

## BAB II LANDASAN TEORI

### 2.1 Definisi Model Pengembangan Perangkat Lunak

Dalam penelitian ini, peneliti menggunakan *prototype* sebagai model pengembangan perangkat lunak dengan definisi dan tahapan sebagai berikut:

#### 2.1.1 Model *Prototype*

*Prototype* merupakan model pengembangan sistem yang menggunakan *prototype* untuk menggambarkan sistem, sehingga pengguna atau pemilik sistem mempunyai gambaran pengembangan sistem yang akan dilakukannya. Teknik ini sering digunakan apabila pemilik sistem tidak terlalu menguasai sistem yang akan dikembangkannya, sehingga dia memerlukan gambaran dari sistem yang akan dikembangkannya tersebut [6].

#### 2.1.2 Tahapan *Prototype*

Menurut [6] dalam melakukan perancangan sistem yang akan dikembangkan dapat menggunakan metode *prototype*. Metode ini cocok digunakan untuk mengembangkan sebuah perangkat yang akan dikembangkan kembali. Tahapan dalam metode perancangan *prototype* dapat dilihat pada gambar dibawah ini.

Dalam model *prototype* terdapat beberapa tahapan yang harus dilakukan, yaitu:

1. Pengumpulan kebutuhan.

Pelanggan dan pengembang bersama-sama mendefinisikan format seluruh perangkat lunak, mengidentifikasi semua kebutuhan, dan garis besar sistem yang akan dibuat.

2. Membangun *prototyping*.

Membuat *prototyping* dengan membuat perancangan sementara yang berfokus pada penyajian kepada pelanggan (misalnya dengan membuat *input* dan format *output*).

3. Evaluasi *prototyping*.  
Evaluasi ini dilakukan oleh pelanggan apakah *prototyping* yang sudah dibangun sudah sesuai dengan keinginan pelanggan. Jika sudah sesuai maka langkah 4 akan diambil. Jika tidak *prototyping* direvisi dengan mengulang langkah 1, 2, dan 3.
4. Mengkodekan sistem.  
Dalam tahap ini *prototyping* yang sudah di sepakati diterjemahkan ke dalam bahasa pemrograman yang sesuai.
5. Menguji sistem.  
Setelah sistem sudah menjadi suatu perangkat lunak yang siap pakai, harus dites dahulu sebelum digunakan. Pengujian ini dilakukan dengan *black box*.
6. Evaluasi sistem.  
Pelanggan mengevaluasi apakah sistem yang sudah jadi sudah sesuai dengan yang diharapkan. Jika iya, langkah 7 dilakukan, jika tidak, ulangi langkah 4 dan 5.
7. Menggunakan sistem.  
Perangkat lunak yang telah diuji dan diterima pelanggan siap untuk digunakan.

## 2.2 Definisi Metode Pengembangan Perangkat Lunak

Dalam penelitian ini, peneliti menggunakan *object oriented programming* (OOP) sebagai metode pengembangan perangkat lunak, dengan penjelasan sebagai berikut:

### 2.2.1. *Object Oriented Programming* (OOP)

*Object oriented programming* (OOP) merupakan metode pemrograman yang mengikuti konsep-konsep berorientasi objek yakni *encapsulation* atau pengkapsulan, *inheritance* atau pewarisan dan *polimorfisme* atau keberagaman dalam bentuk lain. Metode ini juga dapat diartikan sebagai cara bagaimana suatu sistem dalam perangkat lunak dibangun melalui pendekatan objek secara otomatis yang didasarkan pada berbagai prinsip pengelolaan kompleksitas yang meliputi

rangkaian aktivitas analisis, perancangan, pemograman, dan pengujian berorientasi objek [7].

### 2.3 Definisi *Tools* Pengembangan Perangkat Lunak

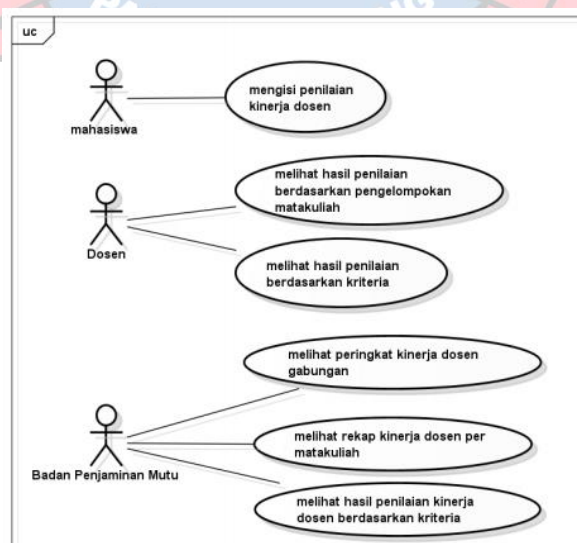
Dalam penelitian ini, peneliti menggunakan *unified modelling language* (UML) sebagai *tools* pengembangan perangkat lunak, dengan definisi dan macam-macam UML yang digunakan, yakni sebagai berikut:

#### 2.3.1. *Unified Modelling Language* (UML)

*Unified modelling language* (UML) merupakan sekumpulan alat yang digunakan untuk melakukan abstraksi terhadap sebuah sistem atau perangkat lunak berbasis objek. UML juga menawarkan sebuah standar untuk merancang model dari sebuah sistem yang dapat dibuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi atau bahkan jaringan dan bahasa pemograman apapun [8].

##### 1. *Use case diagram*

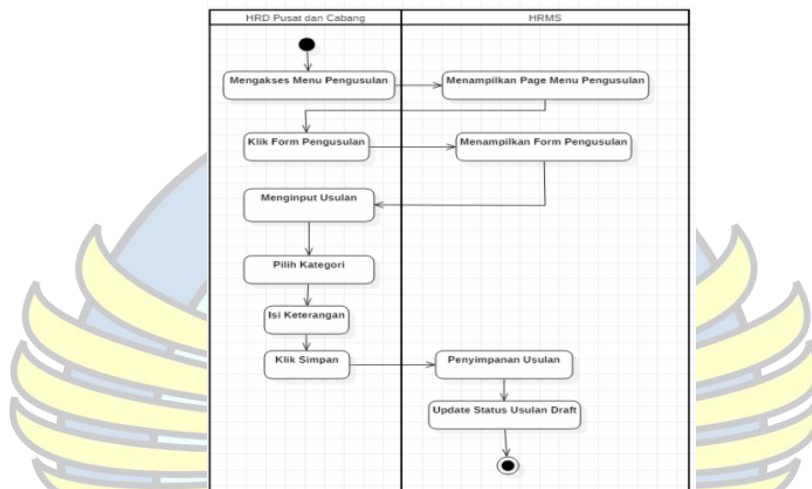
*Use case diagram* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. Diagram ini memiliki 2 fungsi yaitu mendefinisikan fitur apa yang harus disediakan oleh sistem dan menyatakan sifat dari sudut pandang *user* [8].



Gambar 2.1 Contoh *Use Case Diagram* [9]

2. *Activity diagram*

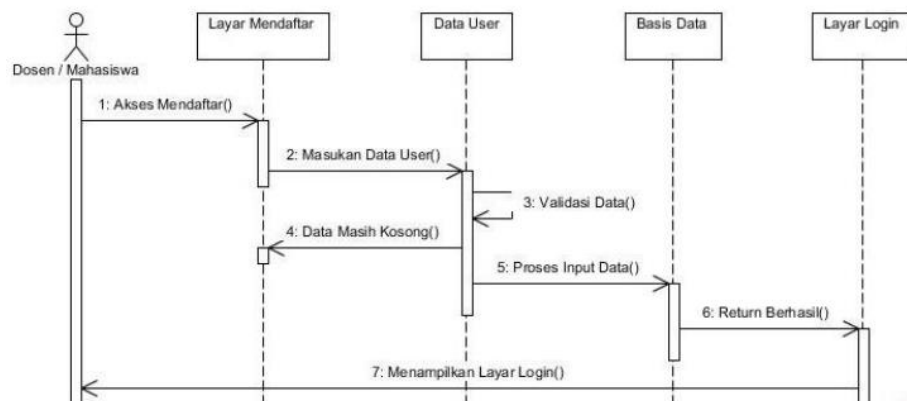
*Activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. *Activity diagram* adalah tipe khusus dari diagram state yang memperlihatkan aliran dari satu aktivitas ke aktivitas lainnya dalam suatu sistem dan berfungsi untuk menganalisa proses [8].



**Gambar 2.2 Contoh Activity Diagram [9]**

3. *Sequence diagram*

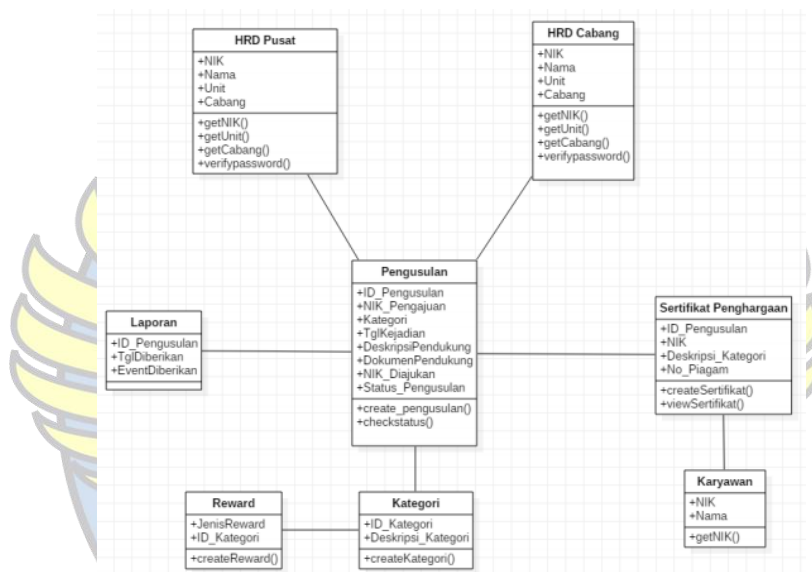
*Sequence diagram* menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Berikut ini adalah daftar simbol dari *sequence diagram* [7].



**Gambar 2.3 Contoh Sequence Diagram [9]**

#### 4. Class diagram

*Class diagram* merupakan salah satu diagram utama dari UML untuk menggambarkan *class* atau *blueprint object* pada sebuah sistem. Analisis pembentukan *class diagram* merupakan aktivitas inti yang sangat mempengaruhi arsitektur perangkat lunak yang dirancang hingga ke tahap pengkodean. Dalam *class diagram* juga digambarkan bagaimana interaksi hubungan antar *class* dalam sebuah konstruksi perangkat lunak seperti hubungan asosiasi, agregasi, komposisi, *inheritance* dan *generalization* [8].



Gambar 2.4 Contoh Class Diagram [9]

### 2.4 Definisi Teori Pendukung

Dalam penelitian ini, terdapat beberapa teori pendukung yang peneliti ambil dari berbagai sumber sebagai referensi. Dibawah ini adalah beberapa teori yang dimaksud:

#### 2.4.1 Arsip

Arsip adalah dokumen tertulis (surat, akta, dan sebagainya), lisan, atau bergambar dari waktu yang lampau, disimpan dalam media tulis (kertas), elektronik (pita kaset, pita video, disket komputer, dan sebagainya), biasanya

dikeluarkan oleh instansi resmi, disimpan dan dipelihara di tempat khusus untuk referensi [10].

#### 2.4.2 Dokumen

Dokumen adalah surat yang tertulis atau tercetak yang dapat dipakai sebagai bukti keterangan (seperti akta kelahiran, surat nikah, surat perjanjian) [10].

#### 2.4.3 Java

*Java* didesain sebagai bahasa pemrograman yang bisa berjalan di berbagai komputer termasuk telepon genggam. Bahasa ini dikembangkan pertama kali oleh James Gosling saat masih bergabung di *sun microsystems*. Bahasa *java* awal dirilis tahun 1995. Sintaksnya banyak meniru sintaks yang terdapat pada C dan C++, hanya saja model objek dibuat lebih sederhana serta dukungan rutin level bawah yang minimal [11].

*Java* sedari awal didesain untuk memanfaatkan dependensi implementasi seminimal mungkin. Karenanya aplikasi *java* mampu berjalan di beberapa *platform* sistem operasi yang berbeda, seperti *linux*, dan *windows* tanpa perlu menulis ulang program lagi. Yang dianggap paling sukar di *java* adalah meng-*install* infrastruktur pemrograman, karena ada banyak hal yang perlu di-*install* terlebih dahulu [11].

#### 2.4.4 Android

*Android* merupakan sistem operasi berbasis *linux* yang mencakup sistem operasi, middleware dan aplikasi. *Android* dapat digunakan untuk telepon seluler, seperti telepon pintar (*smartphone*) dan komputer tablet (PDA). *Android* pada mulanya didirikan oleh Andy Rubin, Rich Miner, Nick Sears, dan Chris White pada tahun 2003 [12]. Versi *android* yaitu:

1. Android 1.0 (API level 1)
2. Android 1.1 (API level 2)
3. Android 1.5 Cupcake (API level 3)

4. Android 1.6 Donut (API level 4)
5. Android 2.0 Eclair (API level 5)
6. Android 2.0.1 Eclair (API level 6)
7. Android 2.1 Eclair (API level 7)
8. Android 2.2–2.2.3 Froyo (API level 8)
9. Android 2.3–2.3.2 Gingerbread (API level 9)
10. Android 2.3.3–2.3.7 Gingerbread (API level 10)
11. Android 3.0 Honeycomb (API level 11)
12. Android 3.1 Honeycomb (API level 12)
13. Android 3.2 Honeycomb (API level 13)
14. Android 4.0–4.0.2 Ice Cream Sandwich (API level 14)
15. Android 4.0.3–4.0.4 Ice Cream Sandwich (API level 15)
16. Android 4.1 Jelly Bean (API level 16)
17. Android 4.2 Jelly Bean (API level 17)
18. Android 4.3 Jelly Bean (API level 18)
19. Android 5.0 Lollipop (API level 21)
20. Android 6.0 Marshmallow (API level 23)
21. Android 7.0 Nougat (API 24)
22. Android 7.1 Nougat (API 25)
23. Android 8.0 Oreo (API 26)
24. Android 8.1.0 Oreo (API 27)
25. Android 9 Pie (API 28)
26. Android 10 (API 29)

#### **2.4.5 Android Studio**

*Android studio* ini adalah lingkungan pengembangan baru dan terintegrasi dengan penuh, yang telah di rilis oleh *google* untuk sistem operasi *android* dan di rancang untuk menjadi peralatan baru dalam pengembangan aplikasi dan memberi alternaif selain *eclips* yang saat ini menjadi IDE yang banyak di pakai [13].

#### 2.4.6 PHP (*Hypertext Preprocessor*)

PHP merupakan salah satu bahasa pemrograman berbasis *web* yang ditulis oleh dan untuk pengembang *web*. PHP pertama kali dikembangkan oleh Rasmus Lerdorf, seorang pengembang *software* dan anggota tim *apache*, dan diliris pada akhir tahun 1994. PHP dikembangkan dengan tujuan awal hanya untuk mencatat pengunjung pada website pribadi Rasmus Lerdorf. Pada rilis keduanya, ditambahkan *form interpreter*, sebuah tools untuk melakukan penerjemahan perintah SQL. Sejak saat itu, PHP mulai diterima sebagai sebuah bahasa pemrograman baru yang sangat diminati [14].

PHP adalah pemrograman *interpreter* yaitu proses penerjemahan baris kode sumber menjadi kode mesin yang dimengerti komputer secara langsung pada saat baris kode dijalankan. PHP disebut sebagai pemrograman *server side programming*, hal ini dikarenakan seluruh prosesnya dijalankan pada server [14].

#### 2.4.7 MySQL

MySQL merupakan salah satu *software* untuk *database server* yang banyak digunakan, MySQL bersifat *open source* dan menggunakan SQL. MySQL bisa dijalankan di berbagai *platform* misalnya *windows*, *linux* dan lain sebagainya [14].

#### 2.4.8 Pengujian *Black Box*

Pengujian *black box* berfokus pada persyaratan fungsional perangkat lunak. Dengan demikian, pengujian *black box* memungkinkan perekayasa perangkat lunak mendapatkan serangkaian kondisi *input* yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program. Pengujian *black box* bukan merupakan alternatif dari teknik *white box*, tetapi merupakan pendekatan komplementer yang kemungkinan besar mampu mengungkapkan kelas kesalahan daripada metode *white box*. Pengujian *black box* berusaha menemukan kesalahan dalam kategori sebagai berikut: (1) fungsi-fungsi yang tidak benar atau hilang, (2) kesalahan interface, (3) kesalahan dalam struktur data



atau akses *database eksternal*, (4) kesalahan kinerja, (5) inisialisasi atau kesalahan terminasi [15].

## 2.5 Ringkasan Penelitian Terdahulu

Beberapa penelitian terdahulu kemudian di ringkas dengan menggunakan tabel.

**Tabel 2.1 Tabel Penelitian Terdahulu**

No	Peneliti	Judul	Hasil
1.	Elisa Asrat, Said Achmad Kabiru Rafiie, 2020 [1]	Penerapan Sistem Aplikasi E-Arsip Pada Kegiatan Pengarsipan Surat-Menyurat Dibagian Umum Sekretariat Kabupaten Aceh Barat	Penggunaan sistem aplikasi E-arsip dalam kegiatan pengarsipan surat masuk dan surat keluar dibagian umum setdakab aceh barat sudah cukup baik akan tetapi masih belum maksimal seperti halnya masih terdapat kendala-kendalan saat sistem aplikasi E-arsip tersebut beroperasi seperti harus terkoneksi jaringan internet dan sering terjadi kemacetan saat dihubungkan ke mesin <i>scan</i> .
2.	Liza Rozana, Rahmat Musfikar, 2020 [2]	Analisis Dan Perancangan Sistem Informasi Pengarsipan Surat Berbasis Web Pada Kantor Lurah Desa Dayah Tuha	Dengan adanya sistem informasi pengarsipan berbasis <i>web</i> ini maka lebih memudahkan dalam kegiatan pengarsipan pada kantor lurah desa dayah tuha sebagai solusi pemecahan permasalahan yang ada, dokumen arsip dapat disimpan dan dikelola dengan baik serta lebih mudah dalam proses pencariannya.
3.	Wahyu	Aplikasi	Berdasarkan hasil pengujian

	Suratman, Fauziah, Ratih Titi Komala Sari, 2021 [3]	Elektronik Arsip (E-Arsip) Surat Berbasis <i>Web</i> Menggunakan Metode First In First Out (FIFO)	aplikasi di atas, bahwa aplikasi telah lulus uji dan aplikasi elektronik arsip (e-arsip) surat berbasis web dengan metode First In First Out (FIFO) di Universitas Nasional sudah bisa digunakan sesuai dengan tujuannya. Dan dari 1000 jumlah data statistik surat masuk dan keluar yang semakin besar tiap tahunnya bisa diatasi dengan aplikasi e-arsip ini.
4.	Muhammad Dedi Irawan, Selli Aprilla Simargolang, 2018 [4]	Implementasi E-Arsip Pada Program Studi Teknik Informatika	Aplikasi e-arsip memiliki fungsi scan yang kegunaan menginputkan scan surat yang ingin disimpan sehingga dapat meminimalkan kehilangan dan rusaknya dokumen-dokumen penting.
5.	Anisah, Delpiah Wahyuningsih, Ellya Helmud, Tedy Suwanda, Parlia Romadiana, Devi Irawan, 2021 [5]	Rancang Bangun Sistem Informasi Manajemen Arsip Digital	Aplikasi yang dibangun berguna bagi admin, sekretaris bidang, Arsiparis, dan bagian kepegawaian dan umum untuk melakukan proses pendataan arsip, proses penyimpanan arsip, proses penelusuran arsip, dan proses menampilkan informasi terkait arsip yang mereka butuhkan. Aplikasi yang dibuat dapat mengefisienkan tempat penyimpanan, memudahkan pendataan, memudahkan

			penelusuran dan monitoring terhadap arsip yang ada.
--	--	--	---

