, M., & Arafiyah, R. (2017). Perancangan dan Implementasi Sistem Absensi Online Berbasis Android di Lingkungan Universitas Negeri Jakarta. *J-KOMA: Jurnal Ilmu Komputer dan Aplikasi*
- [6] Munawaroh, Siti (2005). "Mengeksplorasi Database PostgreSQL dengan PgAdmin III." *Dinamik* 10.2.
- [7] Kartiko, Condro (2019). "Evaluasi kualitas aplikasi web pemantau menggunakan model pengujian perangkat lunak ISO/IEC 9126." *Jurnal Nasional Teknik Elektro dan Teknologi Informasi* 8.1: 16-23.
- [8] Verma, Akanksha, Amita Khatana, and Sarika Chaudhary (2017). "A comparative study of black box testing and white box testing." *International Journal of Computer Sciences and Engineering* 5.12: 301-304.
- [9] Aiyegbusi, Olalekan Lee (2020). "Key methodological considerations for usability testing of electronic patient-reported outcome (ePRO) systems." *Quality of life research* 29: 325-333.
- [10] Zulfa Basmallah Agustyn, Satrio Wibowo, Aqil Jawadal Furqon, "Usability Testing Aplikasi GetContact Menggunakan Metode Post Study System Usability Questionnaire (PSSUQ)," *JITET: Jurnal Informatika dan Teknik Elektro Terapan*, vol. 2, no. 12, p. 830, 2024.
- [11] Sumartini, S., Harahap, K. S., & Sthevany, S. (2020). Kajian pengendalian mutu produk tuna loin precooked frozen menggunakan metode skala likert di perusahaan pembekuan tuna. *Aurelia Journal*, 2(1), 29-38.
- [12] Sonata, F. (2019). Pemanfaatan UML (Unified Modeling Language) dalam perancangan sistem informasi e-commerce jenis customer-to-customer. *Jurnal Komunika: Jurnal Komunikasi, Media Dan Informatika*, 8(1), 22-31.

- [13] Jacobson, L., & Booch, J. R. G. (2021).  
The unified modeling language reference  
manual.