

BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1 Tinjauan Pustaka

Beberapa penelitian yang telah dilakukan sebelumnya yang relevan dengan Sistem Informasi Pengajuan Cuti ini yaitu salah satunya penelitian yang dilakukan oleh Wibowo [4]. Sistem ini bertujuan untuk memberikan solusi pada pelayanan administrasi akademik di masa pandemi sehingga antrian dan kerumunan mahasiswa yang melakukan pengajuan layanan tidak terjadi lagi. Sistem ini berbasis website dengan metode pengembangan sistem *Rapid Application Development (RAD)* melalui tiga tahapan yaitu *Requirements Planning*, *RAD Design Workshop* dan *Implementation*. Manfaat dari sistem ini yaitu mahasiswa dapat melakukan pengajuan layanan akademik dari rumah menggunakan sistem sehingga lebih efektif dan efisien.

Penelitian tentang cuti akademik juga pernah dilakukan oleh Nurhayati [5]. Sistem ini bertujuan agar pelayanan akademik mahasiswa yang banyak jenisnya dapat lebih efisien dari segi waktu. Pembangunan sistem ini memanfaatkan teknologi web sebagai layanan informasi dan fitur tambahan *sms gateway* sebagai alat pendistribusian pesan yang berisi informasi kegiatan akademik dan administrasi mahasiswa yang dapat diakses kapan pun dan dimana pun pengguna berada atau pengguna perlukan. Tahapan-tahapan penelitian yang digunakan untuk menyelesaikan penelitian ini dimulai dari perumusan masalah, pengumpulan data, analisa sistem, perancangan sistem, pengujian sistem, menerapkan sistem, pemeliharaan sistem dan yang terakhir yaitu membuat laporan. Manfaat dari sistem ini yaitu mempermudah mahasiswa dalam mendapatkan pelayanan administrasi dan mempermudah biro administrasi akademik dan kemahasiswaan dalam memberikan pelayanan.

Penelitian serupa juga pernah dilakukan oleh Mardiansyah [6]. Penelitian ini bertujuan agar pengisian *form* pengajuan permohonan dapat dilakukan melalui komputer sepenuhnya. Perancangan sistem ini menggunakan desain model dalam bentuk *use case diagram*, *entity relation diagram*, *class diagram*, *sequence diagram* dan *activity diagram*. Tahapan-tahapan penelitian yang digunakan digambarkan dalam diagram alir penelitian. Tahapan penelitian tersebut yaitu dimulai dari studi literatur, pengumpulan data, analisa sistem, desain, implementasi,

pengujian, dan yang terakhir adalah pembuatan dokumentasi laporan. Manfaat dari sistem ini yaitu membantu mahasiswa untuk melakukan pengisian surat permohonan.

Penelitian yang akan dilakukan mempunyai perbedaan dengan penelitian sebelumnya. Perbedaan tersebut diantaranya proses persetujuan dilakukan di dalam sistem termasuk tanda tangan. Sistem ini memiliki hak akses sehingga tidak semua orang bisa mengaksesnya. Sistem ini juga memiliki fasilitas *upload file* lampiran sebagai pertimbangan. Tahapan pengembangan sistem ini menggunakan metode *waterfall* yang dibangun menggunakan bahasa pemrograman PHP dan *MySQL*.

2.2 Landasan Teori

2.2.1 Sistem Informasi

Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau menyelesaikan suatu sasaran tertentu. Informasi memiliki arti data yang diolah menjadi bentuk yang lebih bermanfaat bagi penggunaannya [7]. Jadi, sistem informasi merupakan seperangkat entitas yang terdiri dari *software*, *hardware*, dan *brainware* yang saling bergotong-rotong untuk menyajikan data yang nantinya diolah sehingga dapat bermanfaat bagi penggunaannya [8].

2.2.1.1 Karakteristik Sistem

Suatu sistem mempunyai karakteristik yang menunjukkan bahwa hal tersebut bisa dikatakan suatu sistem. Sistem memiliki delapan karakteristik. Karakteristik tersebut yaitu komponen sistem, batasan sistem, lingkungan luar sistem, penghubung sistem, masukan sistem, keluaran sistem, pengolah sistem, dan sasaran sistem [9].

1. **Komponen Sistem (*Components*)**
Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, artinya saling bekerja membentuk satu kesatuan.
2. **Batasan Sistem (*Boundary*)**
Batasan sistem merupakan garis abstraksi yang memisahkan anatara sistem dan lingkungannya.
3. **Lingkungan Luar Sistem (*Environment*)**
Lingkungan luar sistem adalah segala hal yang berada di luar ruang lingkup atau batasan sistem yang mempengaruhi operasi dari sistem tersebut.

4. Penghubung Sistem (*Interface*)
Penghubung sistem merupakan media yang menghubungkan sistem dengan subsistem lain yang memungkinkan sumber daya mengalir dari satu subsistem ke subsistem yang lain.
5. Masukan Sistem (*Input*)
Masukan sistem yaitu energi yang dimasukkan ke dalam sistem bisa berupa sinyal dan perawatan.
6. Keluaran Sistem (*Output*)
Keluaran sistem yaitu hasil energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dimana keluaran yang dihasilkan dapat menjadi masukan bagi subsistem yang lain.
7. Pengolah Sistem (*Process*)
Pengolah sistem merupakan proses yang akan mengubah *input* menjadi *output*.
8. Sasaran Sistem (*Objective*)
Suatu sistem harus memiliki sasaran yang pasti karena sistem dikatakan berhasil apabila mengenai sasaran yang sudah direncanakan [10].

2.2.1.2 Kualitas Informasi

Kualitas informasi merupakan kualitas keluaran dalam bentuk informasi yang dihasilkan oleh sistem informasi yang digunakan [11]. Kualitas informasi yang baik akan memberikan kepuasan kepada pengguna sistem informasi. Menurut Tata Sutabri kualitas informasi bergantung pada 3 hal yaitu:

1. Akurat (*Accurate*)
Akurat berarti informasi harus benar sehingga tidak menyesatkan dan tidak ada kesalahan. Informasi yang akurat diartikan juga informasi harus jelas maksudnya. Informasi harus akurat karena ada kemungkinan terjadi gangguan yang menyebabkan berubah atau rusaknya informasi tersebut dari sumber informasi sampai penerima informasi.
2. Tepat Waktu (*Timelines*)
Informasi harus sampai kepada penerima dengan tepat waktu, tidak boleh terlambat. Informasi yang terlambat akan berkurang nilainya karena informasi merupakan landasan dalam mengambil suatu keputusan.
3. Relevan (*Relevance*)

Relevan dalam kualitas informasi diartikan dengan informasi mempunyai manfaat untuk penggunanya, dimana relevansi setiap individu bergantung pada yang menerima dan membutuhkan [12].

2.2.2 Website

Website memiliki arti kumpulan dari halaman-halaman situs yang biasanya terangkum dalam sebuah domain atau subdomain yang bisa diakses dengan jaringan internet yang berisi berbagai informasi. Domain merupakan nama unik untuk menggantikan alamat IP agar lebih mudah diingat. Informasi yang ditampilkan dalam suatu website bisa berupa teks, gambar, animasi, suara ataupun video [9].

2.2.3 Cuti Akademik

Cuti menurut KBBI diartikan dengan berhenti bekerja atau tidak melakukan pekerjaan seperti biasanya dalam beberapa waktu untuk beristirahat ataupun hal lain [13]. Cuti akademik menurut peraturan akademik Politeknik Negeri Cilacap tahun 2016 adalah pemberhentian sementara dari proses belajar mengajar karena sakit, melahirkan, dan hal-hal lain yang dianggap perlu yang ditetapkan melalui rapat program studi dan jurusan yang dapat diajukan oleh mahasiswa ataupun program studi.

2.2.4 Database Management System (DBMS)

Database Management System (DBMS) merupakan software yang digunakan untuk mengendalikan pembuatan, pemeliharaan, pengolahan, dan penggunaan data yang berskala besar. *DBMS* adalah hal penting yang digunakan dalam berbagai aspek seperti media sosial menggunakan *DBMS* untuk menyimpan data pengguna media sosial tersebut ke dalam *DBMS MySQL*. Contoh *DBMS* yang sering digunakan yaitu *MariaDB* dan *MySQL* [14]. Bahasa yang digunakan *Database Management System* yaitu:

1. *Data Definition Language (DDL)*

DDL menyediakan perintah yang berhubungan dengan operasi-operasi dasar seperti mendefinisikan, memodifikasi dan menghapus skema relasi misalnya tabel, view, prosedur dan fungsi. Perintah dalam DDL adalah *Create*, *Alter* dan *Drop*.

a. *Create*

Perintah *create* digunakan untuk membuat objek baru seperti *database* dan tabel. Contoh *create* database: “*CREATE DATABASE sistem_cuti;*” dan contoh *create*

tabel: “*CREATE TABLE* tb_verifikasi (id_verifikasi int *PRIMARY KEY*, id_pengajuan *INT*, nip_npak *CHAR* (20), tgl_verif *DATE*, status *ENUM* ('Diterima', 'Ditolak'))”.

b. *Alter*

Perintah *alter* digunakan untuk mengubah suatu struktur pada tabel yang telah dibuat. Contoh perintah: “*ALTER TABLE* verifikasi *MODIFY* status *VARCHAR* (8);”.

c. *Drop*

Perintah *drop* digunakan untuk menghapus tabel ataupun *database*. Contoh perintah *drop* tabel: “*DROP TABLE* tb_verifikasi;”. Contoh perintah *drop database*: “*DROP DATABASE* sistem_cuti;”.

2. *Data Manipulation Language (DML)*

DML menyediakan perintah untuk menambah, memodifikasi, menghapus dan mengambil data dalam basis data. Perintah dalam DML adalah *Insert*, *Delete*, *Update* dan *Select*.

a. *Insert*

Perintah *insert* digunakan untuk memasukkan data ke dalam *database*. Contoh perintah: “*INSERT INTO* tb_verifikasi *VALUES* (1, 1, '1801020323', '2022-03-31', 'Diterima');”.

b. *Delete*

Perintah *delete* digunakan untuk menghapus data. Contoh perintah: “*DELETE FROM* tb_verifikasi *WHERE* id_verifikasi = '1';”.

c. *Update*

Perintah *update* digunakan untuk memperbaiki data. Contoh perintah: “*UPDATE* tb_verifikasi *SET* status = "Ditolak" *WHERE* id_pengajuan = '1';”.

d. *Select*

Perintah *select* digunakan untuk menampilkan data. Contoh perintah: “*SELECT * FROM* tb_verifikasi;”.

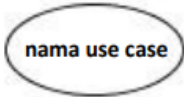

2.2.5 *Unified Modeling Language (UML)*



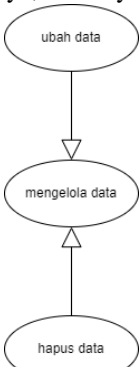
UML merupakan bahasa visual untuk pemodelan dan komunikasi tentang sebuah sistem dengan menggunakan diagram dan teks-teks pendukung. UML hanya berfungsi untuk melakukan pemodelan. Jadi, penggunaan UML tidak terbatas pada metodologi tertentu.


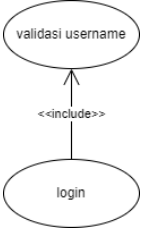
1. *Use Case Diagram*

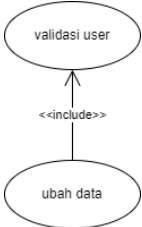
Use case diagram mendeskripsikan interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibangun. *Use case diagram* digunakan untuk mengetahui fungsi apa saja yang ada pada sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Simbol-simbol dari *use case diagram* adalah sebagai berikut:

Tabel 2. 1 Simbol *Use Case Diagram* [15]

| No | Nama | Simbol | Deskripsi |
|----|-------------------------|---|---|
| 1. | <i>Use Case</i> |  | Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; Biasanya dinyatakan dengan menggunakan kata kerja di awal di awal frase nama <i>use case</i> . |
| 2. | Aktor / <i>Actor</i> |  | Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama aktor. |

| No | Nama | Simbol | Deskripsi |
|----|--------------------------------------|---|--|
| 3. | <i>Asosiasi / Association</i> |  | Komunikasi antara aktor dan use case yang berpartisipasi pada use case atau use case memiliki interaksi dengan actor. |
| 4. | <i>Generalisasi / generalization</i> |  | <p>Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah use case dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya, misalnya:</p>  <p>Arah panah mengarah pada <i>use case</i> yang menjadi generalisasinya (umum).</p> |

| No | Nama | Simbol | Deskripsi |
|----|---------------------------------------|---|---|
| 5. | Menggunakan/ <i>Include / uses</i> |  | <p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> di mana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini. Ada dua sudut pandang yang cukup besar mengenai <i>include</i> di <i>use case</i>:</p> <ol style="list-style-type: none"> 1. <i>Include</i> berarti <i>use case</i> yang ditambahkan akan selalu dipanggil saat <i>use case</i> tambahan dijalankan, misal:  <pre> graph TD login((login)) -- <<include>> --> validasi_username((validasi username)) </pre> |

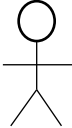



| No | Nama | Simbol | Deskripsi |
|----|------|--------|---|
| | | | <p>2. Include berarti <i>use case</i> tambahan akan selalu melakukan pengecekan apakah <i>use case</i> yang ditambahkan telah dijalankan sebelum <i>use case</i> tambahan dijalankan, misal:</p>  <pre> graph BT UC1([ubah data]) -- "<<include>>" --> UC2([validasi user]) </pre> <p>Kedua interpretasi di atas dapat dianut salah satu atau keduanya tergantung pada pertimbangan dan interpretasi yang dibutuhkan.</p> |


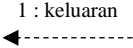
2. *Sequence Diagram*

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Oleh karena itu, untuk

menggambar diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu.

Tabel 2. 2 Simbol *Sequence Diagram* [15]

| No | Nama | Simbol | Deskripsi |
|----|------------------------------|---|---|
| 1. | <i>Actor</i> |  | Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama actor. |
| 2. | <i>Lifeline</i> /garis hidup |  | Menyatakan kehidupan suatu objek. |
| 3. | Objek |  | Menyatakan objek yang berinteraksi dengan pesan. |
| 4. | Pesan tipe send |  | Menyatakan bahwa suatu |

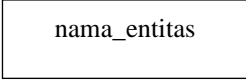
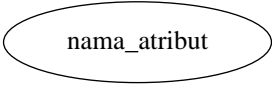
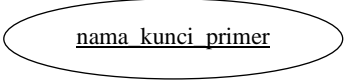
| No | Nama | Simbol | Deskripsi |
|----|-------------------|---|---|
| | |  | objek mengirimkan data/masukan /informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim. |
| 5. | Pesan tipe return |  | Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian |
| 6. | | | |


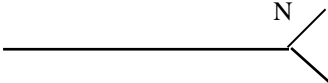
2.2.6 Entity Relationship Diagram (ERD)

Pemodelan awal basis data yang paling banyak digunakan adalah menggunakan ERD (*Entity relationship diagram*). ERD digunakan untuk pemodelan basis data relasional. ERD memiliki beberapa aliran notasi seperti notasi Chen (dikembangkan oleh Peter Chen), Barker (dikembangkan oleh Richard Barker) dan beberapa notasi lain. Tabel 2.5

di bawah ini adalah simbol-simbol yang digunakan pada ERD dengan notasi Chen.

Tabel 2. 3 Simbol *entity relationship diagram* [15]


| No | Simbol | Deskripsi |
|----|--|---|
| 1. | Entitas/ <i>entity</i>  | Entitas merupakan data inti yang akan disimpan; bakal table pada basis data; benda yang memiliki data dan harus disimpan datanya agar dapat diakses oleh aplikasi komputer; penamaan entitas biasanya lebih ke kata benda dan belum merupakan nama tabel. |
| 2. | Atribut  | <i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas. |
| 3. | Atribut kunci primer  | <i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses <i>record</i> yang diinginkan; biasanya berupa id; kunci primer dapat lebih dari satu kolom; asalkan kombinasi dari beberapa kolom tersebut dapat bersifat unik (berbeda tanpa ada yang sama). |

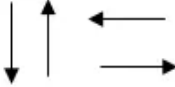

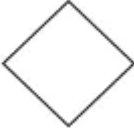



| No | Simbol | Deskripsi |
|----|---|---|
| 4. |  | Relasi yang menghubungkan antar entitas; biasanya diawali dengan kata kerja. |
| 5. |  | Penghubung antara relasi dan entitas dimana di kedua ujungnya memiliki multiplicity kemungkinan jumlah pemakaian. Kemungkinan jumlah maksimum keterhubungan antara entitas satu dengan entitas yang lain disebut dengan kardinalitas. |

2.2.7 Flowchart

Flowchart atau bagan alir merupakan bagan yang menunjukkan alir atau langkah-langkah dan keputusan di dalam suatu program atau prosedur. *Flowchart* digunakan untuk menggambarkan urutan proses program supaya lebih mudah dalam penyampaian alur jalannya program ke orang lain [15].

Tabel 2. 4 Simbol *Entity relationship diagram* [16]

| No | Nama | Simbol | Deskripsi |
|----|-----------------------|---|---|
| 1. | <i>Terminal point</i> |  | Simbol yang menunjukkan mulai atau berakhirnya suatu program. |

| No | Nama | Simbol | Deskripsi |
|----|-------------------------|---|---|
| 2. | <i>Flow direction</i> |  | Simbol yang digunakan untuk menyatakan jalannya arus suatu proses |
| 3. | Proses |  | Simbol yang menunjukkan sebuah kegiatan yang dilakukan oleh komputer atau sistem. |
| 4. | <i>Decision</i> |  | Simbol yang menunjukkan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban yaitu iya atau tidak. |
| 5. | <i>Input output</i> |  | Simbol ini menyatakan proses <i>input</i> atau <i>output</i> tanpa tergantung pada jenis peralatannya. |
| 6. | <i>Manual operation</i> |  | Simbol yang menyatakan suatu proses/tindakan yang dilakukan tanpa menggunakan komputer. |
| 7. | <i>Document</i> |  | Simbol yang menunjukkan <i>input</i> berupa dokumen dalam bentuk kertas atau <i>output</i> yang harus dicetak. |