



**TUGAS AKHIR**

**PENGEMBANGAN SISTEM INFORMASI  
BADAN EKSEKUTIF MAHASISWA  
POLITEKNIK NEGERI CILACAP  
*INFORMATION SYSTEM DEVELOPMENT OF  
STUDENT EXECUTIVE AGENCY OF  
STATE OF POLYTECHNIC CILACAP***

**Oleh**

**SANDY PERMANA PUTRA  
NPM 18.02.02.064**

**DOSEN PEMBIMBING :  
ABDUL ROHMAN S., S.T., M.KOM.  
NIDN. 0615058407**

**M. NUR FAIZ, S.KOM., M.KOM.  
NIDN. 0624039301**

**JURUSAN TEKNIK INFORMATIKA  
POLITEKNIK NEGERI CILACAP  
2022**



## **TUGAS AKHIR**

**PENGEMBANGAN SISTEM INFORMASI  
BADAN EKSEKUTIF MAHASISWA  
POLITEKNIK NEGERI CILACAP  
*INFORMATION SYSTEM DEVELOPMENT OF  
STUDENT EXECUTIVE AGENCY OF  
STATE OF POLYTECHNIC CILACAP***

Oleh

**SANDY PERMANA PUTRA  
NPM 18.02.02.064**

**DOSEN PEMBIMBING :  
ABDUL ROHMAN S., S.T., M.KOM.  
NIDN. 0615058407**

**M. NUR FAIZ, S.KOM., M.KOM.  
NIDN. 0624039301**

**JURUSAN TEKNIK INFORMATIKA  
POLITEKNIK NEGERI CILACAP  
2022**

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Internet sudah menjadi kebutuhan hidup masyarakat, penggunaan internet semakin meningkat seiring berjalannya waktu. Pernyataan ini dapat didukung dengan hasil Survei Sosial ekonomi Nasional (Susenas) pada tahun 2016 – 2020 yang setiap tahunnya memiliki rata-rata peningkatan sebanyak 7,09 persen. Berdasarkan hasil Susenas, persentase penduduk yang menggunakan internet pada tahun 2019 mendapatkan nilai 47,69 persen, kemudian meningkat sebanyak 6,04 persen pada tahun 2020 sehingga mendapatkan nilai 53,73 persen. Ini menandakan bahwa peningkatan penggunaan internet berbanding lurus dengan meningkatnya pengaksesan informasi dan peluang penyebaran informasi melalui internet, selain itu internet juga dibutuhkan oleh banyak sektor termasuk sektor organisasi. Salah satunya adalah organisasi Badan Eksekutif Mahasiswa (BEM) yang berada di Politeknik Negeri Cilacap (PNC).

BEM PNC merupakan organisasi lembaga pemegang kekuasaan tertinggi di bidang eksekutif di kemahasiswaan PNC. Selama waktu berjalannya kepengurusan BEM dalam kegiatan yang dilaksanakan, diperlukan arsip untuk mempermudah proses penyusunan laporan pertanggungjawaban (LPJ) dikemudian hari. Dokumen kegiatan-kegiatan yang disimpan, dapat dijadikan sebagai bahan evaluasi baik lingkup departemen ataupun lingkup BEM PNC secara keseluruhan. Melihat jangka panjang dari arsip tersebut dapat dijadikan sebagai informasi bagi pengurus dalam mengenal kegiatan yang telah terlaksana dan sebagai referensi dalam penyusunan program kerja. Berdasarkan sistem yang sedang berjalan saat ini penyimpanan dokumen masih dalam bentuk fisik (selanjutnya akan disebut dengan *hard file*) dan hanya beberapa dokumen yang disimpan dalam bentuk digital (selanjutnya akan disebut dengan *soft file*), dan *soft file* tersebut tersebar pada anggota-anggota pengurus BEM PNC dan tidak tersimpan secara terpusat. Hal ini mengakibatkan jika

terdapat seorang anggota membutuhkan *soft file* yang diinginkan, seorang anggota tersebut perlu menanyakan seluruh atau seseorang pengurus yang secara khusus menangani terkait *soft file* tersebut. Ketua BEM PNC juga menuturkan bahwa Pembina BEM PNC juga perlu melihat dan membaca *soft file*, namun dengan sistem yang berjalan saat ini akan pengaksesan *soft file* dinilai kurang cepat karena tersebar *soft file* yang belum tahu pasti siapa yang menyimpannya. Sangat dibutuhkan media penyimpanan *online* yang mudah diakses dimanapun dan kapanpun bagi pengurus yang ingin mengakses *soft file* yang dibutuhkan.

*CodeIgniter* merupakan sebuah kerangka kerja (*framework*) pengembangan aplikasi untuk *website developer* menggunakan *PHP* dengan lebih cepat. *CodeIgniter* dibangun menggunakan konsep *Model*, *View*, dan *Controller* (MVC) yang artinya *CodeIgniter* membagi logika data, tampilan antarmuka dan logika pemrograman agar struktur aplikasi menjadi lebih aman dan terstruktur. *CodeIgniter* sudah melalui 4 kali perubahan, dan perubahan keempat ini merupakan perubahan terbesar yang pernah dilakukan oleh tim pengembang *CodeIgniter*, British Columbia Institute of Technology (BCIT). Perubahan ini membuat tim pengembang melakukan *rewrite* atau penulisan ulang pada beberapa skrip kode. Pada 24 Februari 2018, *CodeIgniter* 4 dirilis versi *alpha* dengan banyak pembaruan seperti melakukan instalasi melalui *Command Line Interface* (CLI) hingga fitur pengaturan *environment* yang akan dijalankan pada tahap pengembangan proyek *CodeIgniter* 4 oleh *website developer*. Pada *CodeIgniter* 4 ini memiliki kelebihan berupa ukuran file yang kecil, performa yang cepat, dokumentasi yang jelas dan *developer* tidak diwajibkan untuk mematuhi aturan yang telah dibuat tim pengembang *CodeIgniter*.

Berdasarkan pemaparan penulis diatas, maka penulis berusaha untuk membuat sebuah sistem berupa aplikasi web yang dapat membantu anggota BEM dalam menyimpan informasi-informasi milik BEM PNC secara terpusat dan jelas dengan menggunakan *CodeIgniter* 4 sebagai kerangka kerja *PHP* yang dapat membantu penulis dalam mengembangkan sistem dengan lebih cepat. Penulis berharap dengan

dibuatnya sistem tersebut dapat menjadi sumber informasi bagi seluruh anggota BEM PNC.

## **1.2 Tujuan dan Manfaat**

Tujuan dari penelitian ini adalah menambahkan peluang menyebarkan dan menyimpan informasi BEM PNC secara terpusat melalui internet, dan membantu pengurus dalam menyimpan *soft file* di internet agar mudah diakses dimanapun dan kapanpun.

Manfaat dari penelitian ini adalah sebagai berikut :

- a. Dapat digunakan sebagai sumber referensi untuk penelitian selanjutnya,
- b. Penyimpanan *soft file* menjadi lebih terpusat, jelas dan terdokumentasi dengan baik,
- c. Arsip dapat diakses dengan mudah dan cepat.

## **1.3 Rumusan Masalah**

Bagaimana mengembangkan sebuah sistem yang dapat menyimpan informasi-informasi milik BEM PNC secara *online* dan mudah diakses dimanapun dan kapanpun?

## **1.4 Batasan Masalah**

Adapun batasan masalah penelitian ini sebagai berikut :

- a. Sistem dibuat menggunakan kerangka kerja *CodeIgniter 4*,
- b. Sistem hanya digunakan oleh Pembina, anggota Badan Pengurus Harian (BPH), dan semua Kepala Departemen BEM PNC,
- c. Sistem hanya digunakan untuk mengunggah, melihat, mengunduh, mengganti, dan menghapus *soft file* dan hanya akan menangani *soft file* dalam format berupa *pdf*, *docx*, dan *doc*,
- d. Penulis tidak melakukan *hosting online* pada sistem yang akan dibuat.

## **1.5 Metodologi Penelitian**

Metodologi penelitian adalah proses yang dijalankan untuk memecahkan suatu masalah, yang memerlukan data-data sebagai pendukung terselenggaranya suatu penelitian. Penelitian ini dilakukan di organisasi BEM PNC. Metode penelitian yang digunakan adalah metode deskriptif. Metode deskriptif merupakan metode yang menggambarkan

fakta-fakta dan informasi dalam situasi atau kejadian. Metode ini memiliki 2 tahapan, yaitu tahap pengumpulan data dan tahap pengembangan sistem. Berikut ini penjelasan mengenai 2 tahapan tersebut :

- a. Tahap pengumpulan data dapat diperoleh secara langsung dari objek penelitian, tahapan tersebut diantaranya :
  - Wawancara  
Penulis mengumpulkan data dengan bertanya kepada Ketua BEM PNC mengenai sistem penyimpanan dan pengaksesan dokumen yang sedang berjalan saat ini. Pada tahap ini didapatkan bahwa sistem manajemen informasi mengenai penyimpanan *hard file* dan *soft file*. Hasil dari wawancara yang telah dilakukan penulis dengan Ketua BEM PNC periode 2021/2022 adalah saat ini penyimpanan dokumen/*file* sudah beberapa sudah dalam bentuk *soft file*/digital dan beberapa masih dalam bentuk *hard file*/fisik kertas. Kemudian penyimpanan *soft file* saat ini tidak terkumpul secara terpusat melainkan tersebar ke beberapa anggota.
  - Studi pustaka  
Penulis melakukan pengumpulan data dengan mencari, mempelajari, dan menelaah berbagai data dari sumber seperti buku, modul, jurnal, dan artikel-artikel baik dari perpustakaan ataupun dari internet yang relevan dengan topik yang dibahas dalam penelitian ini.
- b. Tahap pengembangan sistem  
Tahap pengembangan sistem pada penelitian ini menggunakan metode *System Development Life Cycle* (SDLC) dengan model *Rapid Application Development* (RAD). Berikut ini tahapan-tahapan RAD:
  - 1) Menentukan kebutuhan proyek  
Penulis menentukan kebutuhan aplikasi dimulai dengan mewawancarai narasumber yaitu Ketua BEM PNC. Setelah penulis mewawancarai narasumber mengenai aplikasi seperti apa yang diinginkan termasuk fitur-fiturnya, penulis dapat menganalisa dan menentukan

kebutuhan aplikasi sesuai dengan yang diminta oleh narasumber berdasarkan hasil wawancara kebutuhan proyek bersama Ketua BEM PNC.

2) Membuat *prototype*/purwarupa

Penulis membuat *prototype* dari kebutuhan aplikasi yang sudah ditentukan dari tahap pertama. Proses pembuatan dilakukan secara cepat dengan fitur-fitur yang mencukupi kebutuhan disisi *user*.

3) Proses pengembangan dan pengumpulan timbal balik

Penulis meminta narasumber untuk melakukan *testing* dan memberikan timbal balik tentang kekurangan dari *prototype* tersebut. Penulis melakukan pengembangan *prototype* lagi hingga sesuai dengan kebutuhan dan keinginan narasumber. Tahap ini akan terjadi berulang-ulang dan akan berhenti ketika narasumber menyatakan *deal*/setuju dengan *prototype* yang sudah dibuat setelah melewati beberapa perkembangan.

4) Implementasi dan finalisasi produk

Penulis melakukan implementasi secara penuh pada aplikasi yang fungsi-fungsinya sudah terintegrasi, kemudian penulis membuat sebuah dokumentasi mengenai produk tersebut guna mempermudah penggunaan aplikasi sesuai prosedur untuk *end user*

nantinya. Tahap ini adalah langkah terakhir sebelum menyerahkan aplikasi ke *end user*.

## **1.6 Sistematika Penulisan**

Sistematika penulisan dikemukakan dengan maksud agar hasil laporan menjadi lebih terarah dan terstruktur sehingga tidak menyimpang dari permasalahan yang sudah digariskan berdasarkan batasan masalah .

- a) **BAB I Pendahuluan**  
Berisi tentang penjelasan latar belakang masalah, tujuan, manfaat, rumusan masalah, metodologi yang diterapkan, dan sistematika penulisan laporan tugas akhir ini.
- b) **BAB II Dasar Teori**  
Berisi tentang teori dasar dan tinjauan pustaka yang diperoleh dari referensi-referensi yang dipublikasikan secara resmi baik berupa buku, makalah, jurnal, media massa, maupun laporan tugas akhir/penelitian sebelumnya yang diperlukan untuk mendukung pengembangan sistem yang akan dibuat.
- c) **BAB III Perancangan Sistem**  
Berisi dan membahas berbagai aspek yang diperlukan untuk dipertimbangkan dalam pembangunan sistem, antara lain adalah bahan penelitian, analisis sistem yang sedang berjalan, analisis sistem yang akan dikembangkan, analisis kebutuhan perangkat lunak (*software*) dan perangkat keras (*hardware*), rancangan antarmuka dan skenario hasil pengujian sistem.
- d) **BAB IV Hasil dan Pembahasan**  
Berisi tentang implementasi, hasil analisis dari pengujian, dan kuesioner terhadap sistem yang dibuat.
- e) **BAB V Kesimpulan dan Saran**  
Berisi kesimpulan dan saran dari hasil penelitian untuk menyempurnakan sistem pada pengembangan berikutnya.



Halaman ini sengaja dikosongkan

## **BAB II**

### **DASAR TEORI**

#### **2.1 Tinjauan Pustaka**

Sebuah penelitian yang telah dilakukan oleh Ade dan Yuli yang bertujuan untuk membangun sebuah sistem yang dapat membantu staf Kantor Desa Karangrau dalam mengelola arsip surat masuk dan keluar. Sistem dibangun dengan menggunakan metode pengembangan SDLC model *waterfall*, *PHP* dan *Javascript* sebagai bahasa pemrograman dan *MySQL* sebagai media penyimpanan atau basis data. Penelitian tersebut menghasilkan sebuah sistem yang dapat mengelola surat masuk dan keluar milik Kantor Desa Karangrau secara digital.

Penelitian lainnya telah dilakukan Saifudin dan Setiaji. Penelitian tersebut bertujuan untuk membangun sebuah sistem yang dapat menyelesaikan permasalahan kearsipan surat di Kantor Desa Karangsalam Kecamatan Baturaden. Sistem yang berjalan saat itu masih dilakukan secara manual, berarti penyimpanan masih dalam bentuk fisik dan sering terjadi kesalahan dalam penyimpanan dan pencarian surat. Penelitian tersebut menggunakan model *waterfall* sebagai alur pengembangan, dan *black box* sebagai metode pengujian aplikasi. Penelitian tersebut menghasilkan sebuah sistem yang dapat mengelola dan menyimpan data surat dengan baik dan *paperless*.

Penelitian sebelumnya juga telah dilakukan oleh Fadly. Penelitian tersebut bertujuan untuk membangun sistem yang dapat membantu perusahaan Sourci Indonesia dalam mengelola arsip. Penelitian ini menggunakan metodologi Scrum sebagai metodologi perancangannya. Hasil dari penelitian tersebut adalah merancang sistem informasi pengarsipan yang dapat meminimalisir duplikasi arsip dan memudahkan dalam mengelola arsip secara efektif dan efisien.

Pada penelitian ini, penulis bermaksud membangun sistem yang dapat membantu BEM PNC dalam mengarsipkan dokumen-dokumen yang sangat mungkin dibutuhkan sebagai referensi dalam kegiatan maupun kepengurusan berikutnya. Metode perancangan yang penulis

gunakan adalah model *Rapid Application Development* (RAD), dan *Black Box Testing* sebagai metode pengujian aplikasinya.

## **2.2 Dasar Teori**

### **2.2.1 Sistem Informasi (SI)**

Sebuah SI bisa terdiri atas kombinasi yang terorganisasi apapun dari manusia, perangkat keras, perangkat lunak, jaringan komunikasi, sumber data, dan kebijakan serta prosedur yang terorganisasi yang menyimpan, mengambil, mengubah, dan memisahkan informasi dalam sebuah organisasi. Manusia bergantung pada sistem informasi modern untuk berkomunikasi dengan yang lainnya menggunakan berbagai perangkat fisik (perangkat keras), instruksi dan prosedur pemrosesan informasi (perangkat lunak), saluran komunikasi (jaringan), dan data yang tersimpan (sumber data). Walaupun sistem informasi saat ini umumnya dianggap sebagai sesuatu yang dilakukan dengan menggunakan komputer, kita sudah menggunakan sistem informasi sejak awalnya sebuah peradaban. Seperti contoh sistem informasi yang dimaksud adalah sinyal asap untuk berkomunikasi yang memiliki makna semacam instruksi yang ditransmisikan berdasarkan pola dari asap, katalog kartu di perpustakaan yang dirancang untuk menyimpan data tentang buku yang diletakkan secara teratur sehingga pembaca dapat menemukan lokasi dari sebuah buku tertentu berdasarkan judul, nama pengarang, atau berbagai pendekatan lainnya, dan mesin penghitung uang masuk di restoran cepat saji yang dapat melacak produk terjual, waktu penjualan, tingkat persediaan, dan jumlah uang dalam mesin kasir [1].

#### **2.1.2.1 Fundamental SI**

Ketika terdapat begitu banyak aplikasi perangkat lunak, terdapat tiga fundamental untuk semua aplikasi bisnis dari teknologi informasi, yaitu mendukung proses dan kegiatan operasi bisnis, mendukung pengambilan keputusan oleh karyawan dan manajer, dan mendukung strategi bagi keuntungan kompetitif. Pada gambar menggambarkan bagaimana peranan fundamental memengaruhi dalam sebuah organisasi yang umum. Kapanpun, sistem informasi yang dirancang untuk mendukung proses dan kegiatan operasi bisnis mungkin juga menyediakan data untuk, atau menerima data dari, sistem yang berfokus pada pengambilan keputusan bisnis atau mencapai keuntungan

kompetitif. Hal yang sama juga berlaku untuk dua peranan fundamental dari SI yang lainnya. Organisasi yang ada saat ini bekerja keras secara konstan untuk mencapai integrasi sistem mereka yang memberikan fleksibilitas dan dukungan bisnis yang lebih besar lagi dibandingkan yang bisa diberikan oleh sistem individual [1].

#### 2.1.2.2 Definisi Sistem

Sistem didefinisikan sebagai seperangkat komponen yang saling terhubung, dengan sebuah batasan yang jelas, bekerja bersama untuk mencapai sebuah tujuan yang sama dengan menerima masukan dan menghasilkan keluaran dalam sebuah proses transformasi yang terorganisasi. Sistem konsep akan menjadi lebih berguna dengan menyertakan dua elemen tambahan yakni umpan balik dan kendali. Sebuah sistem dengan fungsi umpan balik dan kendali terkadang disebut dengan sistem sibernetik (*cybernetic*), yaitu sebuah sistem yang melakukan pengawasan dan pengaturan sendiri [1]. Berikut ini konsep sistem:

- a. Masukan, melibatkan penangkapan dan perakitan elemen yang masuk kedalam sistem untuk diproses. Sebagai contohnya adalah bahan baku, energi, data, dan usaha manusia yang harus diamankan dan diorganisasi untuk pengolahan,
- b. Pengolahan, melibatkan proses transformasi yang mengubah masukan menjadi keluaran. Contohnya proses manufaktur dan proses bernapas manusia,
- c. Keluaran, melibatkan pemindahan elemen yang telah dihasilkan oleh sebuah proses transformasi ke tujuan akhir mereka. Sebagai contohnya adalah produk jadi, jasa manusia, dan manajemen informasi harus dikirimkan kepada pengguna manusia.
- d. Umpan balik, yaitu mengembalikan data tentang kinerja dari sebuah sistem. Sebagai contoh, data mengenai kecepatan, ketinggian, dan arah dari pesawat terbang merupakan umpan balik bagi pilot pesawat terbang atau pilot otomatis.
- e. Kendali, melibatkan pengawasan dan pengevaluasian umpan balik untuk menentukan apakah sebuah sistem berusaha mencapai tujuannya. Fungsi kendali kemudian membuat

penyesuaian yang diperlukan bagi sebuah masukan sistem dan komponen pemrosesan untuk memastikan bahwa hal itu menghasilkan keluaran yang sesuai. Sebagai contoh, seorang pilot atau pilot otomatis pesawat membuat berita acara penyesuaian setelah mengevaluasi umpan balik dari instrumen untuk memastikan pesawat berada pada jalur yang diinginkan pilot.

#### 2.1.2.3 Definisi Informasi

Informasi adalah suatu pertambahan dalam ilmu pengetahuan yang menyumbangkan kepada konsep kerangka kerja yang umum dan fakta-fakta yang diketahui. Informasi bertumpu pada konteks dan pengetahuan umum si penerima untuk kepentingannya. Informasi yang dihasilkan dari proses pengolahan data dijadikan sebagai sumber daya. Secara tradisional sumber daya terdiri dari manusia, keuangan dan sumber daya material. Setelah dua dasawarsa, informasi diakui sebagai salah satu sumber daya yang sangat penting bagi manajemen dan bagi yang lainnya, didalam keadaan tertentu, informasi merupakan pengganti biaya yang efektif. Informasi dalam hal ini tidak berkaitan dengan penggunaan uang tetapi berkaitan dengan waktu, tanpa pamrih, dan dalam cara yang berbeda dengan sumber daya lainnya. Data yang diolah menjadi informasi harus segar (tidak kedaluwarsa), informasi harus tersedia setiap saat untuk suatu keputusan yang berguna dan nilai suatu informasi menurun sejalan dengan waktu berlalu. Anda dapat mempunyai segudang informasi, tetapi sistem informasi manajemen melindungi pengguna dari informasi yang berlebihan [1].

#### 2.2.2 Basis Data

Sistem basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia disaat dibutuhkan. Pada intinya basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah media untuk menyimpan data agar dapat diakses dengan mudah dan cepat. Sistem informasi tidak dapat dipisahkan dengan kebutuhan akan basis data

apapun bentuknya, berupa file, teks, atau juga *Database Management System* (DBMS).

#### 2.2.2.1 *Database Management System* (DBMS)

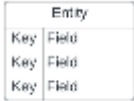
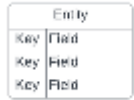





DBMS adalah suatu sistem aplikasi yang digunakan untuk menyimpan, mengelola, dan menampilkan data. Suatu sistem aplikasi bisa disebut DBMS jika memenuhi kriteria :

- a. Menyediakan fasilitas untuk mengelola akses data,
- b. Mampu menangani integritas data,
- c. Mampu menangani akses data yang dilakukan,
- d. Mampu menangani *backup* data.

#### 2.2.2.2 *Entity Relationship Diagram* (ERD)

ERD diartikan sebagai sekumpulan cara atau peralatan untuk menggambarkan data-data maupun objek-objek yang dibuat berdasarkan dan berasal dari dunia nyata yang disebut dengan entitas (*entity*) serta

hubungan (*relationship*) antar entitas tersebut dengan menggunakan beberapa notasi [2]. Komponen pembentuk ERD antara lain :

No	Notasi	Nama	Keterangan
1		<i>Field</i>	<i>Field</i> mewakili bagian dari tabel yang menetapkan atribut entitas. Atribut biasanya dianggap sebagai kolom dalam <i>database</i> .
2		<i>Keys</i>	<i>Keys</i> adalah satu cara untuk mengkategorikan atribut-atribut pada <i>field</i> . <i>Keys</i> digunakan menghubungkan berbagai tabel-tabel di <i>database</i> .
3		One	Instansiasi sebanyak maksimal satu dari sebuah entitas yang dihubungkan ke entitas lain.
4		Many	Instansiasi sebanyak lebih dari satu dari sebuah entitas yang dihubungkan ke entitas lain.
5		Zero or One	Instansiasi sebanyak minimal nol dan maksimal satu dari sebuah entitas yang dihubungkan ke entitas lain.
6		One or Many	Instansiasi sebanyak minimal satu hingga lebih dari satu dari sebuah entitas yang dihubungkan ke entitas lain.
7		Zero or Many	Instansiasi sebanyak minimal nol hingga lebih dari nol dari sebuah entitas yang dihubungkan ke entitas lain.

### 2.2.3 Pemrograman Berorientasi Objek (PBO)

Pemrograman berorientasi objek (PBO) adalah paradigma pemrograman yang merepresentasikan objek-objek yang ada di dunia nyata [3]. Ada empat konsep *object-oriented*, antara lain :

- a. Abstraksi (*Abstraction*)  
Abstraksi bertujuan untuk menyaring properti dan operasi pada suatu objek, sehingga hanya tinggal properti dan operasi yang dibutuhkan saja.
- b. Pewarisan (*Inheritance*)  
Pewarisan adalah kondisi dimana sebagai hasil dari instansiasi sebuah kelas, sebuah objek memiliki semua karakteristik dari kelasnya. Dengan demikian apapun atribut dan operasi dari kelas akan dimiliki pula oleh semua objek yang diwariskan dari kelas tersebut.
- c. Polimorfisme (*Polymorphism*)  
Polimorfisme adalah konsep yang sangat handal bagi pengembang perangkat lunak untuk pemisahan secara jelas diantara sub-sistem yang berbeda. Dengan demikian sebuah sistem akan bisa dimodifikasi secara mudah karena hanya dibutuhkan antarmuka antar kelas. Pada intinya, sebuah kelas memiliki nama operasi yang sama namun hasil akhirnya akan berbeda tergantung parameter dari operasi tersebut.
- d. Enkapsulasi (*Encapsulation*)  
Enkapsulasi merupakan penyembunyian informasi, konsep ini sebenarnya lebih didasari pada fakta yang di dunia nyata bahwa tidak semua hal perlu diperlihatkan. Contoh, untuk memperbesar volume suara pada TV, pengguna hanya perlu menekan satu tombol tertentu saja tanpa perlu tahu bagaimana proses dibelakang itu semua sehingga suara TV sesuai dengan harapan pengguna.

### 2.2.4 Rekayasa Perangkat Lunak (RPL)

Rekayasa perangkat lunak merupakan pembangunan dengan menggunakan prinsip atau konsep rekayasa dengan tujuan menghasilkan perangkat lunak yang bernilai ekonomi yang dipercaya dan bekerja secara efisien menggunakan mesin [4]. RPL lebih fokus pada praktik



pengembangan perangkat lunak dan mengirimkan perangkat lunak yang bermanfaat kepada pengguna. Adapun ilmu komputer lebih fokus pada teori dan konsep dasar perangkat komputer. Rekayasa perangkat lunak lebih fokus pada bagaimana membuat perangkat lunak yang memenuhi kriteria berikut :

- Maintainability, perangkat lunak dapat terus dipelihara setelah perangkat lunak selesai dibuat seiring berkembangnya teknologi dan lingkungan,
- Dependability robust, perangkat lunak dapat diandalkan dengan proses bisnis yang dijalankan dan perubahan yang terjadi,
- Efficient, perangkat lunak dinilai efisien dari segi sumber daya dan penggunaan, dan
- Usability, kemampuan perangkat lunak untuk dipakai sesuai dengan kebutuhan.

#### 2.4.2.1 Metode Pengembangan Sistem

*Software Development Life Cycle* (SDLC) adalah proses mengembangkan atau mengubah suatu sistem perangkat lunak dengan menggunakan model-model dan metodologi yang digunakan orang untuk mengembangkan sistem-sistem perangkat lunak sebelumnya. Model prototipe merupakan model SDLC yang digunakan untuk menyambungkan ketidakpahaman pelanggan mengenai hal teknis dan memperjelas spesifikasi kebutuhan yang diinginkan pelanggan kepada pengembang perangkat lunak [4].



*Prototype* adalah implementasi bagian dari produk software yang secara typical fungsinya dibatasi, reliabilitas rendah, tampilannya miskin, dan kurang ketegasan [5]. Prototype sering dikembangkan secara cepat dalam bahasa tingkat tinggi atau bahasa prototype tertentu, tanpa

memperhatikan kebenaran dan ketegapan dan sebagainya. Berikut ini tahapan pengembangan prototype yang digunakan :

- a. Mengidentifikasi kebutuhan pemakai, analisis sistem mewawancarai pemakai untuk mendapatkan gagasan dari apa yang diinginkan pemakai terhadap sistem,
- b. Mengembangkan prototype, analisis sistem mungkin bekerjasama dengan spesialis informasi lain, menggunakan satu atau lebih peralatan prototyping untuk mengembangkan sebuah prototype,
- c. Menentukan apakah prototype dapat diterima, analisis mendidik pemakai untuk menggunakan prototype dan memberi kesempatan kepada mereka untuk membiasakan diri dengan sistem. Pemakai memberi masukan kepada analisis apakah prototype memuaskan. Jika ya, langkah keempat akan diambil, jika tidak, prototype akan direvisi dengan mengulangi langkah 1, 2, dan 3 dengan pengertian yang lebih baik mengenai kebutuhan pemakai,
- d. Menggunakan prototype, prototype menjadi sistem operasional. Pendekatan ini hanya mungkin jika peralatan prototyping memungkinkan prototype memuat semua elemen penting dari sistem baru.

#### 2.4.2.2 Metode Pengujian Sistem

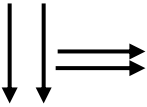


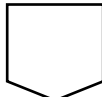


Sistem dalam pengembangannya harus diuji karena proses analisis, perancangan dan juga pemrogramannya yang tidak bebas dari kesalahan. Metode ini diharapkan memiliki mekanisme untuk menentukan data uji yang dapat menguji software secara lengkap dan memiliki kemungkinan tinggi untuk menjumpai kesalahan. Dalam pengujian, diperlukan strategi pengujian agar pengujian dapat terurut. Strategi ini secara terurut adalah pengujian unit, pengujian integritas dan pengujian sistem, perangkat lunak dapat diuji dengan dua cara, salah satunya *Black Box Testing* yaitu pengujian dilakukan dengan mengeksekusi data uji dan mengecek apakah fungsional *software* tanpa menguji desain dan kode program dari sistem. Pengujian ini untuk




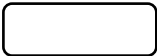




mengetahui fungsi-fungsi, masukan dan keluaran dari software sesuai atau tidak dengan spesifikasi yang dibutuhkan.





#### 2.4.2.3 Flowchart

Flowchart adalah bagan-bagan yang mempunyai arus yang menggambarkan langkah-langkah penyelesaian suatu masalah. Flowchart

merupakan cara penyajian dari suatu algoritma. Berikut ini simbol-simbol flowchart :

Simbol Penghubung/Alur			
No.	Simbol	Nama	Keterangan
1		Arus/Flow	Untuk menyatakan jalannya arus.
2		Communication Link	Untuk menyatakan bahwa adanya transisi suatu data/informasi dari satu lokasi ke lokasi lainnya.
3		Connector	Untuk menyatakan sambungan dari satu proses ke proses lainnya dalam halaman/lembar yang sama
4		Offline Connector	Untuk menyatakan sambungan dari satu proses ke proses lainnya dalam halaman/lembar yang berbeda.
Simbol Proses			
No	Simbol	Nama	Keterangan
1		Process	Untuk menyatakan sambungan dari satu proses ke proses lainnya dalam halaman/lembar yang berbeda.
2		Manual	Untuk menyatakan suatu tindakan yang tidak dilakukan oleh komputer.

3		Decision	Untuk menunjukkan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban, ya/tidak.
4		Predefined Process	Untuk menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberi harga awal.
5		Terminal	Untuk menyatakan permulaan atau akhir suatu program.
6		Keying Operation	Untuk menyatakan segala jenis operasi yang diproses dengan menggunakan suatu mesin yang mempunyai key board.
7		Offline Storage	Untuk menunjukkan bahwa data dalam simbol ini akan disimpan ke suatu media tertentu.
8		Manual Input	Untuk memasukkan data secara manual dengan menggunakan online keyboard.
Simbol Input-Output			
No	Simbol	Nama	Keterangan
1		Input-Output	Untuk menyatakan proses input dan output tanpa tergantung dengan jenis peralatannya.
2		Punched Card	Untuk menyatakan input berasal dari kartu atau output ditulis ke kartu.

3		Magnetic-tape Unit	Untuk menyatakan input berasal dari pita magnetic atau output disimpan ke pita magnetic.
4		Disk Storage	Untuk menyatakan input berasal dari disk atau output disimpan ke disk.
5		Document	Untuk mencetak laporan ke printer.
6		Display	Untuk menyatakan peralatan output yang digunakan berupa layar (video, komputer).

#### 2.4.2.4 Unified Modelling Language (UML)

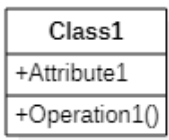





Pemodelan adalah gambaran dari realita yang simpel dan dituangkan dalam bentuk pemetaan dengan aturan tertentu. Pemodelan dapat menggunakan bentuk yang sama dengan realitas misalnya jika seorang arsitek ingin memodelkan sebuah gedung yang akan dibangun maka arsitek tersebut akan memodelkannya dengan membuat sebuah maket (tiruan) arsitektur gedung yang akan dibangun dimana maket itu akan dibuat semirip mungkin dengan desain gedung yang akan dibangun agar arsitektur gedung yang diinginkan dapat terlihat [4].

*Unified Modelling Language* (UML) adalah salah standar bahasa yang banyak digunakan didunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek. UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak. UML merupakan bahasa visual untuk pemodelan dan komunikasi

mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung [4]. Berikut ini diagram-diagram yang terdapat pada UML :

a. *Class Diagram*

*Class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem [4].






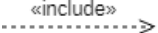
No.	Notasi	Nama	Keterangan
1		<i>Class</i>	Blok pembangunan pada PBO. Bagian atas yaitu <i>class</i> dan bagian tengah mengidentifikasi <i>property</i> /atribut kelas, bagian akhir mendefinisikan <i>method</i> dari sebuah kelas.
2		<i>Association</i>	Relasi yang umum antara 2 kelas, dan dilambangkan dengan garis lurus yang menghubungkan 2 kelas.
3		<i>Composition</i>	Jika sebuah kelas tidak bisa berdiri sendiri dan harus merupakan bagian dari kelas lain, maka kelas tersebut memiliki relasi ini.
4		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri akan mempengaruhi elemen yang bergantung pada elemen yang mandiri.
5		<i>Aggregation</i>	Mengindikasikan keseluruhan bagian <i>relationship</i> dan biasanya disebut sebagai relasi.
6		<i>Generalization</i>	Hubungan dimana objek anak berbagi perilaku dan struktur data dari objek yang ada diatas objek induk.

b. *Use Case Diagram*






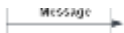



*Use case* diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada didalam sebuah sistem informasi dan siapa saja yang berhak menggunakan

fungsi-fungsiiitu [4]. Berikut ini merupakan komponen-komponen use case diagram :

No	Notassi	Nama	Keterangan
1		<i>Actor</i>	Seseorang atau sesuatu yang berinteraksi dengan sistem yang sedang dikembangkan. Aktor berada diluar lingkup sistem, aktor bersifat eksternal.
2		<i>Use Case</i>	<i>Use case</i> menggambarkan bagaimana aktor akan menggunakan/memanfaatkan sistem.
3		<i>Association</i>	Relasi yang terjadi antar aktor dengan <i>use case</i> .
4		<i>Extend</i>	Memungkinkan suatu <i>use case</i> memiliki kemungkinan untuk memperluas fungsionalitas yang disediakan <i>use case</i> lainnya. Mirip dengan <i>include</i> , namun pada <i>extend</i> tidak harus terjadi apa yang diharapkan.
5		<i>Generalization</i>	Generalisasi digunakan untuk memperlihatkan bahwa beberapa aktor atau use case memiliki sesuatu hal yang bersifat umum. Pada intinya, relasi ini berguna untuk mengelompokkan aktor atau <i>use case</i> .
6		<i>Include</i>	Memungkinkan suatu <i>use case</i> untuk menggunakan fungsionalitas yang disediakan. Include dapat digunakan ketika terdapat dua atau lebih <i>use case</i> yang memiliki fungsi yang identik dapat dipisahkan menjadi suatu <i>use case</i> tersendiri.

### c. Sequence Diagram

Sequence diagram adalah diagram yang menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambar diagram ini maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. [3].

No.	Notasi	Nama	Keterangan
1		<i>Entity Class</i>	Bagian dari sistem yang berisikan sekumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi dasar untuk menyusun basis data.
2		<i>Boundary Class</i>	Berisi sekumpulan kelas yang menjadi tampilan antarmuka atau interaksi antara satu/lebih aktor dengan sistem.
3		<i>Control Class</i>	Suatu objek yang berisi logika aplikasi yang tidak mempunyai tanggung jawab kepada entitas.
4		<i>Message</i>	Simbol mengirim pesan antar class
5		<i>Activation</i>	Berfungsi mewakili sebuah eksekusi operasi dari objek, panjang kotak dalam simbol ini berbanding lurus dengan durasi aktivasi sebuah operasi.
6		<i>Recursive</i>	Menggambarkan pengiriman pesan untuk dikirim ke dirinya sendiri.
7		<i>Lifeline</i>	Berupa garis titik-titik yang terhubung dengan objek, disepanjang garis ini terdapat activation.

Halaman ini sengaja dikosongkan

## DAFTAR PUSTAKA

- [1] J. A. O'Brien, G. M. Marakas, L. N. Puspitasari, dan H. Kurnia, *Sistem Informasi Manajemen*, 9 ed. Jakarta: Salemba Empat, 2014.
- [2] A. Nugroho, *Rational Rose untuk Pemodelan Berorientasi Objek*, 1 ed. Bandung: Informatika Bandung, 2005.
- [3] Munawar, *Pemodelan Visual Dengan UML*, 1 ed. Yogyakarta: Penerbit Graha Ilmu, 2005.
- [4] R. A.S dan S. M, *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*, 3 ed. Bandung: Informatika Bandung, 2015.
- [5] A.-B. bin L. B, *Rekayasa Perangkat Lunak*, 1 ed. Yogyakarta: Graha Ilmu, 2006.

Halaman ini sengaja dikosongkan

