Jurnal Ilmiah Sistem Informasi dan Ilmu Komputer
Volume. 6 Nomor. 2 Juli 2026
p-ISSN: 2827-8135; e-ISSN: 2827-7953; Hal 244-267
DOI: https://doi.org/10.55606/juisik.v6i2.2493
Tersedia: https://journal.sinov.id/index.php/juisik

Perancangan dan Pembuatan Aplikasi Sistem Kontrol Absensi Mahasiswa Politeknik Negeri Kupang

(Studi Kasus pada Jurusan Teknik Elektro)

Jemsrado Sine*, Daniel S Bataona[2], Santi Foenale[3]
[1-3]Program Studi Teknik Komputer dan jaringan, Politeknik Negeri Kupang, Indonesia
*Penulis Korespondensi: jemsrado.sine@gmail.com[1]

Abstract. *The development of a web-based student attendance information system aims to overcome various obstacles inherent in manual systems, such as recording delays, recapitulation errors, and limited data access. This application utilizes web technology to allow online access through a browser, providing convenience for students, lecturers, and administrators to process attendance accurately and efficiently. This system provides main features in the form of attendance recording, lecture scheduling, attendance summary data per course, verification status by lecturers, administrators, and heads of study programs, and class completion by students. All attendance data is stored in real-time, so that it can be accessed directly by related parties to support academic transparency and accountability. This research aims to design and build a web-based student attendance information system application using the Waterfall approach. The research stages include needs analysis, system design, coding, testing, and maintenance. The results of the development show that this system is able to increase the effectiveness of attendance data management, speed up the administrative process, and support more structured academic supervision. Thus, this web-based information system can be an innovative solution in improving the quality of student attendance management in universities.*

Keywords: *Data Real-Time; Information Systems; Student Attendance; Waterfall Approach; Web Technology.*

Abstrak. Pengembangan sistem informasi kehadiran mahasiswa berbasis web bertujuan untuk mengatasi berbagai kendala yang melekat pada sistem manual, seperti keterlambatan pencatatan, kesalahan rekapitulasi, dan keterbatasan akses data. Aplikasi ini memanfaatkan teknologi web sehingga memungkinkan akses secara daring melalui *browser*, memberikan kemudahan bagi mahasiswa, dosen, dan administrator dalam memproses kehadiran secara akurat dan efisien. Sistem ini menyediakan fitur utama berupa pencatatan kehadiran, penjadwalan kuliah, data ringkasan kehadiran per mata kuliah, status verifikasi oleh dosen, administrator, maupun kepala program studi, serta penyelesaian kelas oleh mahasiswa. Seluruh data kehadiran tersimpan secara *real-time*, sehingga dapat diakses langsung oleh pihak terkait untuk mendukung transparansi dan akuntabilitas akademik. Penelitian ini bertujuan merancang dan membangun aplikasi sistem informasi kehadiran mahasiswa berbasis web dengan menggunakan pendekatan *Waterfall*. Tahapan penelitian meliputi analisis kebutuhan, desain sistem, pengkodean, pengujian, dan pemeliharaan. Hasil pengembangan menunjukkan bahwa sistem ini mampu meningkatkan efektivitas pengelolaan data kehadiran, mempercepat proses administrasi, serta mendukung pengawasan akademik yang lebih terstruktur. Dengan demikian, sistem informasi berbasis web ini dapat menjadi solusi inovatif dalam meningkatkan kualitas manajemen kehadiran mahasiswa di perguruan tinggi.

Kata kunci: *Data Real-Time;* Kehadiran Mahasiswa; Pendekatan *Waterfall;* Sistem Informasi; Teknologi Web.

1. LATAR BELAKANG

Salah satu akibat dari perkembangan teknologi yang semakin pesat adalah pengaruhnya pada kemudahan-kemudahan yang diberikan dalam kehidupan sehari-hari terutama dalam bidang pendidikan. Pada bidang pendidikan penerapan dari perkembangan teknologi tersebut terdapat pada sistem presensi dalam kegiatan belajar mengajar terutama di tingkat perguruan tinggi. Dalam proses belajar mengajar, proses pencatatan kehadiran mahasiswa/mahasiswi harus di lakukan setiap hari pada jam perkuliahan. Absensi merupakan salah satu elemen penting dalam kegiatan perkuliahan yaitu sebagai pencatat kehadiran. Daftar hadir merupakan

Naskah Masuk: 21 April 2026; Revisi: 22Mei 2026; Diterima: 23 Juni 2026; Tersedia: 30 Juni 2026

bukti bahwa mahasiswa tersebut telah hadir dalam suatu perkuliahan. Jumlah kehadiran merupakan suatu pertimbangan bagi dosen untuk menentukan nilai bagi mahasiswa.

Sistem informasi absensi mahasiswa pada Politeknik Negeri Kupang terutama jurusan Elektro masih menggunakan sistem manual, sehingga sering terjadi kesalahan berupa hilangnya dokumen absensi, rusaknya dokumen absensi atau seringkali mahasiswa memanfaatkan celah dan bekerjasama dengan mahasiswa lainnya untuk melakukan kecurangan, misalnya mahasiswa yang hadir dalam kelas, sering menandatangani absensi temannya yang tidak hadir dalam perkuliahan, begitupun dengan data absensi masih belum terkomputerisasi karena pengolahannya masih dalam buku absensi. Untuk itu diperlukan sebuah sistem yang dapat menangani masalah diatas, yaitu aplikasi absensi mahasiswa berbasis *website*.

Oleh karena itu, dengan adanya masalah di atas, maka penulis berniat membuat "SISTEM INFORMASI ABSENSI MAHASISWA BERBASIS WEB" dengan tujuan untuk meningkatkan efisiensi dalam pengelolaan absensi, meminimalkan kesalahan, mempermudah absensi mahasiswa setiap hari maupun membantu admin dalam membuat rekap absensi mahasiswa.

2. KAJIAN TEORITIS

Basis Data

Basis data (database) adalah kumpulan terorganisir dari data yang saling terkait, yang disimpan dan diakses melalui sistem komputer. Perangkat lunak yang digunakan untuk mengelola dan memanggil *query*, basis data disebut *database management sistem* (DBMS).

Basis data dirancang untuk menyimpan informasi yang dapat diambil, dimutakhirkan, dan dikelola dengan efisien. Basis data digunakan secara luas dalam berbagai aplikasi dan sistem, termasuk sistem informasi, situs *web*, sistem manajemen pelanggan, sistem manajemen inventaris, dan banyak lagi.

Berikut ini beberapa konsep dasar terkait basis data :

a. Struktur Data : Basis data terdiri dari tabel, yang terdiri dari baris (*rekord*) dan kolom (*atribut*). Setiap baris dalam tabel mewakili entitas tunggal atau objek, sementara setiap kolom menyimpan *atribut* atau informasi spesifik tentang entitas tersebut. Misalnya, dalam sebuah basis data pelanggan, tabel "Pelanggan" dapat memiliki kolom seperti "Nama", "Alamat", "Nomor Telepon", dan lain-lain.
b. Kunci : Kunci digunakan untuk mengidentifikasi unik setiap entitas dalam tabel. Kunci utama (*primary key*) adalah kunci yang unik untuk setiap baris dalam tabel dan digunakan

untuk mengidentifikasi entitas tersebut. Kunci asing (*foreign key*) adalah kunci yang menghubungkan entitas antara tabel yang berbeda dalam basis data.

c. Relasi antar tabel: Basis data relasional adalah jenis basis data yang paling umum digunakan. Dalam basis data relasional, tabel terkait saling berhubungan melalui kunci asing. Misalnya, dalam basis data toko online, tabel "Produk" dan "Pesanan" dapat berhubungan melalui kunci asing "ID Produk".
d. Query : *Query* digunakan untuk mengambil data dari basis data. *Query* memungkinkan pengguna untuk menyusun pertanyaan atau pernyataan yang mengambil data yang diperlukan dari satu atau beberapa tabel. Bahasa yang umum digunakan untuk mengambil data dari basis data relasional adalah SQL (*Structured Query Language*).
e. Integritas Data: Integritas data adalah aspek penting dalam basis data. Integritas data melibatkan pemeliharaan kebenaran, konsistensi, dan validitas data dalam basis data. Hal ini dapat dicapai melalui penggunaan aturan validasi, batasan, dan relasi yang telah ditentukan.
f. Manajemen Basis Data : Manajemen basis data melibatkan pemeliharaan, pengelolaan, dan pengoptimalan basis data. Ini meliputi pembuatan, pembaruan, penghapusan, pemulihan data, serta pengaturan keamanan dan *backup* data.

Diagram Alir (Flowchart)

Flowchart adalah penggambaran secara grafik dari langkah-langkah dan urut-urutan prosedur dari suatu program. *Flowchart* menolong analis dan programmer untuk memecahkan masalah kedalam segmen-segmen yang lebih kecil dan menolong dalam menganalisis alternatif-alternatif lain dalam pengoperasian. *Flowchart* biasanya mempermudah penyelesaian suatu masalah khususnya masalah yang perlu dipelajari dan dievaluasi lebih lanjut.

Flowchart adalah bentuk gambar atau diagram yang mempunyai aliran satu atau dua arah secara sekuensial. *Flowchart* digunakan untuk merepresentasikan maupun mendesain program. Oleh karena itu *flowchart* harus bisa merepresentasikan komponen-komponen dalam bahasa pemrograman.

Bagan alir ini memperlihatkan urutan proses dalam sistem dengan menunjukan alat media input, output, serta jenis media penyimpanan dalam proses pengolahan data (Ladjamudin, 2005:211). Adapun simbol-simbol *flowchart* sistem yang digunakan dapat dilihat pada Tabel di bawah.

Tabel 1. Simbol-simbol *Flowchart* Sistem.

Simbol	Nama	Keterangan
	Proses	Menunjukan kegiatan proses dari operasi program komputer
	Manual Input *Keyboard*	Menunjukan *Input* yang dilakukan secara manual
	Keputusan	Digunakan untuk menyelesaikan kondisi didalam program
	Input/Output	Digunaakan untuk menyatakan proses *Input* dan *Output* tanpa tergantung dengan jenis peralatannya
	Garis Alir	Digunakan untuk menunjukan arus dari proses
	Titik Terminal	Digunakan untuk menunjukan awal dan akhir dari suatu proses
	Manual	Untuk menyatakan suatu tindakan (proses) yang tidak dilakukan oleh komputer (manual)
	Disk Storage	Digunakan untuk menyatakan *input* berasal dari *disk* atau *output* disimpan ke *disk*
	Document	Digunakan untuk mencetak laporan ke printer

Use Case

Use case diagram adalah satu dari berbagai jenis diagram UML (Unified Modelling Language) yang menggambarkan hubungan interaksi antara sistem dan aktor. Use Case dapat mendeskripsikan tipe interaksi antara si pengguna sistem dengan sistemnya. Adapun manfaat dari use case di antaranya:

a. Menggunakannya sebagai kebutuhan verifikasi.
b. Menjadi gambaran *interface* dari sebuah sistem karena setiap sistem yang dibangun haruslah memiliki *interface*.
c. Mengidentifikasi siapa saja orang yang dapat berinteraksi dengan sistem, serta apa yang dapat dilakukan oleh sistem.
d. Memberikan kepastian mengenai kebutuhan sistem, sehingga tidak membingungkan.
e. Memudahkan proses komunikasi antara domain *expert* dan end *user*.

Komponen-komponen pada use case di antaranya sebagai berikut :

a. Sistem

Sebuah sistem digambarkan ke dalam bentuk persegi. Fungsinya untuk membatasi *use case* dengan interaksi dari luar sistem. Sistem pada umumnya diberikan label yang sesuai. Namun, umumnya sistem ini tidaklah diberi gambar karena kita tidak terlalu memberikan makna pada sebuah diagram.

b. Actor

Peran *actor* sangat penting, tentunya menciptakan *use case* jadi lebih mudah. Fungsi *Actor* menjelaskan siapa yang berinteraksi dengan sistem. *Actor* akan memberikan informasi kepada sistem, serta menerima informasi dari sistem. Keduanya bisa terjadi secara bersamaan.*Actor* tidak memberikan kontrol terhadap sistem, namun hanya memberikan gambaran mengenai hubungannya dengan sistem.

Simbol	**Keterangan**
	Aktor : Mewakili peran orang, sistem yang lain, atau alat ketika berkomunikasi dengan *use case*
	Use case : Abstraksi dan interaksi antara sistem dan aktor
	Association : Abstraksi dari penghubung antara aktor dengan use case
	Generalisasi : Menunjukkan spesialisasi aktor untuk dapat berpartisipasi dengan use case
<<include>>	Menunjukkan bahwa suatu use case seluruhnya merupakan fungsionalitas dari use case lainnya
<<extend>>	Menunjukkan bahwa suatu use case merupakan tambahan fungsional dari use case lainnya jika suatu kondisi terpenuhi

Gambar 1. Dafar Simbol-Simbol Use Case.

Activity Diagram

Activity diagram dalam bahasa Indonesia diagram aktivitas, yaitu diagram yang dapat memodelkan proses-proses yang terjadi pada sebuah sistem. Runtutan proses dari suatu sistem digambarkan secara vertikal. *Activity diagram* merupakan pengembangan dari *Use Case* yang memiliki alur aktivitas. Alur atau aktivitas bisa berupa runtutan menu-menu atau proses bisnis yang terdapat di dalam sistem tersebut. *Activity diagram* berfungsi juga untuk menganalisis

diagram *use case* dengan cara mendeskripsikan aktor, tindakan yang perlu dilakukan, dan kapan harus terjadi.

Berikut beberapa tujuan dari *activity diagram :*

a. Menjelaskan urutan aktivitas dalam suatu proses.
b. Mudah dalam memahami proses yang ada dalam sistem secara keseluruhan dan merupakan metode perancangan yang terstruktur, mirip dengan *Flowchart* maupun *Data Flow Diagram* (DFD).
c. Mengetahui aktivitas aktor pengguna berdasarkan *use case* diagram yang dibuat sebelumnya.

NO	GAMBAR	NAMA	KETERANGAN
1		*Actifity*	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2		*Action*	State dari sistem yang mencerminkan eksekusi dari suatu aksi
3		*Initial Node*	Bagaimana objek dibentuk atau diawali.
4		*Actifity Final Node*	Bagaimana objek dibentuk dan dihancurkan
5		*Fork Node*	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran

Gambar 2. Dafar Simbol-Simbol *Activity Diagram.*

Berikut penjelasan lengkapnya mengenai komponen-komponen pada *activity diagram* diatas: *Start Point* adalah lingkaran hitam kecil. Biasanya digunakan untuk menandakan status awal, tindakan awal, atau titik awal aktivitas untuk setiap *activity diagram*. *Activity* (Aktivitas): *Activity* merupakan aktivitas yang dilakukan atau sedang terjadi dalam sistem. Biasanya diawali dengan "kata kerja" dari aktivitas yang dilakukan. Percabangan atau *decision* merupakan suatu titik atau *point* yang mengindikasikan suatu kondisi di mana adanya kemungkinan dalam perbedaan transisi. Hal tersebut diperlukan ketika sistem yang dimiliki memiliki beberapa kemungkinan atau jalan alternatif. *Synchronization* dibagi menjadi 2 bagian, yaitu *fork* dan *join*.

a. *Fork* (percabangan) digunakan untuk memecah *behaviour* (tingkah laku) menjadi *activity* atau *action* (aksi) secara paralel.
b. *Join* untuk menghubungkan kembali *activity* dengan *action* secara paralel.

c. *Merge:* Menggabungkan *flow* yang sudah dipecah menjadi beberapa bagian oleh suatu *flow*.
d. *Swimlanes:* Memecah *activity diagram* menjadi kolom dan baris untuk membagi tanggung jawab objek-objek yang melakukan suatu aktivitas.
e. *Transition* Digunakan untuk menunjukan aktivitas selanjutnya dan sebelumnya.
f. Notasi akhir digunakan untuk menandakan proses tersebut berakhir. Pada UML, notasi akhir dapat digambarkan dengan simbol sebuah *bull's eye* (mata sapi).

Terkadang *decision* dengan *fork* adalah hal yang keliru, sebab *Decision* digunakan untuk memecah aktivitas yang bersifat kondisional. Contohnya adalah pilihan Ya atau Tidak, jika opsi Ya maka akan terjadi aksi baru dan bila jika Tidak, maka menolak aksi baru. Sedangkan *fork* digunakan untuk memecah *behaviour* menjadi aktivitas yang *paralel*, contohnya seperti pengguna dapat memilih, menambah, mengubah, serta bisa juga menghapus.

Website

Website adalah kumpulan dari berbagai macam halaman situs, yang terangkum didalam sebuah domain atau juga sub domain, yang lebih tepatnya berada di dalam www (*Word Wide Web*) yang terdapat di dalam internet. Halam web biasanya berupa dokumen yang ditulis dalm format *Hyper Text Markup Language* (HTML), yang bisa diakses melalui HTTP, HTTP adalah suatu protokol yang menyampaikan berbagai informasi dari server *website* untuk ditampilkan kepada para user atau pemakai melalui *web browser*.

Xampp

Xampp adalah aplikasi pengembang yang berguna untuk pengembangan *web* PHP dan MYSQL. *Software* Xampp dibuat dan dikembangkan oleh *Apache Friends*. Perangkat lunak ini berperan sebagai *server web* apache untuk simulasi pengembangan *website*.

Melalui program ini, programmer *web* dapat menguji aplikasi *web* yang di kembangkan dan mempresentasikannya ke pihak lain secara langsung dari komputer, tanpa perlu terkoneksi ke internet.

XAMPP adalah singkatan dari "X" yang menandakan lintas platform (*cross-platform*), "Apache" sebagai *server web*, "MySQL" sebagai sistem manajemen basis data, "PHP" sebagai bahasa pemrograman, dan "Perl" sebagai bahasa pemrograman. XAMPP adalah paket perangkat lunak yang dirancang untuk memungkinkan pengguna untuk membuat dan mengelola lingkungan pengembangan *web* lokal. XAMPP menyediakan bundel perangkat lunak yang diperlukan untuk mengembangkan dan menjalankan aplikasi *web* secara lokal di komputer Anda.

Apache

Apache merupakan *web server* yang digunakan untuk menampilkan *website* di internet seperti menggunakan *mozila fire fox, google crome, IE,* dan lain-lain. Berdasarkan kode-kode yang ditulis di dalam *website* tersebut baik menggunakan bahasa pemrograman HTML maupun PHP yang mengambil suatu database yang dibangun di mysql, sehingga terbentuklah sebuah *website* yang dapat dilihat di internet.

Apache sendiri bersifat *opensource* sehingga dapat digunakan oleh siapa saja dan dikembangkan oleh siapa saja tentunya bagi yang mampu mengembangkannya. *Web server* adalah suatu internet yang menggunakan protocol HTTP (*Hypertext transfer protocol*) untuk melayani semua proses pentransferan data. *Web server* bukan hanya dapat melayani jenis data dalam bentuk text, tetapi juga dalam bentuk gambar dengan jenis 2D atau 3D, suara dan juga dapat berinteraksi dalam dunia *wireless* internet yang menjadikannya sebagai *wireles acces protocol* (WAP) *gateway* dan sebagainya.

Apache juga memiliki program pendukung yang dapat memberikan layanan yang cukup bagi penggunanya diantaranya kontrol akses, *common gateway interface*, PHP dan SSI (*Server Side Include*).

MySQL

MySQL adalah perangkat lunak sistem manajemen basis data SQL atau DBMS yang *multithread* dan *multi-user*. MYSQL dapat digunakan untuk membuat dan mengelola databse beserta isinya. Kita dapat memanfaatkan database untuk menambahkan, mengubah dan menghapus data yang berada dalam database. Mysql merupakan sistem manajemen database yang bersifat relational, artinya data-data yang dikelola dalam database akan diletakan pada beberapa tabel yang terpisah sehingga manipulasi data akan menjadi jauh lebih cepat.

MYSQL adalah *Relational Database Management System* yang di distribusikan secara gratis dibawah lisensi GPL (*general public license*), dimana setiap orang bebas untuk menggunakan MYSQL, namun tidak boleh dijadikan produk turunan yang bersifat komersial. SQL adalah sebuah konsep pengoperasian database, terutama untuk pemilihan atau seleksi dan pemasukan data yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis.

PHP

XAMPP menyertakan versi PHP yang dapat digunakan untuk menjalankan skrip PHP di *server web Apache*. PHP merupakan bahasa pemrograman yang digunakan untuk membuat *website dinamis* yang memungkinkan kita melakukan update setiap saat. Berbeda dengan HTML yang *source codenya* ditampilkan di *website*, *source code* PHP tidak ditampilkan di

halaman muka suatu *website* karena PHP diolah dan di proses di server, PHP bersifat *server-side scripting* yang mampu berjalan di berbagai sistem operasi.

PHP memiliki kedinamisan dalam hal database yang bisa dihubungkan dengan PHP seperti mysql, oracle, ms access, postgreSQL. Namun untuk pemrograman *website* yang paling digunakan adalah mysql.

Perl

Perl adalah bahasa pemrograman skrip serbaguna yang sering digunakan untuk pemrosesan teks dan tugas pemrograman lainnya. XAMPP menyertakan *interpreter* Perl yang dapat digunakan untuk menjalankan skrip Perl.

PHPMyAdmin

PHPMyAdmin adalah antarmuka *web* yang populer untuk mengelola dan mengelola basis data MySQL. XAMPP menyertakan PHPMyAdmin sehingga Anda dapat dengan mudah memanipulasi basis data melalui antarmuka *web*.

XAMPP tersedia secara gratis dan dapat diunduh dari situs resminya. Setelah di instal, anda dapat mengkonfigurasi dan mengelola *server web*, basis data, dan bahasa pemrograman yang disertakan melalui antarmuka pengguna yang mudah digunakan.

UML

UML (*Unified Modelling Language*) adalah suatu metode dalam pemodelan secara visual yang digunakan sebagai sarana perancangan sistem berorientasi objek. Awal mulanya, UML diciptakan oleh *Object Management Group* dengan versi awal 1.0 pada bulan Januari 1997. UML juga dapat didefinisikan sebagai suatu bahasa standar visualisasi, perancangan, dan pendokumentasian sistem, atau dikenal juga sebagai bahasa standar penulisan *blueprint* sebuah *software.*UML diharapkan mampu mempermudah pengembangan piranti lunak (RPL) serta memenuhi semua kebutuhan pengguna dengan efektif, lengkap, dan tepat. Hal itu termasuk faktor-faktor *scalability, robustness, security*, dan sebagainya.Perlu kamu ketahui bahwa sistem yang baik itu berawal dari perancangan dan pemodelan yang matang. Salah satu yang bisa kamu praktekkan, yaitu dengan menggunakan UML. Adapun tujuan dan fungsi perlu adanya UML yaitu sebagai berikut:

a. Dapat memberikan bahasa pemodelan visual atau gambar kepada para pengguna dari berbagai macam pemrograman maupun proses umum rekayasa.
b. Memberikan suatu gambaran model atau sebagai bahasa pemodelan visual yang ekspresif dalam pengembangan sistem.
c. Tidak hanya menggambarkan model sistem *software* saja, namun dapat memodelkan sistem berorientasi objek.

d. Mempermudah pengguna untuk membaca suatu sistem.

e. Menjelaskan informasi yang lebih detail dalam perancangan berupa *coding* suatu program.

Contoh Diagram UML yang Sering Digunakan

Use Case Diagram

Use Case Diagram adalah satu jenis dari diagram UML (*Unified Modelling Language*) yang menggambarkan hubungan interaksi antara sistem dan aktor. *Use Case* dapat mendeskripsikan tipe interaksi antara si pengguna sistem dengan sistemnya. *Use Case* merupakan sesuatu yang mudah dipelajari. Langkah awal untuk melakukan pemodelan perlu adanya suatu diagram yang mampu menjabarkan aksi aktor dengan aksi dalam sistem itu sendiri, seperti yang terdapat pada *Use Case*.

Activity Diagram

Activity diagram atau dalam bahasa Indonesia berarti diagram aktivitas, merupakan sebuah diagram yang dapat memodelkan berbagai proses yang tejadi pada sistem. Seperti layaknya runtutan proses berjalannya suatu sistem dan digambarkan secara vertikal. *Activity* diagram adalah salah satu contoh diagram dari UML dalam pengembangan dari *Use Case*.

Class Diagram

Class diagram atau diagram kelas merupakan suatu diagram yang digunakan untuk menampilkan kelas-kelas berupa pake-paket untuk memenuhi salah satu kebutuhan paket yang akan digunakan nantinya. Namun, pada Class diagram desain modelnya dibagi menjadi 2 bagian. Class diagram yang pertama merupakan penjabaran dari domain model yang merupakan abstraksi dari basis data. Class diagram yang kedua merupakan bagian dari modul program MVC pattern (Model View Controller), di mana terdapat class boundary sebagai class *interface*, *class control* sebagai tempat ditemukannya algoritma, dan *class entity* sebagai tabel dalam basis data dan *query* program.

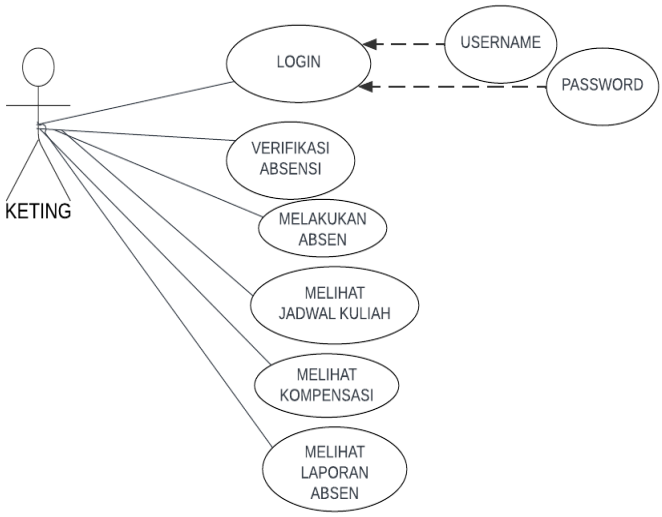
3. METODE PENELITIAN

Use case

Use case diagram digunakan untuk menjelaskan kegiatan apa saja yang dapat dilakukan oleh pengguna terhadap sistem yang sedang berjalan. Pada perancangan sistem informasi absensi mahasiswa ini, terdapat beberapa aktor yang terlibat dalam penggunaan sistem tersebut yaitu sebagai berikut:

Use Case Diagram Keting

Use case diagram Keting merupakan gambaran bagaimana bentuk sistem yang akan dibangun. Dengan adanya sistem usulan ini akan mempermudah dalam pembuatan sistem yang akan dibangun. Berikut *use case* diagram Keting yang terdapat pada Gambar 3.1



Gambar 3. *Use case keting.*

Deskripsi:

a. Keting login dengan menggunakan *ussernamе* dan *password*

Pada tahap ini, keting akan memasukan informasi login yang valid seperti *ussernamе*, *password*, dan level untuk mengakses ke sistem. Tujuannya untuk memverifikasi identitas keting untuk masuk ke sistem.

b. Keting verifikasi Dosen

Setelah berhasil login, keting dapat menggunakan fitur ini untuk memverifikasi kehadiran dosen untuk memastikan bahwa dosen tersebut benar-benar hadir dalam perkuliahan yang di jadwalkan.

c. Keting melihat informasi jadwal kuliah

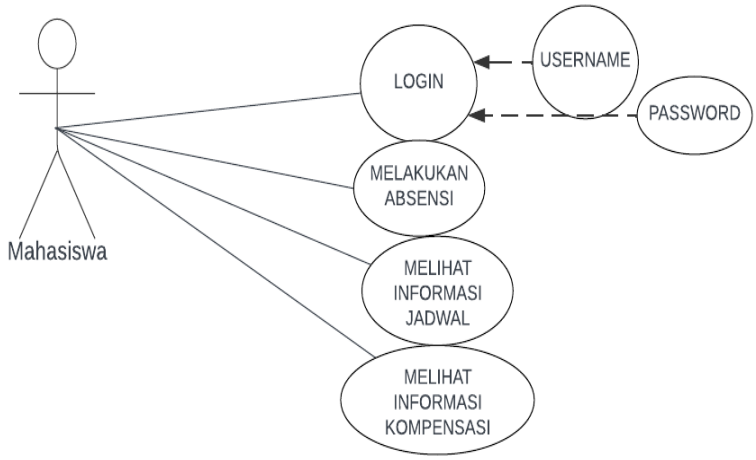
Keting dapat melihat jadwal kuliah yang terkait dengan mata kuliah, ruang, waktu, dan informasi penting lainya. Tujuannya untuk memberikan keting informasi yang jelas tentang jadwal kulih.

d. Keting dapat melihat kompensasi

e. Keting dapat melihat laporan absensi mahasiswa

f. Keting dapat melakukan absensi

Use case diagram Mahasiswa



Gambar 4. *use case* Mahasiswa.

Deskripsi :

a. Mahasiswa login menggunakan *ussernamе* dan *password*

Mahasiswa login ke sistem dengan menggunakan *ussernamе* dan *password*. Tujuannya adalah untuk memverifikasi identitas mahasiswa dan memberikan akses ke fitur-fitur yang relevan di dalam sistem.

b. Mahasiswa dapat melakukan absensi

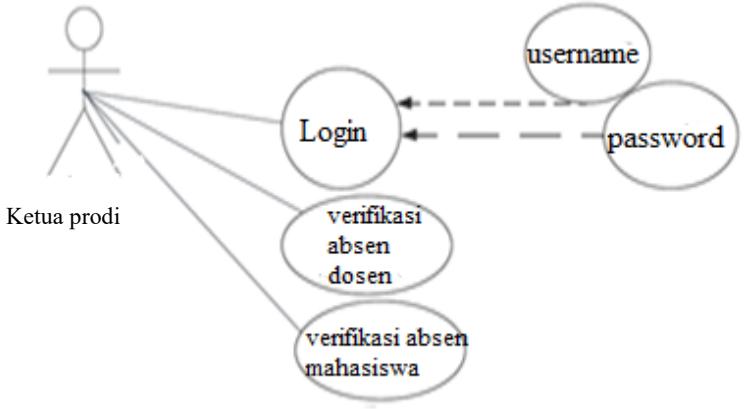
Mahasiswa dapat memilih mata kuliah yang sedang berlangsung sesuai jadwal untuk melakukan absensi. Tujuannya adalah untuk memastikan pencatatan kehadiran mahasiswa secara akurat dan membantu dalam pemantauan kehadiran.

c. Mahasiswa dapat melihat jadwal kuliah

Mahasiswa dapat melihat informasi seperti mata kuliah yang diambil, waktu dan tempat kuliah, dan dosen pengajar. Tujuannya untuk memberikan mahasiswa informasi yang jelas tentang jadwal kuliah mereka sehingga ketika melakukan absen dapat memilih mata kuliah yang sesuai.

d. Mahasiswa dapat melihat informasi kompensasi

Use case diagram Ketua prodi



Gambar 5. *use case* Ketua prodi.

Deskripsi:

a. Ketua prodi login

Ketua prodi login ke sistem dengan menggunakan menggunakan *ussernamе*, *password* dan level..

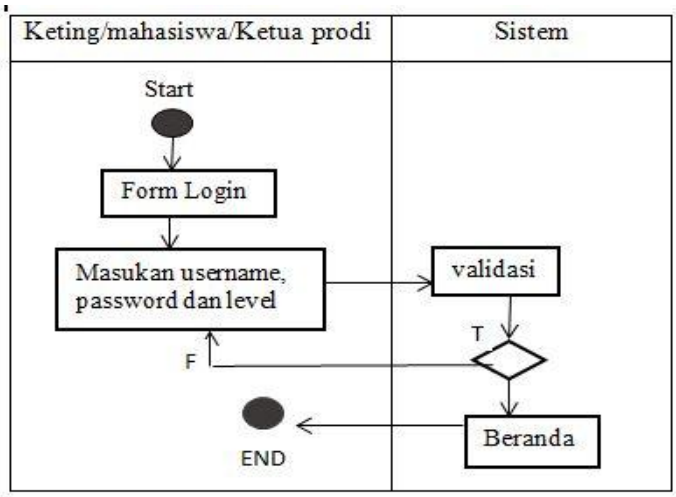
b. Ketua prodi melakukan verifikasi absensi mahasiswa dan Dosen

Verifikasi Absen Dosen: Ketua prodi dapat melihat daftar absensi dosen dan memverifikasi kebenaran dan keakuratan absensi yang dilakukan oleh dosen. Tujuannya untuk memastikan bahwa absensi dosen tercatat dengan benar dan memberikan informasi yang akurat terkait kehadiran dosen. Sedangkan Verifikasi Absen Mahasiswa: pada bagian ketua prodi dapat melihat daftar absensi mahasiswa dan memverifikasi kebenaran dan keakuratan absensi yang dilakukan oleh mahasiswa. Tujuannya adalah untuk memastikan bahwa absensi mahasiswa tercatat dengan benar dan memberikan informasi yang akurat terkait kehadiran mahasiswa dalam kegiatan perkuliahan.

Activity Diagram

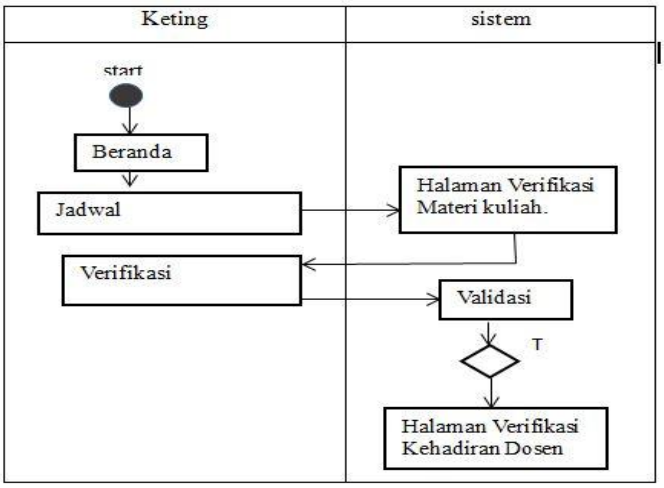
Activity diagram digunakan untuk menggambarkan kegiatan-kegiatan yang ada didalam suatu sistem. Agar dapat lebih memahami tentang sistem yang dibuat, maka perlu dibuatkan *activity* diagram tentang sistem yang akan diusulkan. Adapun *activity* diagram yang sedang diusulkan adalah sebagai berikut:

Activity Diagram Login



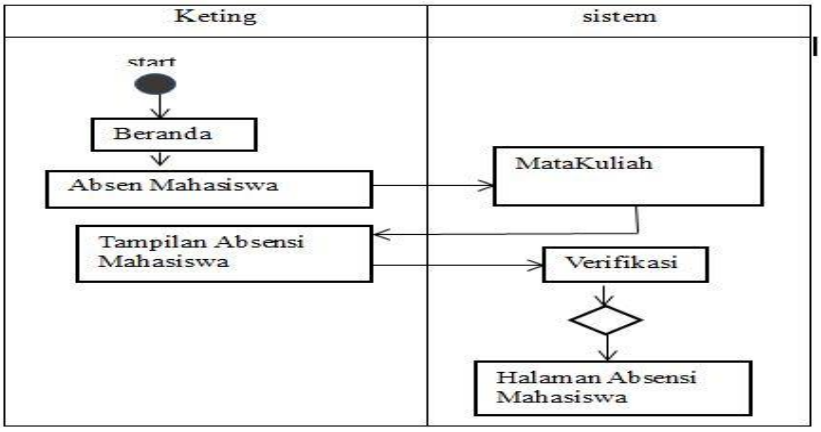
Gambar 6. *Activity* Diagram Login.

Activity Diagram verifikasi Materi kuliah oleh keting



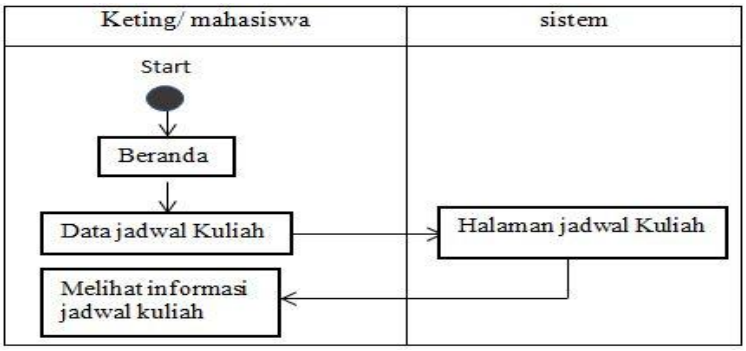
Gambar 7. *Activity* Diagram verifikasi dosen oleh keting.

Activity Diagram verifikasi mahasiswa oleh keting



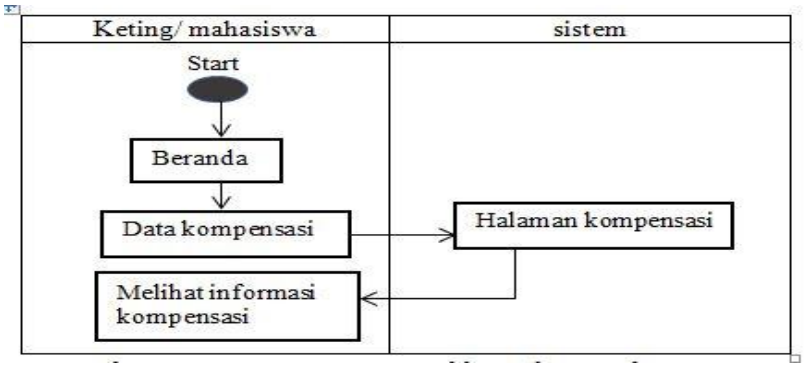
Gambar 8. *Activity* Diagram verifikasi mahasiswa oleh keting.

Activity Diagram melihat informasi jadwal kuliah



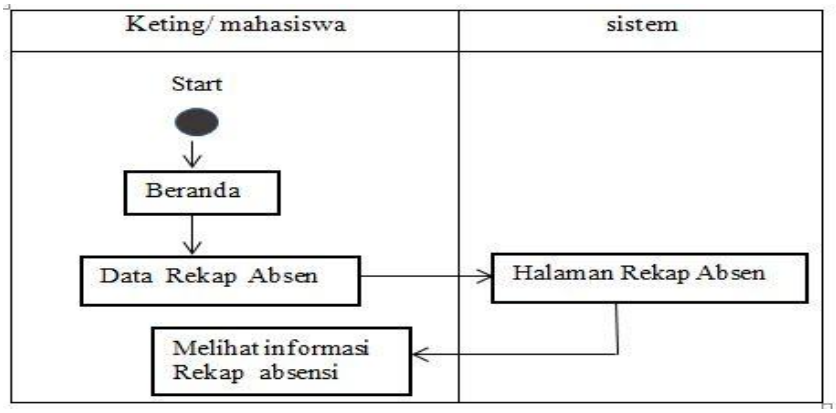
Gambar 9. *Activity* Diagram melihat informasi jadwal kuliah.

Activity Diagram melihat informasi kompensasi



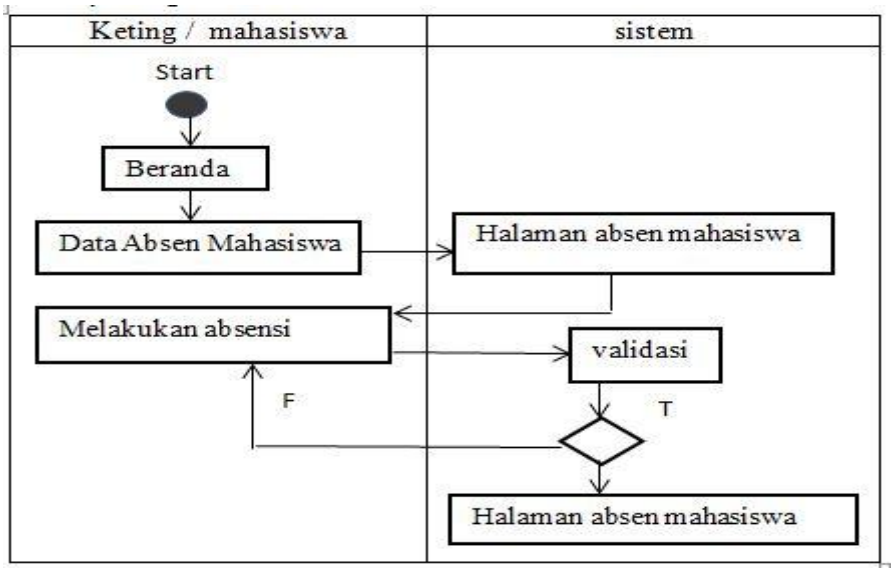
Gambar 10. *Activity* Diagram melihat informasi kompensasi.

Activity Diagram melihat laporan/Rekapan absensi



Gambar 11. *Activity* Diagram melihat laporan absensi.

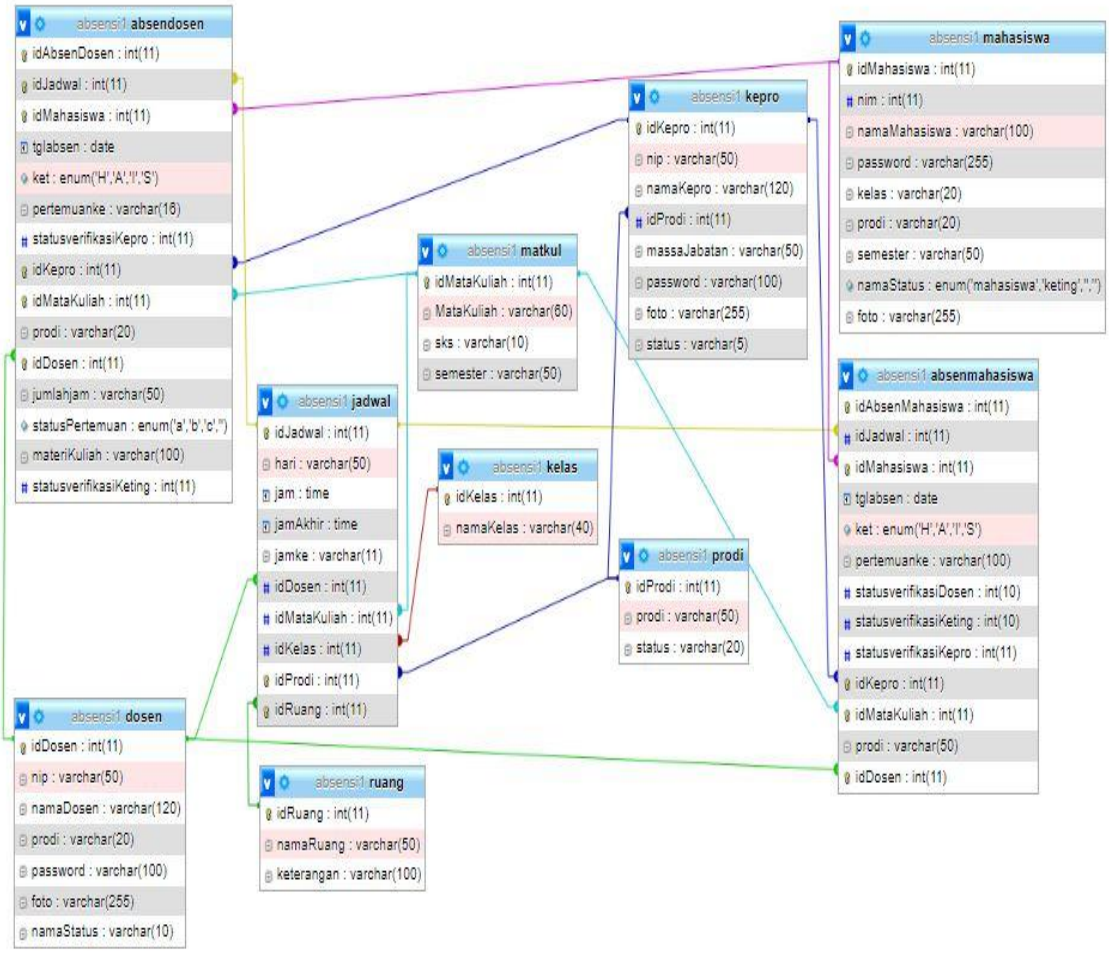
Activity Diagram melakukan absensi



Gambar 12. *Activity* Diagram melakukan absen.

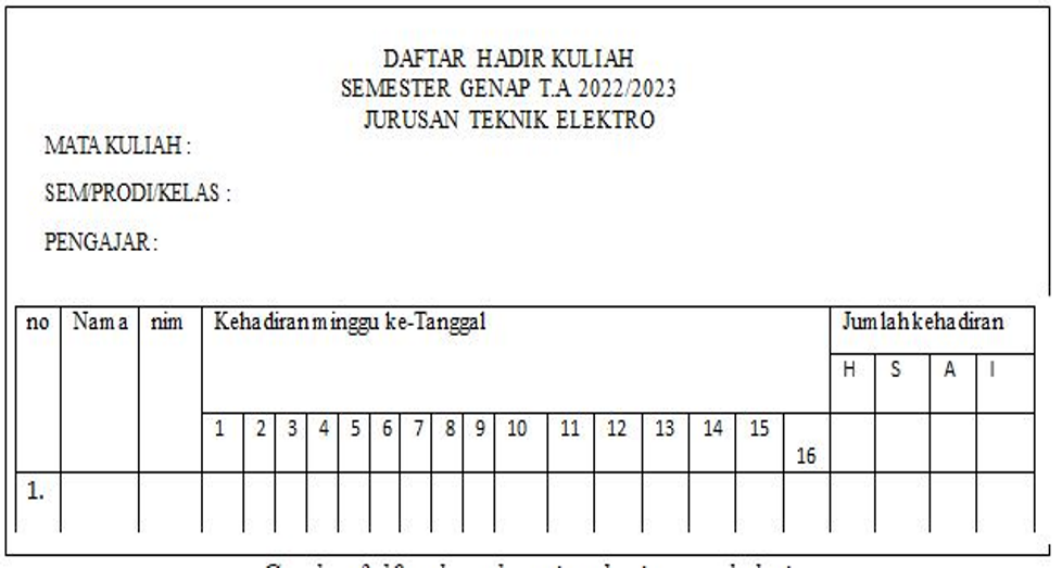
Desain

Class Diagram



Gambar 13. *Class Diagram.*

Antar Muka (Interface)



DAFTAR HADIR KULIAH
SEMESTER GENAP T.A 2022/2023
JURUSAN TEKNIK ELEKTRO

MATA KULIAH :

SEM/PRODI/KELAS :

PENGAJAR :

no	Nama	nim	Kehadiran minggu ke-Tanggal															Jumlah kehadiran			
																		H	S	A	I
			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16				
1.																					

Gambar 14. Rekap absensi mahasiswa pada keting.

Tabel 2. Rekap realisasi pengajaran dan verifikasi materi kuliah pada dosen.

No	Hari	Tanggal absen	Jam ke	Mata Kuliah	Jumlah jam	Nama Dosen	Status pertemuan	Materi kuliah	Status Verifikasi keting	Paraf Keting	Paraf Dosen
1.											
2											

4. HASIL DAN PEMBAHASAN

Implementasi Database

Pembuatan Database

Pertama yang kita lakukan adalah membuat *database* terlebih dahulu untuk menampung tabel yang kita buat. Cara membuat *database* adalah dengan mengklik *attribute 'new'* lalu isikan nama *database* sesuai dengan keinginan pada '*create database*' kemudian *create*.

Berikut adalah tabel-tabel yang ada dalam *database* adalah sebagai berikut.

Table	Action	Rows	Type	Collation	Size	Overhead
absendosen	Browse Structure Search Insert Empty Drop	34	InnoDB	utf8mb4_general_ci	16.0 KiB	-
absenmahasiswa	Browse Structure Search Insert Empty Drop	363	InnoDB	utf8mb4_general_ci	64.0 KiB	-
dosen	Browse Structure Search Insert Empty Drop	7	InnoDB	utf8mb4_general_ci	16.0 KiB	-
jadwal	Browse Structure Search Insert Empty Drop	12	InnoDB	utf8mb4_general_ci	64.0 KiB	-
kelas	Browse Structure Search Insert Empty Drop	6	InnoDB	utf8mb4_general_ci	32.0 KiB	-
kepro	Browse Structure Search Insert Empty Drop	4	InnoDB	utf8mb4_general_ci	32.0 KiB	-
kompensasi	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	16.0 KiB	-
mahasiswa	Browse Structure Search Insert Empty Drop	12	InnoDB	utf8mb4_general_ci	16.0 KiB	-
matkul	Browse Structure Search Insert Empty Drop	10	InnoDB	utf8mb4_general_ci	32.0 KiB	-
prodi	Browse Structure Search Insert Empty Drop	4	InnoDB	utf8mb4_general_ci	16.0 KiB	-
ruang	Browse Structure Search Insert Empty Drop	2	InnoDB	utf8mb4_general_ci	16.0 KiB	-

Gambar 15. Tabel *database.*

Implementasi Interface

Implementasi dari tampilan antar muka (*interface*) Sistem Informasi Manajemen Proyek ini adalah sebagai berikut :

Halaman Login Masuk untuk Keting, Mahasiswa dan Ketua Prodi

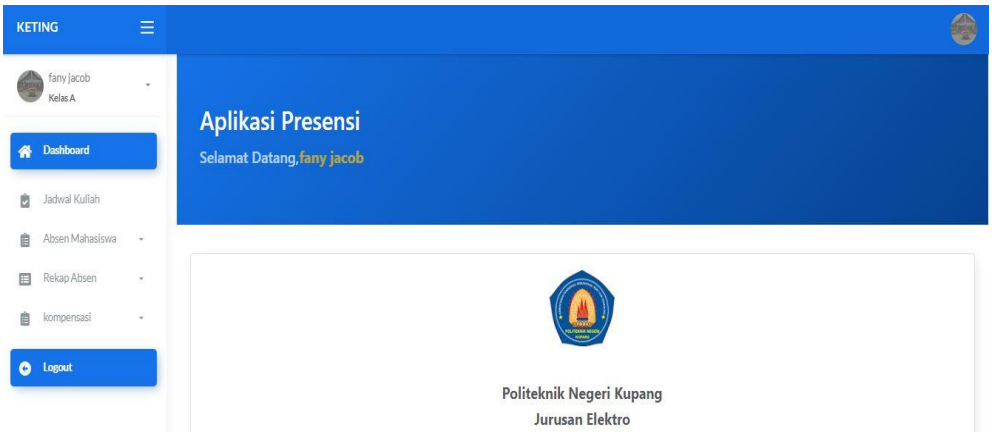
Pada halaman login : Keting, Mahasiswa, kepro akan memasukan username, password dan level untuk login kedalam sistem. Halaman ini membatasi untuk masuk dan mempunyai hak akses ke dalam sistem informasi agar keamanan dan yang berhak mengakses terjamin

Gambar 16. Halaman login.

Halaman Keting

Dashboard

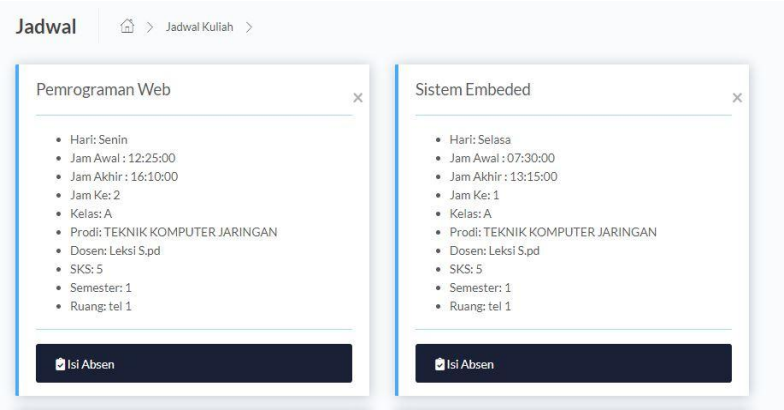
Ketika keting login akan menampilkan halaman dashboard seperti berikut :



Gambar 17. Halaman *dashboard.*

Jadwal Kuliah

Halaman jadwal kuliah memuat informasi jadwal kuliah sesuai kelas dan semester mahasiswa tersebut.



Gambar 18. Halaman jadwal.

Verifikasi Absen

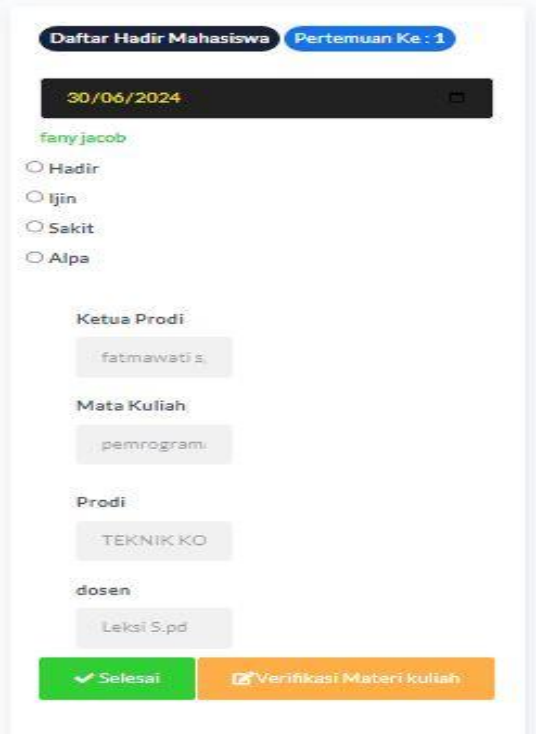
Pada halaman verifikasi absen, setelah dosen melakukan input data bagian realisasi pengajaran maka keting akan melakukan verifikasi materi kulih apakah sesuai atau tidak. Berikut adalah tabel verifikasi absen :

#	Hari	Tanggal Absen	Jam ke	Mata Kuliah	Jumlah Jam	Nama Dosen	Status Pertemuan	Materi Kuliah	Status Verifikasi Keting
1	Senin	2024-07-08	2	Pemrograman Web	2	Leksi S.pd	a	dddd	Verifikasi

Gambar 19. Halaman *verifiksi absen.*

Isi Absen

Halaman isi absen berada di halaman jadwal, jadi ketika keting melakukan absen sesuai jadwal dengan mengklik tombol isi absen maka akan muncul halaman untuk melakukan absen seperti gambar di bawah ini :



Gambar 20. Halaman isi absen.

Absen Mahasiswa

Pada halaman Absen Mahasiswa, keting akan melakukan verifikasi setiap mahasiswa yang sudah melakukan absen untuk memastikan bahwa mahasiswa tersebut benar-benar ada dalam kelas dan mengikuti pembelajaran sesuai jadwal.

No	Hari	Nama Mahasiswa	Tanggal Absen	Keterangan	Pertemuan Ke	Opsi	Verifikasi	Status Verifikasi Dosen	Status Verifikasi Kepro
1	Senin	nina adonis	2024-07-08	A	1	Edit	Verifikasi	Belum Diverifikasi	Belum Diverifikasi

Gambar 21. Halaman absen mahasiswa.

Rekap Absen

a. Rekap absen

Pada halaman Rekap absen hanya merekap setiap mata kuliah selama 16 pertemuan.

ABSENSI MAHASISWA
JURUSAN ELEKTRO

- Nama Mata Kuliah : PROJECT 1
- Nama Dosen : LEKSI S.PD
- Semester : 1
- Kelas : A

NO	NAMA Mahasiswa	NIM	PERTEMUAN KE:									JUMLAH			
			1	2	3	4	5	6	7	8	9	S	I	A	H
1	nina adonis	2023736105	A	A	H	S	I	H	H	H	H	2	4	15	14
2	fany jacob	2023736106	H	H	H	I	S	I	A	S	H	4	3	5	6
3	fany tlaan	2023736107	H	A	H	I	A	I	I	I	S	2	8	15	2
4	laura riwu	2023736108	H	H	H	I	S	A	I	I	I	1	5	5	3
5	Tae Missa	2023736109	H	A	H	I	I	I	H	H	H	0	3	5	7
6	andrew anin	2023736110	H	H	H	S	A	H	H	H	H	1	1	5	7
7	Rival nissi	2023736111	I	H	H	I	S	H	H	H	H	1	2	0	7
8	novita banmeta	2023736112	H	H	H	I	I	S	I	I	S	2	5	0	3
9	Leonard Kantur	2023736113	H	I	H	I	H	H	H	I	H	0	4	0	6
10	Jarvis	2023736114	H	H	I	I	H	A	H	H	H	0	3	5	7
11	Mario Kleden	2023736115	H	H	H	I	S	H	I	H	H	1	2	0	7

Mengetahui
Ketua Program Studi

FATMAWATI S.PD.
NIP. 197901012008012061

Kupang, 30 June 2024
Pengajar Mata Kuliah

LEKSI S.PD
NIP.197901012008012011

Gambar 22. Halaman rekap absen.

b. Realisasi pengajaran dan verifikasi materi kuliah

Pada halaman ini hanya merekap setiap mata kuliah selama 16 pertemuan yang di input oleh dosen.

REALISASI PENGAJARAN DAN
VERIFIKASI MATERI KULIAH

- Semester : 1
- Kelas : A

#	Hari	Tanggal Absen	Jam ke	Mata Kuliah	Jumlah Jam	Nama Dosen	Status Pertemuan	Materi Kuliah	Status Verifikasi Keting	Paraf Ketua Kelas	Paraf Dosen

Direkap : Admin
Tanggal : 09 July 2024

Diverifikasi dan Disetujui :
ka.Prodi

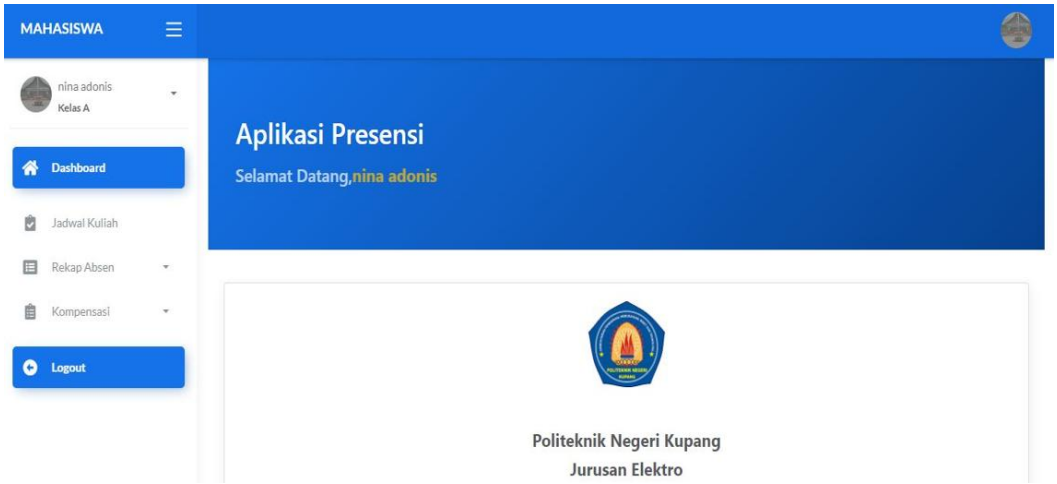
Gambar 23. Halaman verifiksi materi.

Halaman Mahasiswa

Halaman mahasiswa pada bagian jadwal kuliah, isi absen, rekap absen dan kompensasi memiliki tampilan yang sama dengan halaman pada keting, yang membedakan ke dua halaman tersebut adalah bagian verifikasi mahasiswa dan verifikasi materi kuliah hanya terdapat di halaman keting sedangkan halaman mahasiswa tidak ada.

Dashboard

Halaman dashboard untuk mahasiswa sebagai berikut :



Gambar 24. Halaman *dashbord* untuk mahasiswa.

Pengujian Sistem

Pengujian sistem adalah proses yang dilakukan untuk mengevaluasi kinerja suatu sistem atau perangkat lunak guna memastikan bahwa itu berfungsi sebagaimana yang diharapkan. Tujuan utama dari pengujian sistem adalah untuk mengidentifikasi masalah atau ketidaksesuaian dalam sistem dan memastikan bahwa sistem tersebut dapat memenuhi persyaratan dan ekspektasi yang telah ditetapkan sebelumnya.

Pengujian Kompatibilitas Sistem

Pengujian kompatibilitas sistem adalah proses yang dilakukan untuk memastikan bahwa perangkat lunak atau perangkat keras berfungsi dengan baik dan berinteraksi dengan sistem lain atau komponen yang relevan. Pengujian ini penting untuk memastikan bahwa sistem atau aplikasi dapat beroperasi dengan lancar di berbagai konfigurasi perangkat keras, sistem operasi, perangkat lunak lain, dan lingkungan yang berbeda.

Tabel 3. Pengujian Kompabilitas Sistem.

No	Web Browser	Kompitibel
1	Microsoft Edge	V
2	Google Chrome	V
3	Opera	V

Pengujian Fungsi Sistem Berdasarkan Alur Logika Sistem Secara Umur

Pengujian fungsi sistem berdasarkan alur logika sistem secara umur adalah suatu metode pengujian yang digunakan untuk memverifikasi apakah sistem beroperasi sesuai dengan alur logika atau aliran kerja yang telah dirancang untuk sistem tersebut selama berbagai fase umur atau tahap perkembangannya. Metode ini membantu memastikan bahwa sistem berperilaku

sesuai dengan harapan selama seluruh siklus hidupnya, dari awal pengembangan hingga pemeliharaan dan pembaruan.

5. KESIMPULAN DAN SARAN

Kesimpulan

Berdasarkan hasil Sistem Informasi Absensi Mahasiswa Berbasis web di Politeknik Negeri Kupang, dapat disimpulkan bahwa: Sistem ini berbasis website, sehingga memudahkan dalam melakukan absensi. Memudahkan admin dalam rekap absensi. Memudahkan mahasiswa untuk melihat kompensasi. Memudahkan mahasiswa untuk melihat kehadian mereka dalam setiap mata kuliah. Mempermudah dosen dalam melakukan rekap absen. Mempermudah admin dalam menghitung kompensasi.

Saran

Berdasarkan kesimpulan di atas, berikut adalah beberapa pengembangan dan penggunaan sistem informasi di masa mendatang saran untuk: Disarankan untuk sistem informasi absensi mahasiswa ini di gunakan pada semua jurusan di Politeknik Negeri Kupang. Mengembangkan sistem ini menjadi aplikasi mobile sehingga lebih mudah diakses dan digunakan oleh mahasiswa dan dosen. Lakukan pemeriksaan dan pemeliharaan secara berkala untuk menjaga performa sistem tetap stabil, dan implementasikan perbaikan yang disarankan guna meningkatkan performa lebih lanjut.

Dengan kesimpulan dan saran ini, diharapkan Sistem Informasi Absensi Mahasiswa Berbasis di Politeknik Negeri Kupang dapat terus berkembang dan memberikan manfaat maksimal bagi semua pihak yang terlibat.

DAFTAR REFERENSI

Bangun, R., dkk. (2014). Sistem informasi monitoring TV berlangganan di TV regional Bali. *Jurnal Sistem Informasi, 3*(2), 72–77.

Dennis, A., Wixom, B. H., & Tegarden, D. (2015). *Systems analysis and design: An object-oriented approach with UML* (6th ed.). Wiley.

Fitriani, L., & Hidayat, T. (2021). Penerapan framework Laravel pada pengembangan sistem informasi akademik berbasis web. *Jurnal Media Informatika Budidarma, 5*(4), 1377–1385. https://doi.org/10.30865/mib.v5i4.3268

Handayani, P. W., Hidayanto, A. N., Budi, I., & Sandhyaduhita, P. I. (2017). Critical success factors for information system implementation: A systematic literature review. *Heliyon, 3*(9), e00359. https://doi.org/10.1016/j.heliyon.2017.e00359

Hendini, A. (2016). Pemodelan UML sistem informasi monitoring penjualan dan stok barang. *Jurnal Khatulistiwa Informatika, 4*(2), 107–116. https://doi.org/10.31294/jki.v4i2.1262

Lengkong, O. H., Fiden, D. H., & Masrikat, A. (2016). Sistem informasi absensi real-time di Universitas Klabat. *CogITo Smart Journal.*

Mell, P., & Grance, T. (2011). *The NIST definition of cloud computing* (NIST Special Publication 800-145). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-145

Mulia, A. G. (2020). Sistem informasi absensi berbasis web di Politeknik Negeri Padang. *Jurnal Teknologi Informasi Indonesia (JTII).*

Mulyani, S. (2016). *Metode analisis dan perancangan sistem*. Abdi Sistematika.

Prasetyo, A., & Rahayu, S. (2023). Implementasi Unified Modeling Language (UML) dalam perancangan sistem informasi berbasis web. *Jurnal Sisfokom (Sistem Informasi dan Komputer), 12*(2), 252–260. https://doi.org/10.32736/sisfokom.v12i2.1725

Pressman, R. S., & Maxim, B. R. (2024). *Software engineering: A practitioner's approach* (9th ed.). McGraw-Hill Education.

Ramadhan, W., & Putra, S. H. (2022). Aplikasi absensi mahasiswa dan dosen Politeknik Ganesha Medan berbasis web menggunakan PHP dan MySQL. *REMIK: Riset dan E-Jurnal Manajemen Informatika Komputer.*

Robi Yanto. (2019). *Manajemen basis data menggunakan MySQL*. Deepublish.

Rosa, A. S., & Shalahuddin, M. (2018). *Rekayasa perangkat lunak terstruktur dan berorientasi objek*. Informatika.

Sallaby, A. F., & Kanedi, I. (2020). Perancangan sistem informasi jadwal dokter menggunakan framework CodeIgniter. *Jurnal Media Infotama, 16*(1), 48–53. https://doi.org/10.37676/jmi.v16i1.1121

Setiawan, D., & Azizah, N. (2022). Pengembangan sistem informasi akademik berbasis web menggunakan metode Agile. *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi), 6*(5), 884–891. https://doi.org/10.29207/resti.v6i5.4384

Sitio, A. S., & Laia, E. (2014). Perancangan dan pembuatan sistem informasi absensi mahasiswa berbasis web pada STMIK Pelita Nusantara Medan. *Jurnal Mantik Penusa.*

Sommerville, I. (2015). *Software engineering* (10th ed.). Pearson.

Wandri, R., Fadhilla, M., Setiawan, P. R., Daulay, S., Hanafiah, A., & Qurniawan, D. F. (2025). Optimalisasi pengelolaan data sekolah melalui sistem informasi sekolah berbasis web dengan pendekatan Agile. *SAINSTEK, 13*(1), 157–165. https://doi.org/10.35583/js.v13i1.327

Zhang, Y., & Zhang, X. (2021). Web-based information systems development: Trends and challenges. *Journal of Software Engineering and Applications, 14*(8), 345–358. https://doi.org/10.4236/jsea.2021.148021