

BAB 2

TINJAUAN PUSTAKA

2.1 Teori Dasar

Teori dasar merupakan teori yang menjadi suatu teori yang lain untuk digunakan dalam penelitian ini, teori dasar pada bab ini yaitu:

2.1.1 *Android*



Gambar 2. 1 *Logo Android*

Sumber: id.wikipedia.org (2022)

Android merupakan suatu sistem operasi yang umum digunakan untuk perangkat seperti *handphone* dan *tablet* berbasis *Linux* yang mencakup *middleware*, aplikasi, dan sistem operasi, android juga *platform* yang terbuka bagi mereka yang ingin menciptakan dan mengembangkan sendiri. Tidak seperti saingan *platform* lainnya contoh seperti sistem operasi IOS dari *Apple* yang bersifat tertutup yang berarti pengguna tidak bisa mengembangkan sesuai keinginan *user*. *Android* awalnya dikembangkan oleh *Android Inc.* dan mendapat dukungan anggaran dari *Google* yang kemudian akhirnya pada tahun 2005 di akuisisi oleh *Google* kemudian dirilis resmi tahun 2007. *Android* merupakan pendatang baru *platform mobile* yang membuat piranti lunak sebuah *smartphone*, kemudian untuk mengembangkannya dibangunlah *Open Handset Alliance* atau aliansi bisnis yang diciptakan bertujuan

mengembangkan standar perangkat terbuka, konsorsium (pembiayaan bersama suatu proyek) tersebut diikuti 34 perusahaan seperti *Google, Qualcomm, T-Mobile, Nvidia*, dan masih banyak lagi (Nurhidayati dan Nur 2021).

Android didasari sistem operasi yang dilisensikan dibawah naungan GNU sistem operasi komputer yang terdiri dari perangkat lunak bebas, *General Public License version 2 (GPLv2)*, yang dikenal dengan istilah *Copyleft* yang artinya setiap perbaikan pihak ketiga harus terus jatuh dibawah ketentuan yang ditetapkan. distribusi *Android* dibawah lisensi *Apache Software (ASL/Apache2)*, yang memungkinkan untuk distribusi seterusnya (Nurhidayati Nur 2021).

Android sama seperti sistem operasi pada umumnya yang memiliki fungsi sebagai yaitu:

1. Pengelola sumber daya perangkat mulai dari penyimpanan, memori dan lain lain
2. Sebagai jembatan antara pengguna dan aplikasi
3. Memberikan antarmuka bagi pengguna

Sejak berdiri sistem operasi *android* sampai sekarang banyak terjadi perkembangan dari waktu ke waktu, berikut jenis dan versi *android*:

Tabel 2.1 *Versi Android*

| Versi | Nama | API Level | Rilis |
|---------------------|--------------------|-----------|-------------------|
| Android 1.0 | Alpha | 1 | 23 September 2008 |
| Android 1.1 | Beta | 2 | 9 Februari 2009 |
| Android 1.5 | Cupcake | 3 | 27 April 2009 |
| Android 1.6 | Donut | 4 | 15 September 2009 |
| Android 2.0 | Eclair | 5 | 26 Oktober 2009 |
| Android 2.0.1 | Eclair | 6 | 3 Desember 2009 |
| Android 2.1 | Eclair | 7 | 12 Januari 2010 |
| Android 2.2–2.2.3 | Froyo | 8 | 20 Mei 2010 |
| Android 2.3–2.3.2 | Gingerbread | 9 | 6 Desember 2010 |
| Android 2.3.3–2.3.7 | Gingerbread | 10 | 9 Februari 2011 |
| Android 3.0 | Honeycomb | 11 | 22 Februari 2011 |
| Android 3.1 | Honeycomb | 12 | 10 Mei 2011 |
| Android 3.2 | Honeycomb | 13 | 15 Juli 2011 |
| Android 4.0–4.0.2 | Ice Cream Sandwich | 14 | 19 Oktober 2011 |
| Android 4.0.3–4.0.4 | Ice Cream Sandwich | 15 | 16 Desember 2011 |
| Android 4.1 | Jelly Bean | 16 | 9 Juli 2012 |
| Android 4.2 | Jelly Bean | 17 | 13 November 2012 |
| Android 4.3 | Jelly Bean | 18 | 24 Juli 2013 |
| Android 4.4 | Kitkat | 19 | 31 Oktober 2013 |
| Android 5.0 | Lollipop | 21 | 25 Juni 2014 |
| Android 6.0 - 6.0.1 | Marshmallow | 23 | 28 Mei 2015 |
| Android 7.0 | Nougat | 24 | 9 Maret 2016 |
| Android 7.1 | Nougat | 25 | 19 Oktober 2016 |
| Android 8.0 | Oreo | 26 | 21 Agustus 2017 |
| Android 8.1.0 | Oreo | 27 | 5 Desember 2017 |
| Android 9 | Pie | 28 | 6 Agustus 2018 |
| Android 10 | Android Q | 29 | 3 September 2019 |

Sumber: bhinneka.com (2021)

2.1.2 Arsip

Arsip adalah hal yang mendasar dan harus sangat diperhatikan dalam semua kegiatan administrasi, karena arsip menjadi informasi data yang tersimpan mengenai segala hal yang dikerjakan dalam suatu rutinitas yang terus berlangsung, dalam administrasi arsip berperan sangat penting dalam segala aspek mulai dari perencanaan, bahan laporan dan sebagai bahan utama dalam mengambil suatu keputusan sebuah lembaga karna tanpa adanya sebuah arsip maka tidak mungkin suatu lembaga atau instansi dapat beroperasi secara teratur dan tertib, pengolahan arsip yang tidak terstruktur dengan baik dapat menyulitkan pencarian data, terlebih ketika dibutuhkan dalam keadaan yang mendesak, maka dari itu perlunya pengolahan arsip yang teroganisir dengan baik (Irawan and Simargolang 2018).



Gambar 2. 2 Arsip

Sumber: djkn.kemenkeu.go.id (2021)

2.1.3 E-Arsip

E-Arsip merupakan sebuah sistem kearsipan yang terstruktur secara elektronik dan dapat digunakan oleh siapa saja sebagai suatu perlengkapan yang berguna dalam mengelola suatu dokumen terkait sistem kearsipan berbentuk audio, visual, berkas untuk menjamin terpenuhinya kebulatan dan ketentuan hukum dikemudian hari (Irawan and Simargolang 2018).

2.1.4 Sistem Kearsipan

Sistem kearsipan ada untuk kepentingan operasional dengan aturan bahwa informasi yang akurat digunakan untuk kepentingan yang benar dengan biaya yang minim sebagai subsistem dalam manajemen kearsipan, yang bekerja untuk mencapai tujuan agar arsip dapat tersusun rapi dalam bagian informasi siap pakai. Sistem kearsipan harus mencakup semua manajemen kearsipan. Daur hidup arsip mencakup proses pendistribusian, penggunaan, dan penyimpanan arsip permanen (Irawan and Simargolang 2018).

2.1.5 Jenis Arsip

Menurut kutipan dari artikel (Astuti 2021) Arsip mempunyai pengertian berupa catatan yang ditulis dengan tertata dalam bentuk kertas atau bahan yang menyangkut keterangan tentang suatu subjek suatu permasalahan yang dibuat manusia untuk mempermudah daya ingat yang lampau.

Sedangkan menurut ANRI (Arsip Nasional Republik Indonesia) yaitu sebuah lembaga pemerintahan menerangkan bahwa Arsip merupakan segala berkas, naskah dalam bentuk apapun yang dapat dikelola.

2.1.5.1 Jenis Arsip Berdasarkan Sifatnya

Adapun jenis arsip dilihat dari sifatnya yaitu:

1. Arsip tidak penting

Adalah arsip yang mempunyai kegunaan sebagai sumber informasi seperti: surat undangan dan pemberitahuan.

2. Arsip biasa

Adalah arsip yang awalnya penting digunakan, kemudian setelah digunakan atau waktu berlakunya berlalu menjadi tidak dibutuhkan lagi, seperti: surat lamaran kerja dan surat tagihan listrik dan air, dan juga surat sewa rumah.

3. Arsip penting

Adalah arsip yang berhubungan dengan masa lampau dan masa depan, sehingga harus disimpan, seperti: surat kontrak perjanjian.

4. Arsip vital (sangat penting)

Adalah arsip yang dijadikan patokan selama – lamanya, contoh seperti Pancasila dan undang – undang dasar.

5. Arsip rahasia

Adalah arsip yang hanya boleh diketahui suatu organisasi tertentu, seperti: taktik pemasaran.

2.1.5.2 Jenis Arsip Berdasarkan Keahlian

Beberapa jenis arsip menurut keahliannya yaitu:

1. Arsip asli merupakan dokumen utama yang langsung dari cetak printer, tanda tangan, cap stempel dsb.
2. Arsip salinan merupakan dokumen yang dibuat setelah dokumen asli namun sama dengan arsip asli, biasanya memiliki ciri khas hitam putih.
3. Arsip tembusan merupakan dokumen setelah arsip asli yang sama tetapi dituju pada pihak tertentu.

2.1.5.3 Jenis Arsip Berdasarkan Kegunaannya

Adapun jenis arsip dilihat dari gunanya yaitu:

1. Arsip bernilai hukum seperti surat kuasa.
2. Arsip pendidikan seperti kurikulum dan program pembelajaran tahunan.
3. Arsip administrasi seperti surat keputusan organisasi.
4. Arsip informasi seperti surat pemberitahuan.
5. Arsip keuangan seperti laporan keuangan bulanan.

2.1.5.4 Jenis Arsip Berdasarkan Fungsinya

Adapun jenis arsip dilihat dari fungsinya yaitu:

1. Arsip dinamis merupakan arsip yang digunakan terkait penyelenggaraan dan perencanaan administrasi negara secara langsung dalam sehari – hari, berikut beberapa arsip dinamis:

- a. Arsip aktif merupakan arsip yang dipakai terus – menerus dalam kegiatan kerja, dan juga sering digunakan untuk keperluan khusus.
 - b. Arsip semi aktif merupakan arsip yang jarang digunakan tetapi masih diperlukan.
 - c. Arsip inaktif merupakan arsip yang sangat jarang digunakan, juga sebagai keterangan dan referensi.
2. Arsip statis merupakan kebalikan dari arsip dinamis yaitu arsip yang tidak dipergunakan secara langsung seperti laporan tahunan.

2.1.5.5 Jenis Arsip Berdasarkan Fisik

Arsip yang dilihat dari jenis fisik yaitu:

1. Arsip berbentuk lembaran, seperti surat – surat, laporan bulanan dan tahunan, notulen rapat, surat perjanjian, surat pemberitahuan dan lain – lain.
2. Arsip berbentuk selain lembaran, seperti audio, flashdisk, kaset, dan lain – lain.

2.1.5.6 Jenis Arsip Berdasarkan Pemilik

Berikut arsip berdasarkan jenis pemiliknya yaitu:

1. Lembaga Pemeritahan
 - a. Arsip nasional di Indonesai seperti ANRI (Arsip nasional republik Indonesia) merupakan lembaga non pemerintahan yang bertanggung

jawab kepada presiden untuk melaksanakan tugas di bidang kearsipan sesuai perundang – undangan yang berlaku.

- b. Arsip nasional bagian ibukota daerah tingkat 1 termasuk daerah tingkat 1 seperti ibukota Jakarta yang seterusnya dinamakan arsip nasional daerah.
2. Instansi pemerintahan dan swasta
 - a. Arsip primer dan sekunder berbeda dari segi keaslian, arsip primer merupakan arsip asli sedangkan arsip sekunder adalah arsip salinan dari arsip primer.
 - b. Arsip sentral dan unit. berbeda dari segi penyimpanannya, arsip sentral disimpan di pusat penyimpanan sedangkan arsip unit disimpan di setiap organisasi masing – masing.

2.1.5.7 Jenis Arsip Berdasarkan Subjek

Beberapa macam arsip berdasarkan subjek dan isinya yaitu:

1. Arsip keuangan contoh seperti laporan masuk dan keluar bulanan hingga tahunan, struk pembayaran dan lain – lain.
2. Arsip pendidikan contoh seperti absensi siswa dan guru, buku rapot, surat bukti masih berkuliah disuatu universitas dan lain – lain.
3. Arsip kepegawaian contoh seperti riwayat kerja, surat keputusan pengangkatan dan lain – lain.
4. Arsip pemasaran contoh seperti brosur penawaran, daftar harga barang, daftar barang masuk dan keluar dan lain – lain.

2.1.5.8 Jenis Arsip Berdasarkan Masalah

Adapun jenis arsip dilihat berdasarkan permasalahannya yaitu:

1. *Financial record* ialah arsip berupa catatan tentang keuangan.
2. *Sales record* ialah arsip yang menyangkut penjualan.
3. *Production record* ialah arsip terkait masalah profilsu.
4. *Personal record* ialah arsip yang berkaitan dengan masalah kepegawaian
5. *Inventory record* ialah arsip yang berhubungan langsung dengan barang inventaris.

2.1.5.9 Jenis Arsip Berdasarkan Kekuatan Hukum

Adapun jenis arsip berdasarkan kekuatan sumber hukum yaitu:

1. Arsip autentik yakni arsip yang memiliki tanda tangan orisinal bukan salinan sebagai bukti kebenaran yang dapat digunakan sebagai bukti aturan hukum yang valid.
2. Arsip tidak autentik kebalikan dari arsip autentik yakni arsip yang telah disalin dari yang aslinya.

2.1.6 Perancangan

Menurut penelitian dari (Latif and Pratama 2015) istilah perancangan bisa disebut juga desain yang dimaknai sebagai proses penciptaan objek baru, proses dalam membuat sebuah desain basis data bertujuan dalam membangun sebuah

sistem yang tertara sesuai rencana awal, sehingga rancangan dapat terpenuhi dengan baik.

2.1.7 Implementasi

Menurut penelitian dari (Latif and Pratama 2015) istilah perancangan bisa disebut juga desain yang dimaknai sebagai proses penciptaan objek baru, proses dalam membuat sebuah desain basis data bertujuan dalam membangun sebuah sistem yang tertara sesuai rencana awal, sehingga rancangan dapat terpenuhi dengan baik.

Implementasi merupakan suatu kegiatan untuk mendapatkan suatu hasil dengan perencanaan yang sesuai dengan tujuan dan mengacu kepada sasaran kebijakan, dimana pelaksana menjalankan kegiatan sesuai dengan aktivitas kegiatan (Irawan and Simargolang 2018).

Menurut (Rifan Aditya 2021) tujuan dari implementasi yaitu sebagai terapan dalam mewujudkan rencana yang disusun dengan teratur. Karena dalam menerapkan rencana akan ada tujuan yang akan di targetkan, sehingga implementasi bisa dianggap sebagai pencapaian suatu tujuan terikat. Implementasi juga berarti menguji rencana kebijakan yang disusun, guna mengetahui dampak dan manfaat yang diperoleh di masa yang akan datang.

2.1.8 Database

Database adalah sekumpulan informasi data yang tersimpan di komputer secara sistematis dapat diperiksa dan digunakan program komputer untuk

memperoleh basis data tersebut, perangkat lunak yang digunakan untuk mengelola dan memanggil *query* basis data disebut sistem manajemen basis data contoh seperti SQLite dan *firebase* (Nurhidayati and Nur 2021).

2.1.9 TPQ (Taman Pendidikan Al -Qur'an)

Taman Pendidikan Al – Qur'an atau biasa dikenal TPQ adalah sebuah lembaga Pendidikan dan Pengajaran yang menyelenggarakan Pendidikan jenis keagamaan Islam bersifat nonformal, bertujuan untuk memberikan pengajaran membaca Iqro dan Al – Qur'an sejak dini, sehingga anak – anak memahami dinul islam yang diajarkan Nabi dan Rasul dari dulu hingga saat ini. Pendidikan karakter setiap anak merupakan karakter religious berdasarkan nilai – nilai keagamaan, karna merupakan langkah awal dalam menumbuhkan sifat akhlakul karimah.



Gambar 2. 3 Logo TPQ

Sumber: salmawa.com(2021)

Dalam tahapannya TPQ adalah lembaga pendidikan Al-Qur'an yang merupakan tingkatan lanjut dari TKQ (Taman Kanak - kanak Al-Qur'an). Kemudian setelah TPQ tingkatan lanjutannya berupa TQA (Ta'limul Qur'an lil Aulad). Dalam buku panduan yang diterbitkan Kemenag (Kementrian Agama), pendidikan TPQ menempuh waktu selama 3 tahun terbagi atas 6 semester (Singorejo 2018).

2.2 Teori Khusus

Teori yang digunakan peneliti dan berhubungan dengan judul dalam skripsi ini yaitu:

2.2.1 *Android Studio*

Android Studio adalah *Software* pengembangan baru dan terintegrasi penuh yang telah dirilis *Google* untuk *Android*, *Android Studio* dirancang dalam pengembangan aplikasi sebagai alternatif lain dari *Eclipse* yang menjadi IDE (*Integrated Development Environment*) merupakan program komputer yang memiliki fasilitas yang diperlukan dalam pembangunan perangkat lunak yang sering digunakan.



Gambar 2.4 Logo *Android Studio*

Sumber: developer.android.com(2022)

Proyek yang dibangun pertama kali di *Android Studio* berisi beberapa *file resource* dan kode sumber yang meliputi *Manifest*, *Gradle Scripts*, *res* dan masih banyak lagi yang ada dalam direktori *app* dan *gradle*, peralihan ke sistem *gradle* memberikan fleksibilitas dalam menyatukan lingkungan untuk pengembangan semua perangkat *Android*, *software* ini memungkinkan melihat secara langsung perubahan visual yang dilakukan pada saat coding berlangsung, kita dapat juga melihat perbedaan dari hasil setiap perangkat *Android* yang di konfigurasi dalam resolusi yang berbeda - beda secara spontan (Nurhidayati Nur 2021).

2.2.2 Java

Java merupakan sebuah bahasa pemrograman populer yang serbaguna yang dapat mengembangkan aplikasi pada *mobile* (terutama android), *desktop*, *web*, *server*, *game*, *database* dan lain – lain, *Java* adalah bahasa pemrograman yang berorientasi pada objek yang berarti sebuah bahasa yang membentuk atau mengatur desain aplikasi sesuai dengan objek sebagai bidang data yang memiliki atribut. Sehingga memudahkan untuk membangun, mendesain, dan mengembangkan program *Java* dengan cepat dan tertata (Anjuliani et al. 2015).

Menurut (Rizki Sari Dewi 2020) *Java* adalah bahasa pemrograman tingkat tinggi, berorientasi objek, dan program *Java* terdiri dari bagian-bagian yang disebut kelas. Kelas terdiri dari metode yang bekerja dan mengembalikan informasi setelah pekerjaan selesai. Pemrogram *Java* bekerja dengan kumpulan kelas di perpustakaan kelas *Java* yang disebut *Java Application Programming Interface (API)*. Kelas-kelas ini diatur ke dalam kelompok yang disebut paket. *Java API* menyediakan fungsionalitas yang cukup untuk membuat applet dan aplikasi yang kuat.

Kelebihan yang ada bahasa program java yakni:

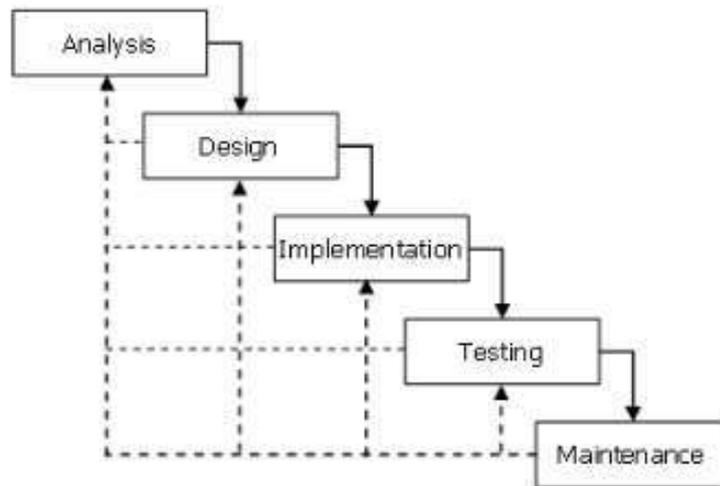
1. Pengembangan yang mudah,
2. bersifat *multiplatform* atau universal, bisa dikembangkan di semua *platform*
3. Sintaks yang mudah untuk membangun script
4. Bahasa pemrograman berorientasi objek sesuai dengan tampilan aplikasi tersebut
5. Dinamis dan mudah menyesuaikan dengan programmer
6. Memiliki komunitas yang besar

2.2.3 Metode *Waterfall*

Dalam pengembangan perangkat lunak peneliti menggunakan Model *Waterfall* yang merupakan salah satu model dari SDLC (*Software Development Life Cycle*), Metode *Waterfall* bersifat *plan – driven* maksudnya proses dimana semua kegiatan proses yang direncanakan terlebih dahulu dan kemajuan diukur terhadap rencana tersebut seperti aliran yang mengalir kebawah (Dhanuari Indra Bastari1, Fajar Pradana2 2017).

Model *Waterfall* bersifat berurut dalam tahap penyelesaian satu per satu dan lanjut ke fase berikutnya hanya ketika tahap sebelumnya sudah dilakukan dimulai

dari proses perencanaan, analisa, desain, dan implementasi pada suatu sistem (Gunawan and Saputro 2018).



Gambar 2.5 Alur Metode *Waterfall*

Sumber: Gunawan (2018)

2.2.4 UML (*Unified Modeling Language*)

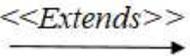
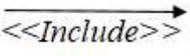
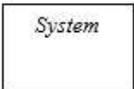
UML (Unified Modeling Language) adalah metodologi dalam pengembangan suatu sistem berorientasi objek dan alat pendukung untuk mengembangkan sistem, juga untuk menciptakan pemodelan bahasa yang bisa dipahami baik oleh manusia ataupun mesin yang dapat mendokumentasikan, menspesifikasikan, dan membangun perangkat lunak. Alat bantu perancangan berorientasi objek berbasis *UML* yaitu:

1. *Use Case diagram*

Menurut (Hutabri and Putri 2019) dan (Visual Paradigm 2019) *Use Case diagram* digunakan untuk mengetahui fungsi dalam sistem informasi, siapa

yang berhak menggunakan fungsi tersebut, *use case* merupakan pemodelan untuk kelakuan suatu sistem yang dibuat

Tabel 2.2 Notasi *Use Case Diagram*

| Nama | Simbol | Keterangan |
|-----------------------|---|--|
| <i>Use Case</i> |  | Pertukaran pesan antara unit dan aktor |
| <i>Actor</i> |  | Orang yang berperan dan berhubungan dengan sistem yang dibuat |
| <i>Association</i> |  | Komunikasi antara aktor dengan use case |
| <i>Extends</i> |  | Relasi tambahan use case dengan use case yang lain, arah panah ke arah use case yang ditambah |
| <i>Generalization</i> |  | Hubungan umum ke khusus, use satu memiliki fungsi umum, dari pada yang lain, arah panah menunjuk ke use case umum |
| <i>Include</i> |  | Hubungan dua use case berbeda, use case yang ditambahkan dengan use case tambahan. Use case yang ditambahkan memerlukan use case tambahan untuk menjalankan kegunaan sebagai syarat menjalankan use case |
| <i>System</i> |  | Seluruh sistem dapat menjangkau semua modul ini yang menggambarkan batas sistem secara keseluruhan |

Sumber : (Hutabri and Putri 2019) dan (visual-paradigm 2022)

2. Activity Diagram

Diagram yang mana dapat memodelkan proses – proses yang dilakukan sebuah sistem, *activity diagram* merupakan pengembangan dari *Use Case* yang memiliki alur aktivitas.

Tabel 2.3 Notasi *Activity Diagram*

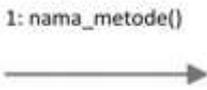
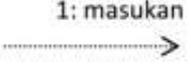
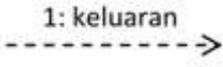
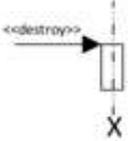
| Nama | Simbol | Keterangan |
|----------------------------------|---|---|
| Status awal dan akhir |  | Status awal dan akhir dari semua proses aktifitas sistem |
| Aktivitas |  | Kegiatan yang dilakukan sistem |
| <i>Decision</i> (percabangan) |  | Status untuk menentukan suatu pilihan seperti iya atau tidak |
| <i>Join</i> (gabungan) |  | Penggabungan satu atau lebih aktifitas menjadi satu |
| <i>Swimlane</i> |  | Memisahkan antara kegiatan sistem dan user untuk aktifitas yang di proses |
| <i>Fork</i> |  | Pemecahan satu aktifitas menjadi paralel |
| <i>Control flow</i> |  | simbol untuk mengurutkan aktivitas |

Sumber : andydharmalau.com(2017)

3. Sequence Diagram

Menggambarkan *behavior* objek pada *use case* dengan menjelaskan waktu hidup objek dari pesan yang dikirim dan diterima oleh sebuah masing – masing objek.

Tabel 2.4 Notasi *Sequence Diagram*

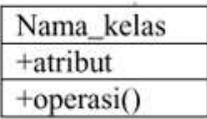
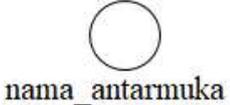
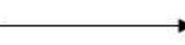
| Nama | Simbol | Keterangan |
|------------------------------|---|---|
| <i>Actor</i> |  | Adalah orang, sistem lain, atau proses diluar sistem, yang berhubungan dengan sistem yang dibangun, aktor bisa berupa orang atau sistem dan bukan hanya orang |
| <i>Lifeline</i> |  | Garis hidup yang menerangkan kehidupan objek |
| <i>Object</i> |  | Adalah orang yang melakukan interaksi pesan antar kegiatan |
| <i>Activation time</i> |  | Adalah simbol melaporkan bahwa objek dalam keadaan aktif |
| Pesan tipe <i>Create</i> |  | Pernyataan sebuah objek yang membuat objek baru |
| Pesan tipe <i>Call</i> |  | Adalah pernyataan sebuah objek memanggil metode diri sendiri atau yang lain |
| Pesan tipe <i>Send</i> |  | Adalah simbol mengirimkan informasi kepada objek lain berupa masukan atau data |
| Pesan tipe <i>Return</i> |  | Adalah simbol pengambilan dan keluaran hasil kepada objek tertentu berupa masukan atau data |
| Pesan tipe <i>Destroy</i> |  | Merupakan simbol yang menyatakan bahwa objek mengakhiri hidup, dari objek lain, jika ada <i>create</i> maka ada <i>destroy</i> |

Sumber: (Hutabri and Putri 2019)

4. *Class Diagram*

Merupakan hubungan antar kelas dalam model desain dari suatu sistem, juga memperlihatkan tanggung jawab entitas yang menentukan perilaku sistem dan objek (Irawan and Simargolang 2018).

Tabel 2.5 Notasi *Class Diagram*

| Nama | Simbol | Keterangan |
|-----------------------------|---|---|
| <i>Class</i> |  | Adalah simbol yang menjelaskan kelas yang ada dalam struktur sistem, berupa atribut dan operasi |
| <i>Interface</i> |  | Sama dengan kelas namun dinyatakan tanpa isi dan atribut kelas |
| <i>Association</i> |  | Asosiasi adalah simbol relasi umum |
| <i>Directed Association</i> |  | Adalah simbol relasi antarkelas yang berarti satu kelas digunakan kelas lain |
| Generalisasi |  | Adalah relasi antarkelas berarti umum ke khusus |
| <i>Dependency</i> |  | Adalah simbol relasi ketergantungan |
| <i>Aggregation</i> |  | Adalah simbol realsi yang berarti semua <i>part</i> |

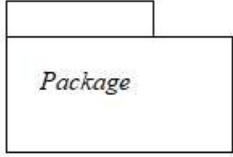
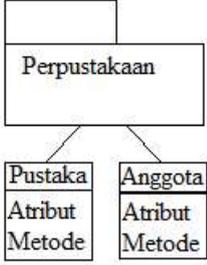
Sumber: (Hutabri and Putri 2019)

5. *Package Diagram*

package diagram biasanya berfungsi untuk menggolongkan atribut *diagram UML* yang berbeda secara berkelompok ke dalam kelas pembangunan yang

lebih atas yakni sebuah *package, diagram* ini pada dasarnya hanya menampilkan hubungan ketergantungan pada *class*.

Tabel 2.6 Notasi *Package Diagram*

| Nama | Simbol | Keterangan |
|----------------|---|--|
| <i>Package</i> |  | Sebuah bungkus satu atau lebih dari elemen dalam diagram UML |
| Elemen Dalam |  | Elemen yang ada dalam <i>package diagram</i> digambarkan dari dalam <i>package</i> |
| Elemen Luar |  | Elemen yang ada dalam <i>package diagram</i> digambarkan dari luar <i>package</i> |

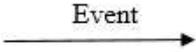
Sumber: M A.S., Rosa dan Shalahuddin (2016)

6. State Diagram

Menggambarkan alur keadaan yang dilewati objek dalam sebuah *class*, karena setiap kejadian menimbulkan perpindahan aktivitas, aktivitas dari objek adalah pengelompokkan dari satu atau lebih nilai atribut. bersifat dinamis menunjukkan keadaan pada sistem, memuat aktivitas, serta *event*.

Diagram ini terutama krusial buat menampilkan sifat bergerak maju berdasarkan antarmuka, kelas, kerja sama dalam desain sistem yg reaktif.

Tabel 2.7 Notasi *State Diagram*

| Nama | Simbol | Keterangan |
|---------------------------|---|---|
| <i>Start/ Status Awal</i> |  | Status awal merupakan <i>state</i> atau keadaan awal suatu sistem mulai |
| <i>End/ Status Akhir</i> |  | Status akhir merupakan <i>state</i> atau keadaan berakhir suatu sistem |
| <i>Event</i> |  | <i>Event</i> merupakan kegiatan merubah status mesin |
| <i>State</i> |  | Status atau <i>state</i> merupakan keadaan sistem pada saat tertentu, dan dapat berubah ketika ada <i>event</i> |

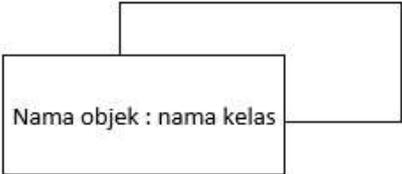
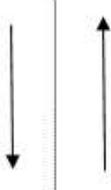
Sumber: M A.S., Rosa dan Shalahuddin (2016)

7. *Communication Diagram*

Diagram ini menunjukkan interaksi sesama objek seperti *sequence diagram*, namun lebih menonjolkan penekanan pada fungsi dari masing – masing objek, setiap *message* memiliki nomor 1 untuk yang tertinggi, seperti

halnya *sequence diagram*, namun lebih memfokuskan kegiatan objek ketika *message* diantarkan.

Tabel 2.8 Notasi *Communication Diagram*

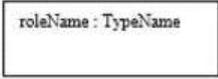
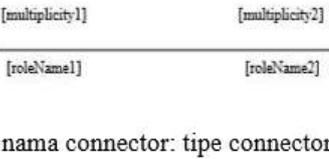
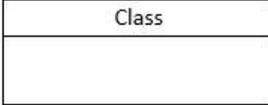
| Nama | Simbol | Keterangan |
|-----------------|---|--|
| Objek |  | Status awal merupakan <i>state</i> atau keadaan awal suatu sistem mulai |
| <i>Link</i> |  | Hubungan antar - objek yang merelasikan antar objek atau pada dirinya sendiri  |
| <i>Stimulus</i> |  | <i>Stimulus</i> atau arah pesan yang bekerja, jika sebuah <i>link</i> memiliki dua arah pesan yang berbeda maka digambarkan dua arah pada kedua sisi link  |

Sumber: M A.S., Rosa dan Shalahuddin (2016)

8. Composite Structure Diagram

Merupakan *diagram* yang memperlihatkan secara rinci tentang struktur dalam *classifier*, juga hubungan antar sistem yang memiliki perilaku *classifier*, diagram ini bersifat statis yang menjelaskan sekitar internal kelas.

Tabel 2.9 Notasi *Composite Structure Diagram*

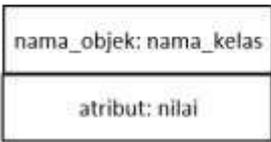
| Nama | Simbol | Keterangan |
|------------------|---|--|
| <i>Property</i> |  | <p><i>Property</i> merupakan satu set dari sebuah <i>instance</i></p> <p><i>roleName</i>: peran / nama / identitas</p> <p><i>TypeName</i>: tipe kelas(harus ada)</p> |
| <i>Connector</i> |  | <p><i>Connector</i> adalah komunikasi antar 2 <i>instance</i></p> <p>connName: nama <i>connector</i> (opsional)</p> <p>ConnType: tipe <i>connector</i> (opsional)</p> |
| <i>Port</i> |  | <p><i>Port</i> merupakan cara yang dipakai pada diagram tanpa menampilkan detail <i>internal system</i></p> <p><i>Port</i> digambarkan dalam kotak kecil yang menempel dalam sebuah <i>property</i> jika dapat diakses <i>public</i>, jika didalam <i>property</i> bersifat <i>protected</i></p> |
| <i>Class</i> |  | <p>Jika yang dijabarkan merupakan struktur <i>class</i>/kelas</p> |

Sumber: M A.S., Rosa dan Shalahuddin (2016)

9. Object Diagram

Object diagram adalah suatu gambaran mengenai objek dalam sistem pada satu waktu, disebut sebagai *diagram* perintah karena lebih menunjukkan perintah antar kelas dan objek, *diagram* ini sama dengan *class diagram* namun yang yaitu diagram objek menampilkan objek dan hubungan antar objek yang berfungsi untuk memudahkan analisis untuk menyingkap *detail* tambahan *class*.

Tabel 2.10 Notasi *Object Diagram*

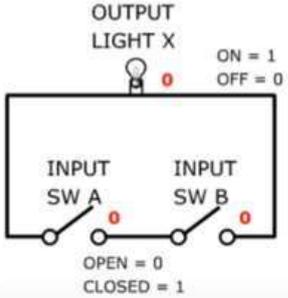
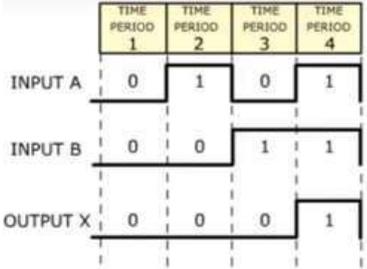
| Nama | Simbol | Keterangan |
|---------------|---|---|
| <i>Object</i> |  | <i>Object</i> /objek pada kelas yang berjalan dari sistem yang dijalankan |
| <i>Link</i> |  | Hubungan antar objek |

Sumber: M A.S., Rosa dan Shalahuddin (2016)

10. Timing Diagram

Timing diagram digunakan dalam memperlihatkan perilaku objek selama rentan waktu tertentu, juga menunjukkan perubahan keadaan tugas berlebihan dari pengelompokkan dalam merespon *event* besar.

Tabel 2.11 Aliran *Timing Diagram*

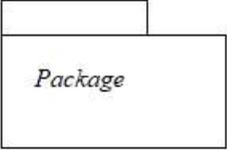
| Simbol | Keterangan | | | | | | | | | | | | | | | | | | | | |
|---|--|---------------|---------------|---------------|---------------|---------|---|---|---|---|---------|---|---|---|---|----------|---|---|---|---|--|
|  <p>OUTPUT LIGHT X ON = 1 OFF = 0</p> <p>INPUT SW A INPUT SW B</p> <p>OPEN = 0 CLOSED = 1</p> | <p>Contoh aliran sirkuit menggambarkan operasi diagram, status <i>OPEN</i> = 0 artinya <i>switch</i> dalam posisi terbuka (tidak terhubung), sedangkan status <i>CLOSED</i> = 1 adalah posisi <i>switch</i> terhubung, lampu akan menyala, status <i>ON</i> = 1, jika kedua <i>switch</i> terhubung.</p> | | | | | | | | | | | | | | | | | | | | |
|  <table border="1" data-bbox="359 824 726 1093"> <thead> <tr> <th></th> <th>TIME PERIOD 1</th> <th>TIME PERIOD 2</th> <th>TIME PERIOD 3</th> <th>TIME PERIOD 4</th> </tr> </thead> <tbody> <tr> <td>INPUT A</td> <td>0</td> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>INPUT B</td> <td>0</td> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>OUTPUT X</td> <td>0</td> <td>0</td> <td>0</td> <td>1</td> </tr> </tbody> </table> | | TIME PERIOD 1 | TIME PERIOD 2 | TIME PERIOD 3 | TIME PERIOD 4 | INPUT A | 0 | 1 | 0 | 1 | INPUT B | 0 | 0 | 1 | 1 | OUTPUT X | 0 | 0 | 0 | 1 | <p>Aliran waktu pada diagram dibaca dari kiri ke kanan, lampu akan menyala jika kedua <i>switch</i> atau <i>INPUT A</i> dan <i>INPUT B</i> dalam posisi <i>CLOSED</i> = 1, jika salah satu <i>switch</i> tidak terhubung maka lampu tidak akan hidup</p> |
| | TIME PERIOD 1 | TIME PERIOD 2 | TIME PERIOD 3 | TIME PERIOD 4 | | | | | | | | | | | | | | | | | |
| INPUT A | 0 | 1 | 0 | 1 | | | | | | | | | | | | | | | | | |
| INPUT B | 0 | 0 | 1 | 1 | | | | | | | | | | | | | | | | | |
| OUTPUT X | 0 | 0 | 0 | 1 | | | | | | | | | | | | | | | | | |

Sumber: M A.S., Rosa dan Shalahuddin (2016)

11. *Component Diagram*

Component diagram digunakan dalam merancang dan mengembangkan suatu sistem yang berbasis komponen karena berpusat pada objek dan desain, *diagram* ini juga dikombinasikan *diagram* lainnya untuk menunjukkan sirkulasi fisik pada modul perangkat lunak.

Tabel 2.12 Notasi *Component Diagram*

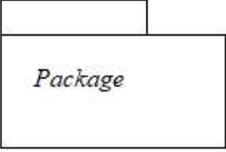
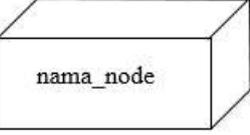
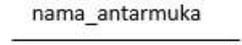
| Nama | Simbol | Keterangan |
|-------------------|---|---|
| <i>Package</i> |  | Merupakan bungkusan dari satu atau lebih komponen |
| <i>Component</i> |  | <i>Component</i> /komponen sistem |
| <i>Dependency</i> |  | <i>Dependency</i> /ketergantungan antar komponen, panah mengarah pada komponen yang digunakan |
| <i>Interface</i> |  | Sebagai antarmuka komponen |
| <i>Link</i> |  | Hubungan antar komponen |

Sumber: M A.S., Rosa dan Shalahuddin (2016)

12. Deployment Diagram

Diagram ini berguna dalam menampilkan rincian komponen pada *deploy* dalam struktur sistem, komponen yang digunakan terletak pada server atau *hardware* yang memiliki sifat fisik.

Tabel 2.13 Notasi *Deployment Diagram*

| Nama | Simbol | Keterangan |
|-------------------|---|--|
| <i>Package</i> |  | Merupakan bungkusan dari satu atau lebih komponen |
| <i>Node</i> |  | <i>Node</i> mengacu pada <i>hardware</i> /perangkat keras, <i>software</i> /perangkat lunak tidak dibuat sendiri, jika dalam <i>node</i> disertakan komponen untuk menstabilkan rancangan maka komponen yang dimasukkan harus sesuai dengan yang didefinisikan di dalam diagram komponen |
| <i>Dependency</i> |  | <i>Dependency</i> /ketergantungan antar komponen, panah mengarah pada komponen yang digunakan |
| <i>Link</i> |  | Hubungan antar komponen |

Sumber: M A.S., Rosa dan Shalahuddin (2016)

13. *Interaction Overview Diagram*

Merupakan gabungan antara *activity diagram* dan *sequence diagram*. karena saling menghubungkan kerjasama antara kedua notasi diagram tersebut (Arni 2021).

Tabel 2.14 Notasi *Interaction Overview Diagram*

| Nama | Simbol | Keterangan |
|----------------------------------|---|--|
| <i>Initial</i> |  | Status awal ketika sebuah kegiatan dipanggil |
| <i>Final</i> |  | Status akhir ketika sebuah keseluruhan kegiatan selesai |
| <i>Decision</i> (percabangan) |  | Unsur suatu diagram yang menunjukkan pilihan benar dan salah |
| Activity |  | Kegiatan yang dilakukan sistem saat sedang berlangsung |
| <i>Control flow</i> |  | Konnektor yang menghubungkan dua node, dan juga sebagai jembatan aliran antar node |

Sumber: ansoriweb.com (2022)

2.2.5 *Black Box Testing*

Menurut (Ibnu 2022) *Black box testing* merupakan sebuah metode dalam pengujian fungsionalitas suatu *software* yang dilakukan secara penuh, menjangkau masukan dan keluaran sistem tanpa terikat *internal program*, biasanya pengujian dilakukan agar dapat menampilkan sudut pandang penguji seperti melihat kotak hitam, pengujian yang dilakukan sangat penting untuk melihat adanya masalah dalam aplikasi yang dibuat sebelum diluncurkan.

1. Tipe *Black box testing*

terdapat 3 macam pengujian yang dapat digunakan yaitu:

a. *Functional testing*

Merupakan pengujian fitur dan fungsi rincian suatu *software*, bertujuan untuk memeriksa dan menilai kepehaman pengguna aplikasi dalam melakukan pengisian akun, *testing* ini berfokus pada pengujian fitur penting dalam sebuah aplikasi dan hubungan antara aktifitas secara menyeluruh

b. *Non – functional testing*

Pengujian yang dilakukan untuk fitur tambahan agar dapat mengetahui bagaimana suatu perintah dan tugas berjalan, jenis *testing* ini juga untuk memeriksa apakah aplikasi mampu berjalan dimacam – macam perangkat dengan spesifikasi yang berbeda – beda

c. *Regression Testing*

Dalam *test* ini akan diuji apakah aplikasi yang ditingkatkan mengalami penurunan, *testing* ini dilakukan pemeriksaan aspek fungsionalitas dan non fungsionalitas, untuk aspek fungsionalitas akan diperiksa apakah fitur – fitur sebelumnya dapat bekerja dengan baik pada versi terbaru, sedangkan pada aspek nonfungsionalitas akan diperiksa apakah performa menjadi lambat dalam pembaharuan versi terbaru.

2. Teknik – teknik *Black box testing*

Beberapa teknik yang biasanya dilakukan dalam pengujian perangkat lunak yaitu:

a. *All pair testing*

Dikenal dengan *pairwise testing*, pengujian ini dilakukan sebagai pengujian semua kemungkinan seluruh hubungan antar sistem berdasarkan inputnya.

b. *Boundary value analysis.*

Teknik yang bertujuan mencari kesalahan diluar *software*

c. *Cause-effect graph*

Teknik ini berfokus pada statistik sebagai barometernya, teknik ini menggambarkan hubungan antara penyebab kesalahan dan akibatnya

d. *Equivalence partitioning*

Teknik ini bekerja dengan membagi masukan menjadi beberapa bagian partisi dari beberapa perangkat lunak

e. *Fuzzing*

Teknik pencarian kesalahan aplikasi dengan cara menginput data yang tidak benar

f. *Orthogonal array testing*

Teknik ini dilakukan jika masukkan berskala kecil, namun berat jika digunakan dalam skala besar

g. *State transition*

Teknik ini bertujuan sebagai pengujian mesin dan UI dalam struktur grafik.

Black box testing mempunyai kelebihan yaitu untuk bisa melakukan pengujian pengguna tidak perlu memiliki kemampuan yang spesifik, pengguna

hanya perlu memahami dan melihat sudut pandang yang baik dalam menggunakan *software*. dan juga *test* ini dapat dilakukan dengan cepat dari awal hingga tahap pengembangan selesai dilaksanakan.

2.2.6 Adobe XD



Gambar 2.6 Logo *Adobe XD*

Sumber: adobe.com(2022)

Adobe telah menaruh pengetahuan desain grafis terbaiknya kedalam wadah Adobe Experience Design atau lebih sering dikenal Adobe XD, banyak pengguna Adobe yang sudah membuat design UX dengan menggunakan aplikasi lain seperti photoshop dan illustrator, namun creative cloud layanan dari adobe.inc membangun suatu alat khusus untuk desain UX dikenal Adobe XD yang memiliki banyak fitur menarik membuat para user mudah dalam membuat desain UI dan UX.

Program istimewa yang membuat penggunaanya dengan mudah dan cepat dalam merangkai gambar dan membangun prototipe situs web dan aplikasi sendiri.

Adobe XD merupakan alat yang disediakan gratis oleh Adobe untuk kebutuhan desain UX dan UI bermacam *platform* seperti ponsel genggam, tablet, web dan perangkat lainnya (Nastainullah 2020).

2.2.7 *Firestore*



Gambar 2.7 Logo *Firestore*

Sumber: Nurhidayati (2021)

Firestore adalah suatu layanan *database online* yang dapat mempercepat sekaligus mempermudah pekerjaan para *developer* aplikasi dalam membangun aplikasi, karena dengan *firebase* para *developer* bisa berfokus pada pengembangan dan pembangunan aplikasi tanpa harus memberikan upaya yang lebih pada bagian *back end*, oleh karena itu *firebase* banyak diminati sebab banyaknya fitur yang menarik (Nurhidayati and Nur 2021).

2.2.8 Microsoft Visio



Gambar 2.8 Logo *Microsoft Visio*

Sumber: wikipedia.org (2022)

Microsoft visio merupakan salah satu aplikasi dari *Microsoft office* yang dirilis oleh *Microsoft corporation*, banyak digunakan sebagai alat untuk membuat desain *diagram, diagram* yang bisa dibangun menggunakan *Microsoft visio* sangat banyak, pengguna hanya tinggal menetikkan ke dalam pencarian *template diagram* apa yang ingin mereka bangun, karena *Microsoft visio* menyediakan banyak *template* seperti *template business, flowchart, general, network*, dan masih banyak lagi (Aulia et al. 2020).

2.3 Penelitian Terdahulu

Beberapa lampiran terdahulu mengenai E - Arsip didalam jurnal yang menjadi referensi bagi peneliti yaitu:

1. Menurut (Irawan and Simargolang 2018) Penelitian yang berjudul **“Implementasi E-Arsip Pada Program Studi Teknik Informatika”**. Membahas tentang pengelolaan arsip khususnya arsip berbasis elektronik, dan masih banyak ditemukan arsip kertas. Hal ini mengakibatkan

banyaknya arsip kertas menumpuk, sehingga menimbulkan bermacam masalah dengan tempat penyimpanan, biaya pemeliharaan, pengelola, fasilitas, ataupun faktor lain penyebab kerusakan arsip. Dalam mengatasi permasalahan tersebut dicarilah solusi dengan metode pengumpulan data berupa observasi dan studi Pustaka, kemudian menghasilkan pemograman e-arsip yang berbasis web, yang dapat digunakan dalam membantu proses penyelenggara kearsipan yang sesuai kebutuhan.

2. Penelitian yang dilakukan (Gunanto and Sudarmilah 2020) berupa **“Pengembangan Website E-Arsip Di Kantor Kelurahan Pabelan”**. Bertempat di kantor kelurahan pabelan yang mana masih menggunakan pengarsipan secara manual berbentuk surat dimana dalam penyediaan surat harus dilakukan manual yang menyebabkan lambatnya kinerja di kantor. Dengan metode *Waterfall* yang merupakan metode pilihan peneliti dalam menyelesaikan masalah untuk mengembangkan aplikasi ini secara bertahap. Hasil dari pengembangan e - arsip ini nanti akan adanya sistem pengarsipan yang terkomputerisasi dengan baik sehingga mempermudah tenaga kantor untuk menyimpan dan menyediakan surat yang masuk dan keluar di kantor kelurahan pabelan tersebut.
3. Penelitian (Kalinda 2019) yang dilakukan tentang **“Pengelolaan Arsip Berbasis Digital Oleh Pegawai Di Kantor Pengadilan Agama Ciamis”**. Menemukan adanya permasalahan dilapangan seperti kurang mengertinya pegawai dalam pengarsipan surat pada komputer, juga kurangnya sumber daya yang memadai seperti tenaga pengelola dan faktor lainnya yang berada

di ruangan kesekretariatan, menyebabkan keterlambatan penanganan arsip VC keluar dan masuk, sehingga kinerja tenaga pengelola dalam menyelesaikan tugasnya belum maksimal, untuk menyelesaikan masalah tersebut peneliti menggunakan metode kualitatif, cara yang digunakan dalam mengatasi kendala pengelolaan arsip berbasis digital meliputi tenaga pengelola yang bertugas pengelola arsip diikutkan dalam program magang pengelolaan kearsipan yang benar, mengajukan pembuatan gedung arsip dan fasilitas pendukung pengelolaan kearsipan.

4. Menurut penelitian (Latif and Pratama 2015) berjudul **“Perancangan Sistem Informasi Manajemen Arsip Elektronik (E-Arsip) Berbasis Microsoft Access Pada PT. Hi-Test”** kearsipan sangat diperlukan dalam sebuah perusahaan sebagai daya ingat kegiatan dalam rutinitas sehari – hari, akan tetapi masalah yang terjadi di PT. Hi-Test masih menerapkan sistem informasi konvensional dan banyak waktu yang dihabiskan untuk menemukan kembali arsip perusahaan, metode yang digunakan dalam penelitian ini adalah metode eksplorasi, metode yang dipakai antara lain analisis, perancangan sistem, desain, pengujian, kemudian implementasi, yang nantinya menghasilkan perancangan program aplikasi dan aplikasi berbasis Microsoft access yang dapat digunakan sebagai alternatif dalam memecahkan permasalahan pada PT. Hi-test dengan media elektronik dalam mengelola arsip secara efektif dan efisien
5. Berdasarkan penelitian (Endra, Rizal, and Ariani 2017) **“E-Arsip Berbasis *Image Archives Management Process Model* untuk Meningkatkan**

Efektifitas Pengelolaan Arsip”, permasalahan yang terjadi saat pengolahan arsip yakni tidak terkordinirnya arsip karena belum terstruktur dalam penyimpanan arsip, dalam menyelesaikan permasalahan ini peneliti menggunakan metode penelitian kuantitatif yakni penelitian yang memuat gambaran objek penelitian yang dijelaskan dari sudut pandang peneliti, nantinya penelitian ini menghasilkan inovasi dari sistem berbasis *web services* dalam pengembangan informasi pengarsipan yang terealisasi secara terstruktur agar dapat digunakan masyarakat manapun.

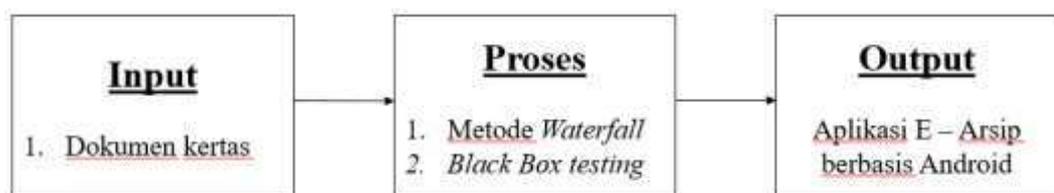
6. Menurut penelitian dari (Indrawati 2021) berjudul **“Efektivitas Pengarsipan Video dan Pembelajaran Online pada Siswa Keterampilan Belajar dan Inovasi”** Penelitian ini bertujuan untuk menganalisis hierarki hubungan dan pengaruh pengarsipan video media dan pembelajaran online untuk keterampilan dan inovasi belajar siswa. Belajar. masalah penelitian yang dapat dirumuskan adalah melakukan pengarsipan media *video* dan pembelajaran *online* secara simultan berpengaruh positif dan signifikan terhadap keterampilan belajar dan inovasi secara bersamaan ?, Apakah media *video* pengarsipan memiliki berpengaruh positif dan signifikan terhadap keterampilan belajar dan inovasi siswa?, tidak pembelajaran online berpengaruh positif dan signifikan terhadap hasil belajar siswa dan keterampilan inovasi?, metode yang digunakan adalah metode kuantitatif dengan alat bantu kuesioner. Analisis data termasuk analisis uji prasyarat yang meliputi validitas dan pengujian keandalan. Hasil penelitian menunjukkan bahwa: Menggunakan media video arsip dan online

Pembelajaran berpengaruh positif dan signifikan terhadap hasil belajar dan inovasi siswa Sementara itu, penggunaan media video arsip memiliki dampak positif dan Dampak signifikan pada pembelajaran siswa dan keterampilan inovasi, dan *online* Pembelajaran berpengaruh positif dan signifikan terhadap pembelajaran dan inovasi siswa Keahlian. Hasil ini menunjukkan bahwa penggunaan media *video* arsip dan online Pembelajaran telah terbukti mempengaruhi kemampuan siswa untuk belajar dan berinovasi, dengan demikian untuk dijadikan sebagai referensi penelitian selanjutnya.

2.4 Kerangka Pemikiran

Kerangka pemikiran merupakan penjelasan dari alur suatu penelitian dan juga menggambarkan secara garis besar bagaimana menyelesaikan penelitian tersebut.

Adapun tahapan dalam kerangka pemikiran penelitian ini yaitu:



Gambar 2.9 Desain Kerangka Pemikiran

Sumber: Data Olahan Peneliti (2022)

1. Input

Indikator dari penelitian ini yaitu dokumen kertas berdasarkan sifatnya yang berupa arsip penting, dokumen kertas dalam penelitian ini memiliki masalah yaitu:

- a. Penyimpanan bersifat manual berarti belum adanya penyimpanan digital yang terpusat dalam satu sistem dan juga penyimpanan yang masih menggunakan rak, lemari dan lain – lain sehingga dapat menimbulkan masalah dikemudian hari
- b. Dokumen rentan rusak yang diakibatkan banyak faktor seperti bencana alam, timbulnya jamur karena faktor perubahan cuaca dan masih banyak lagi
- c. Biaya pengelolaan yang semakin besar terkait ruang penyimpanan, biaya penyimpanan, pemeliharaan, tenaga pengelola atau faktor yang menyebabkan kerusakan dokumen.

2. Proses

Metode yang dilakukan dalam proses penelitian ini yaitu:

a. Metode *waterfall*

Dalam metode ini yang digunakan meliputi analisis kebutuhan sistem, desain *database* dan tampilan antarmuka, implementasi dengan bahasa pemrograman java, pengujian dengan *black box testing*, dan pemeliharaan dalam memperbaiki kinerja aplikasi yang dirancang.

b. *Black box testing*

Menguji di setiap fitur di dalam aplikasi, seperti ketika dimasukkan dokumen apakah proses input dan output sudah benar.

3. *Output*

Output/Hasil akhir dari sebuah penelitian ini akan dihasilkan aplikasi yang dapat menunjang pengelolaan arsip berbasis digital berbasis *mobile* dengan sistem operasi *Android*.