

## BAB II

### LANDASAN TEORI

#### 1.1. Tinjauan Studi

Dalam penulisan skripsi penulis menggali informasi dari penelitian sebelumnya sebagai bahan referensi. Baik mengenal permasalahan, metode dan hasil yang fungsinya memudahkan peneliti dalam melakukan penelitian sesuai dengan tema. Beberapa penelitian terkait sebagai berikut :

Penelitian yang diteliti oleh Widiana Mulyani yang berjudul “Pembangunan Sistem Informasi Data Balita Pada Posyandu Desa Ploso Kecamatan Punung Kabupaten Pacitan”, Dalam penelitian ini membahas mengenai Sistem Informasi Posyandu di Desa Ploso yang awalnya masih konvensional (Manual) yang perlu dikembangkan menjadi Sistem Komputerisasi. Tujuan dari penelitian ini yaitu untuk menghasilkan sistem informasi data bayi pada Posyandu Desa Ploso Kecamatan Punung Kabupaten Pacitan. Sistem yang digunakan dalam penelitian ini adalah Software Microsoft Visual Basic Net yang memiliki penanganan debug untuk mengetahui kesalahan kode yang terjadi secara cepat dan Microsoft Office Acces 2003 yang mampu menyimpan berbagai informasi untuk diolah dengan mudah dan cepat. Dengan adanya sistem informasi ini diharapkan mempermudah proses input, output data Posyandu Desa Ploso”[2].

Penelitian yang diteliti oleh Yusuf Ismail Nakhoda yang berjudul “Pemanfaatan Aplikasi Android Sebagai Sarana Penunjang Kegiatan Posyandu”, Membahas mengenai pemanfaatan aplikasi android untuk menunjang kegiatan Posyandu dengan menggunakan OpenMRS (sebuah rekam medis elektronik perusahaan) yang dikembangkan oleh D.Eridani, E.D.Widianto yang dilengkapi dengan pengaturan hak akses yang terdiri dari : Kader Posyandu, Petugas Gizi, Kepala Puskesmas dan Orang tua Balita. Dengan aplikasi tersebut mampu menunjang administrasi Posyandu dengan

mudah untuk membantu petugas Posyandu memasukkan data dan melihat data balita[3].

Pada penelitian yang diteliti oleh Yudi Wahyu Wibowo yang berjudul “Perancangan Sistem Informasi Posyandu Online”, Dalam penelitian ini membahas beberapa kasus seorang ibu rumah tangga yang lupa tidak tahu jadwal untuk melakukan pengecekan kesehatan terhadap anaknya di posyandu. Sehingga penelitian ini mengembangkan aplikasi informasi posyandu didalam sistem informasi android yang diharapkan menjadi jalan keluar bagi ibu yang sibuk atau lupa melakukan pengecekan kesehatan balitanya, dengan metode SDLC yang dirancang punarupa (Prototype) untuk sebuah aplikasi *open source* posyandu. Sehingga mempermudah ibu rumah tangga dalam mengakses sistem yang ada dengan perangkat yang sudah dimiliki berupa smartphone[1].

Penelitian yang dilakukan oleh Lailil Muflikhah yang berjudul “Peningkatan Manajemen Melalui Sistem Aplikasi Posyandu Di Kecamatan Lowokwaru Malang”, Menjelaskan tentang Pos Pelayanan Terpadu (Posyandu) merupakan upaya dari masyarakat dalam meningkatkan kesehatan masyarakat dengan memberikan layanan kesehatan dasar secara mudah dan ekonomis. Kurangnya partisipasi dari masyarakat akan pentingnya Posyandu mendorong dilakukannya kegiatan pengabdian masyarakat ini dengan membuat sistem aplikasi berbasis android. Sistem ini digunakan guna mencatat seluruh kegiatan posyandu dan menginformasikan tentang kegiatan informasi serta perkembangan balita. Beberapa tahap kegiatan diawali melalui survei kegiatan meliputi analisis kebutuhan sistem, rancangan dan hasil dari sistem, lalu tahap berikutnya ialah pelatihan yang diberikan kepada pengguna aplikasi, bagi kader posyandu dan ibu balita. Tahap akhir merupakan evaluasi terhadap tingkat kegunaan sistem bagi petugas dan peserta posyandu. Berdasarkan hasil kuisioner diperoleh bahwasannya sistem aplikasi tersebut telah memenuhi kebutuhan untuk dioperasikan secara efektif (83.67% responden), efisien (65% responden) dengan kemudahan (89.3% responden) serta 79% menyatakan tingkat kepuasan yang relatif tinggi[4]. Kelebihannya yaitu adanya fitur yang tersedia meliputi, update data dan notifikasi kegiatan posyandu. Kelemahannya informasinya tentang posyandu masih kurang.

Perbedaan dari keempat penelitian di atas dengan penelitian yang akan peneliti buat adalah dengan adanya sistem informasi posyandu akan mempermudah masyarakat terutama di Desa Jambu Timur untuk melihat informasi dari kegiatan posyandu.

## 1.2. Tinjauan Pustaka

### 1.2.1. Posyandu

Posyandu adalah salah satu bentuk dari Upaya Kesehatan Berbasis Masyarakat (UKBM) yang dijalankan dan dikelola dari, oleh, dan untuk masyarakat guna pembangunan kesehatan. Selain itu Posyandu dapat memberikan kemudahan bagi masyarakat untuk memperoleh pelayanan kesehatan dasar/sosial untuk menurunkan Angka Kematian Ibu dan Angka Kematian Bayi.

Berikut manfaat dari Posyandu sebagai berikut :

- a. Mendukung keadaan gizi dan kesehatan keluarga.
- b. Guna hidup bersih dan sehat.
- c. Mencegah tersebarnya penyakit di lingkungan dan penyakit yang dapat dicegah dengan imunisasi.
- d. Memperoleh Pelayanan Keluarga Berencana, sehingga Pasangan Usia Subur (PUS).
- e. Pemberdayaan keluarga dan masyarakat dalam penganekaragaman pangan melalui pemanfaatan pekarangan untuk memotivasi kelompok dasawisma berperan aktif.

Berikut pelayanan dari Posyandu Sebagai berikut[5] :

- a. Imunisasi .
- b. Ibu hamil.
- c. Pemberian vitamin setiap bulan.

### 1.2.2. Sistem Informasi

Sistem informasi dapat didefinisikan sebagai suatu sistem didalam sebuah organisasi dari kombinasi sumber daya manusia, sarana prasarana/fasilitas, sebuah teknologi, media, prosedur dan pengendalian untuk mendapat jalur komunikasi, proses transaksi secara rutin, pemberian sinyal pada manajemen dan SI menjadi sangat begitu penting dan tergantung kepada kepada lainnya terhadap peristiwa di dalam dan di luar yang menyajikan dasar informasi untuk menentukan keputusan yang baik.

Informasi diperoleh dari sistem informasi (*information sistem*) atau di sebut juga dengan processing sistem atau *information processing sistem*[6].



### 1.2.3. Android

Hermawan (2011 : 1), Android merupakan OS (*Operating System*) Mobile yang berkembang di tengah OS lainnya. OS lainnya seperti Windows Mobile, i-Phone OS, Symbian, dan masih banyak lagi. OS yang berjalan dengan mengutamakan aplikasi itu yang kembangkan sendiri tanpa memperhatikan potensi yang besar dari aplikasi pihak ketiga. Data asli ponsel dapat diperoleh dikarenakan adanya keterbatasan dari aplikasi pihak ketiga[7].

**Tabel 2.1 : Versi Android**

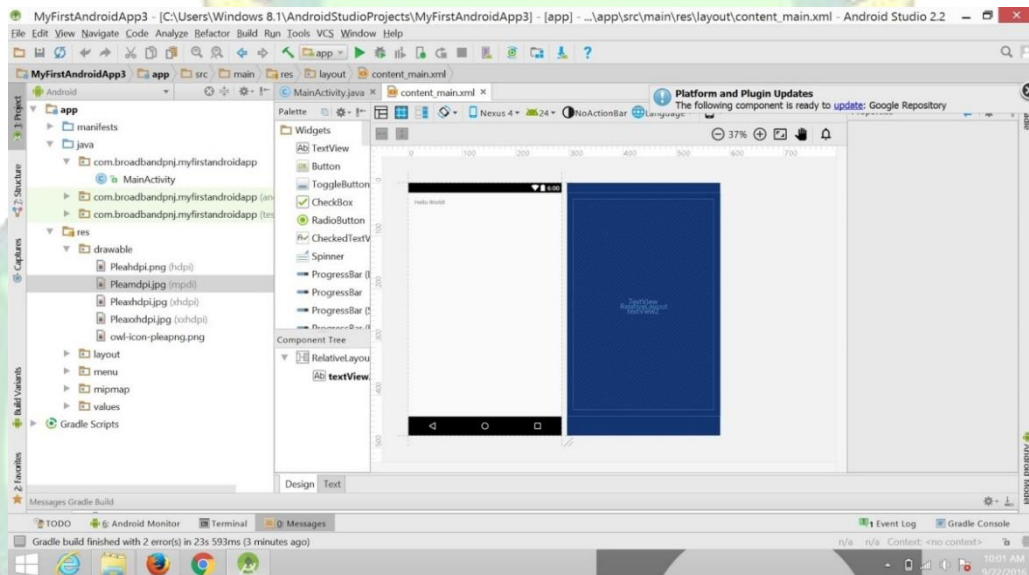
No	Nama Kode	API Level
1	Alpha	1
2	Beta	2
3	Cupcake	3
4	Donut	4
5	Éclair	5-7
6	Froyo	8
7	Gingerbread	9-10
8	Honeycomb	11-13
9	Ice Cream Sandwitch	14-15
10	Jelly Bean	16-18
11	Kitkat	19-20
12	Lolipop	21-22
13	Marshmallow	23
14	Nougat	24

### 1.2.4. Android Studio

Falker (2013), Android Studio adalah sebuah IDE dari Google yang diperkenalkan saat event Google I/O pada bulan Mei tahun 2013 dan merupakan IDE. Alternative selain IDE Eclipse. Dalam website resmi dikatakan bahwa Android Studio adalah IDE resmi untuk mengembangkan aplikasi android, yang berbasis intellij IDEA. Pada proses pembuatan

aplikasi, penulis menggunakan software Android Studio 2.3.3, Java Development Kit (JDK) versi 7, Standard Development Kit (SDK) versi 14[7]. Fitur-fitur lainnya adalah sebagai berikut:

- 1) Menggunakan Gradle-based build system yang lebih fleksibel.
- 2) Bisa mem-build banyak APK.
- 3) Template yang support untuk Google Service dan berbagai macam tipe perangkat.
- 4) Layout editor yang akan lebih bagus dibanding aplikasi lainnya.



**Gambar 2.1 : Tampilan Utama Android Studio**

### 1.2.5. Android SDK

Android sdk merupakan tools bagi para programmer yang ingin mengembangkan aplikasi berbasis google android. Android sdk mencakup seperangkat alat pengembangan yang komprehensif. Untuk saat ini disiapkan android sdk (software development kit) yang digunakan sebagai alat atau peralatan bantu dan api yang digunakan dalam mulai pengembangan aplikasi pada platform android dengan menggunakan suatu bahasa program java sebagai platform aplikasi, android menyediakan kemudahan kepada anda dalam membuat suatu aplikasi yang ingin dibuat dan layak digunakan, yang bukan bawahan dari Smartphone.

### 1.2.6. Kotlin

Kotlin sendiri termasuk bahasa pemrograman yang menggunakan pengetikan *statis* yang berjalan menggunakan mesin *virtual java* atau menggunakan kompiler LLVM yang dapat

dikompilasikan di dalam bentuk kode *java*. Mungkin sebagian *developer android* sudah mengetahui kotlin. Awalnya Kotlin dikembangkan oleh JetBrains, yang merupakan sebuah perusahaan yang berada di balik IntelliJ IDEA. JetBrains secara resmi merilis kotlin secara *opensource* dan kini perkembangannya semakin maju di tambah dengan bantuan/support dari google. Jadi koding untuk kotlin lebih sederhana dari pada java, kotlin banyak digunakan untuk membuat program android.

Kotlin sendiri sebagai sebuah bahasa pemrograman yang menjadi bahasa penyempurnaan dari Java. Perancangan bahasa digunakan untuk memenuhi industry bahasa *object-oriented* yang lebih baik dari *java*, tetapi tetap bisa dioperasikan sepenuhnya menggunakan kode *java* dan memungkinkan untuk dilakukan perpindahan *java* ke kotlin menggunakan infrastruktur *compiler LLVM*.

 <b>Kotlin</b>	VS	 <b>Java</b>
<pre>class MyKotlinClass {     val name = "Omar"     val surname = "Matello"     val example = "My name is \$name \$surname" }</pre>		<pre>class MyJavaClass {     final String getName() {         return "Omar";     }      final String getSurname() {         return "Matello";     }      final String getExample() {         return String.format("My name is %s %s",             getName(), getSurname());     } }</pre>

Gambar 2.2 Contoh Bahasa Pemrograman Kotlin Dan Java

Kotlin sendiri memiliki kelebihan dan kekurangan, kelebihanya adalah :

- a. *Consie*(ringkas) : kotlin dapat mengurangi tingkat kerumitan dari kode yang ketika menggunakan Bahasa java.
- b. *Safe*(aman) : kotlin mampu menjamin setiap hubungan kata yang di ketik tidak akan *error*.
- c. *Versatile*(serbaguna) : kotlin dapat digunakan dalam pembuatan aplikasi *web* maupun *mobile*.
- d. *Interoperable*(ringkas) : kotlin dapat digunakan untuk membaca kode lama atau *library* yang kita gunakan.



Sedangkan kekurangan dari kotlin dalam penggunaannya, Kotlin memiliki ukuran runtime yang dapat bertambah, meskipun tambahan untuk ukuran hanya 800kb tetapi itu sangat mempengaruhi para pengguna android dan memiliki tingkat kesulitan dalam kode awal yang menurut beberapa *developer* sulit untuk dibaca[8].

### 1.2.7. Firebase

*Firebase* merupakan layanan DBaaS mempunyai kepanjangan *Database Backend as a Service* didirikan oleh Andrew Lee dan James Tampilan pada tahun 2011 dan tahun 2014 sudah diambil alih oleh google. *Firebase* ini dapat menyimpan dan membuat update data secara otomatis ke banyak pengguna (Sinkronisasi Data). Beberapa kelebihan yang dimiliki *firebase* antara lain : Realtime backend, memakai JSON (bukan SQL), SDK tersedia untuk banyak platform, powerful API untuk menyimpan data melakukan sinkronisasi. Hal yang ini sangat membantu terutama bagi pengembang aplikasi *mobile hybrid* yang dimana memiliki kemampuan *front end development*, *front end developers* dapat menggunakan kode javascript yang dapat berintegrasi dengan *firebase* yang sudah tidak asing[7].

Beberapa fitur yang dimiliki *Firebase* :

1. *Google Analytic*

*Analytics* menyediakan data seputar pengguna pada aplikasi Android dan iOS agar dapat mengambil keputusan yang lebih baik tentang produk dan pengoptimalan pemasaran.

2. *Real-Time Database*

Menyimpan serta mensinkronkan data yang dimiliki pengguna dan perangkat secara *realtime* menggunakan database noSQL yang dihosting secara *cloud*.

3. *Authentication*

Untuk mengelola pengguna dengan cara yang mudah dan aman.

4. *Cloud Storage*

Menyimpan dan membagikan gambar, audio, video, atau konten lain secara mudah dengan penyimpanan objek yang andal, sederhana, dan hemat biaya.

5. *Hosting*

Mempermudah hosting web statis dengan fitur yang dibuat khusus untuk web aplikasi modern.

Tabel 2.2 : Perbandingan *Firestore* dan MySQL

NO	Firestore	MySQL
1.	Database bertipe NoSQL	Database bertipe RDBMS
2.	Bersifat <i>Realtime</i>	Tidak bersifat <i>Realtime</i>
3.	Database yang dibuat tidak perlu membuat kodingan di sisi server	Database yang dibuat harus membuat kodingan di sisi server
4.	Punya versi berbayar	Gratis sepenuhnya
5.	Kodingan yang dipakai Lebih rumit dalam mengolah database	Kodingan yang dipakai lebih mudah dalam mengolah database

### 1.2.8. Black Box Testing

*Black Box Testing* berfokus pada spesifikasi fungsional dari perangkat lunak. Tester dapat mendefinisikan kumpulan kondisi input dan melakukan pengujian pada spesifikasi fungsional program.

*Black Box Testing* bukanlah solusi alternatif dari *White Box Testing* tapi lebih merupakan pelengkap untuk menguji hal-hal yang tidak dicakup oleh *White Box Testing*. *Black Box Testing* cenderung untuk menemukan hal-hal berikut[9].

1. Fungsi yang tidak benar atau tidak ada.
2. Kesalahan antarmuka (*interface errors*).
3. Kesalahan pada struktur data dan akses basis data.
4. Kesalahan performansi (*performance errors*).
5. Kesalahan inisialisasi dan terminasi.

Pengujian dalam metode *Black Box Testing* memiliki tiga tahapan :

1. Pengujian *Graph Based testing*.

Dalam melakukan pengujian *black box* yang pertama memahami objek yang ada pada model perangkat lunak dan menentukan hubungan yang dimiliki antara objek-objek



tersebut. Pengujian model *graph* dilakukan terhadap perilaku *system*, sedangkan *graph based testing* menggambarkan *graph* yang mewakili hubungan antar objek pada modul sehingga setiap objek dan hubungannya dapat diuji.

2. Pengujian *Equivalence partitioning* (partisi ekuivalensi).

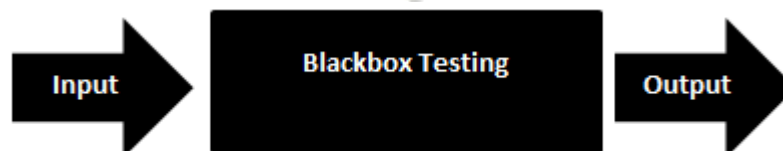
Dalam pengujian partisi ekuivalensi merupakan metode yang menguji pembagian domain input dari suatu program ke dalam kelas data, mengungkapkan kelas-kelas kesalahan setelah menentukan kasus pengujian, sehingga dapat mengurangi jumlah keseluruhan kasus pengujian.

3. Pengujian *Boundary value analysis* (analisis nilai batas)

Pada pengujian analisis nilai batas merupakan proses teknik desain yang melengkapi partisi ekuivalensi dengan berfokus pada domain output. Biasanya dipakai bila suatu kondisi input mengkhususkan sejumlah nilai, dilakukan pengujian dengan menggunakan jumlah minimum dan maksimum.

Kelebihan dari metode *black box testing* adalah :

1. Spesifikasi mengenai program dapat ditentukan diawal.
2. Mudah digunakan untuk menilai konsistensi program.
3. Pengujian akan dilakukan mengikuti spesifikasi.
4. Pengujian tidak akan melihat code program melainkan fungsi dari aplikasi berfungsi sesuai spesifikasi.



### Gambar 2.3 : Model pengujian *Blackbox*

Langkah –langkah dari metode *black box* :

1. Analisis kebutuhan dan spesifikasi
2. Pemilihan input
3. Pemilihan output
4. Pengujian
5. Review hasil
6. Evaluasi



Teknik Pengujian *Blackbox*

Tabel 2.3 : Pengujian Black Box Testing

No	Skenario Pengujian	Test Case	Hasil Yang Diharapkan	Hasil Pengujian	Status
1	Mengosongkan semua isian data login, lalu langsung mengklik tombol 'Masuk'.	Nama : - Kata sandi : -	Sistem akan menolak akses login dan menampilkan pesan "Mohon isi dulu nama admin dan kata sandi "	Sesuai harapan	Valid
2	Hanya mengisi data nama admin dan mengosongkan data kata sandi, lalu langsung mengklik tombol 'Masuk'.	Nama : admin Kata sandi : -	Sistem akan menolak akses login dan menampilkan pesan "Mohon isi dulu salah satu data yang masih kosong "	Sesuai harapan	Valid
3	Hanya mengisi data kata sandi dan mengosongkan data nama admin, lalu langsung mengklik tombol 'Masuk'.	Nama : - Kata sandi : 123	Sistem akan menolak akses login dan menampilkan pesan "Mohon isi dulu salah satu data yang masih kosong "	Sesuai harapan	Valid
4	Menginputkan dengan kondisi salah satu data benar dan satu lagi salah, lalu langsung mengklik tombol 'Masuk'.	Nama : (salah). Kata sandi : 123 (benar)	Sistem akan menolak akses login dan menampilkan pesan nama pengguna atau kata sandi salah ! Silahkan ulangi dengan data yang benar"	Sesuai harapan	Valid
5	Menginputkan Sintak SQL Injection pada kedua data login, lalu mengklik tombol 'Masuk'.	Nama : ' OR ' 1 ' = ' 1 Kata sandi : ' OR ' 1 ' = ' 1	Sistem akan menolak akses login dan menampilkan pesan nama pengguna atau kata sandi salah ! Silahkan ulangi dengan data yang benar"	Sesuai harapan	Valid
6	Menginputkan data login yang benar, lalu mengklik tombol 'masuk'.	Nama : admin Kata sandi : 123	Sistem menerima akses login dan kemudian langsung menampilkan form pakar/admin	Sesuai harapan	Valid

### 1.2.9 Metode *Rapid Application Development*

*Rapid Application Development* (RAD) adalah model proses pembangunan perangkat lunak yang tergolong dalam teknik *incremental* (bertingkah). RAD menekankan pada siklus pembangunan pendek, singkat dan cepat.

*Rapid Application Development* menggunakan metode literatif (berulang) dalam mengembangkan sistem *working model* (model kerja) sistem dikonstruksikan di awal tahap pengembangan dengan tujuan menetapkan kebutuhan (*requirement*) pengguna. Model kerja digunakan hanya sesekali saja sebagai basis desain dan implementasi sistem akhir.



Metode yang diusulkan adalah metode *Rapid Application Development*. Metode *Rapid Application Development* merupakan model proses pengembangan perangkat lunak sekuensial linier yang menekankan siklus pendekatan yang pendek. Metode RAD menggunakan pendekatan yang hampir sama dengan metode *Waterfall*, dapat dikatakan bahwa metode ini menggunakan tahap demi tahap dalam proses pembangunan sistem. Misalnya setelah mengetahui kebutuhan yang dibutuhkan sistem perangkat lunak. Dalam penelitian ini metode yang digunakan peneliti adalah metode *Rapid Application Development* yang sudah terstruktur dalam penerapan sistemnya yang meliputi *Requirements Planning* (Perencanaan Syarat-Syarat), *RAD Design Workshop* (Workshop Desain RAD), *Implementation* seperti gambar dibawah ini :



**Gambar 2.4 : Metode Rapid Application Development**

Dengan metode *Rapid Application Development* ini, nantinya akan mempermudah peneliti dalam membuat dan menerapkan sebuah informasi yang tertata secara terstruktur[10].

Beberapa kelebihan dari metode *Rapid Application Development* adalah :

- a. Proses pengiriman lebih mudah, hal ini di karenakan proses pembuatan lebih banyan menggunakan potongan-potongan script.
- b. Bisa mengurangi penulisan kode yang kompleks karena menggunakan wizard.
- c. Mampu meminimalkan kesalahan-kesalahan dengan menggunakan alat –alat bantuan (CASE tools).

- d. Lebih fleksibel karena pengembangan dapat melakukan proses desain ulang pada saat yang bersamaan.

Kekurang dari metode Rapid Application Development terdiri dari :

- a. Lebih banyak terjadi kesalahn apabilahanya mengutamakan kecepatan di bangdingkan dengan biaya dan kualitas.
- b. Ketelitian menjadi berkurang karena tidak menggunakan metode yang formal dalam melakukan pengkodean.
- c. Fasilitas-fasilitas banyak yang dikurangi karena terbatasnya waktu yang tersedia.
- d. Sistem sulit diaplikasikan di tempat yang lain.

Tahapan-tahapan dari metode Rapid Application Development yaitu :

### **1. *Requirements Planning (Perencanaan Syarat-Syarat)***

Tahap ini, pengguna dan penganalisis melakukan pertemuan untuk membahas tujuan dan syarat- syarat dari aplikasi atau sistem yang dihasilkan dari tujuan-tujuan posyandu. Hal penting dalam tahap ini adalah menyelesaikan masalah-masalah dari penelitian dan mencapai tujuan-tujuan dari posyandu.

### **2. *RAD Design Workshop (Workshop Desain RAD)***

Tahap ini, penganalisis dan pemrogram dapat bekerja membangun dan menunjukkan representasi visual desain dan pola kerja kepada pengguna posyandu. Workshop Desain dilakukan untuk merancang sebuah aplikasi atau sistem yang sesuai dengan kebutuhan pengguna posyandu. Untuk membangun aplikasi atau sistem berdasarkan hasil dari penelitian yang telah dilakukan sesuai kebutuhan posyandu.


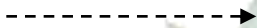




### **3. *Implementation (Implementasi)***

Tahap ini, penganalisis bekerja sama dengan para pihak posyandu secara berkala melalui pertemuan yang telah dijadwalkan. Setelah aplikasi atau sistem diuji coba dan disetujui oleh pihak posyandu lalu diperkenalkan kepada pengguna posyandu.


#### **1) *Diagram Use Case***

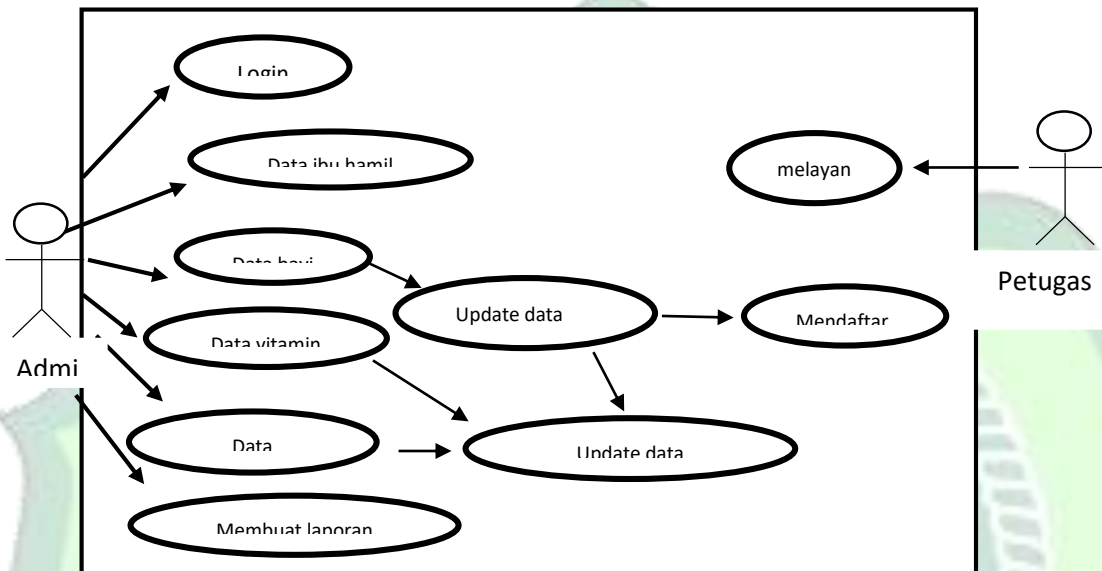
*Use case diagram* sebuah perilaku yang menggambarkan yang menggambarkan fungsi tertentu. Diagram *use case* bersifat statis. Diagram ini sangat penting untuk mengorganisir dan memodelan perilaku suatu sistem yang dibutuhkan serta diharapkan pengguna.

Tabel 2.4 : Tabel Use Case Diagram

No	Simbol	Nama	Keterangan
1		Aktor	Pengguna sistem yang berhubungan dengan sistem atau waktu.
2		<i>Include</i>	Memungkinkan 1 <i>Use case</i> menggunakan fungsionalitas yang disediakan oleh <i>use case</i> yang lain.
3		<i>Generalization</i>	Hubungan pada objek anak berbagi perilaku dan struktur data dari objek.
4		<i>Extend</i>	Menspesifikasi bahwa <i>use case</i> target memperluas perilaku yang dari <i>use case</i> sumber pada titik yang diberikan.
5		<i>Use case</i>	Deskripsi dari urutan aksi yang ditampilkan sistem.
6		<i>Collaboration</i>	Interaksi elemen lain yang bekerja sama untuk menyediakan perilaku yang besar dari jumlah yang ada



7		<i>Association</i>	Menghubungkan antara objek yang lain.
---	---	--------------------	---------------------------------------




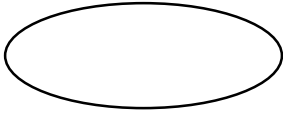



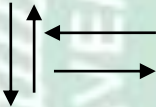
**Gambar 2.5 : Contoh Use Case**

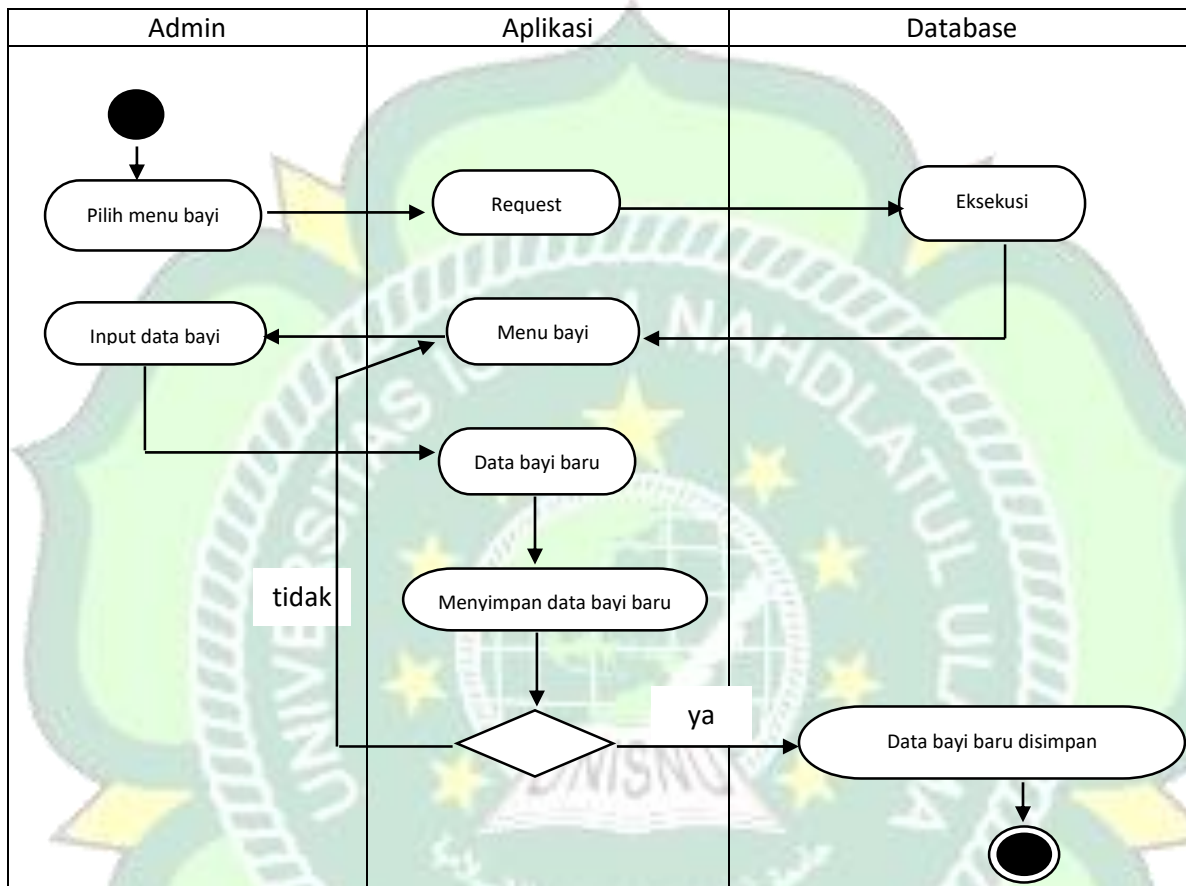
**2) Activity Diagram**

*Activity Diagram* merupakan diagram yang bersifat dinamis, diagram aktivitas adalah versi khusus dari diagram status yang menunjukkan aliran dari aktivitas menuju aktivitas lainnya pada sistem. Diagram ini penting dalam pemodelan dari suatu sistem dan memberikan tekanan pada aliran objek.

**Tabel 2.5 : Tabel Activity Diagram**

No	Simbol	Nama	Keterangan
1.		<i>Activity</i>	Memperlihatkan masing-masing class antarmuka saling berinteraksi satu sama lain.

2		<i>Action</i>	State dari sistem yang mencerminkan eksekusi dari suatu aksi.
3		<i>Intial Node</i>	Bagaimana objek dibentuk atau diawali.
4		<i>Activity Final Node</i>	Bagaimana objek dibentuk dan diakhiri.
5		<i>Decision</i>	Digunakan untuk menggambarkan suatu keputusan atau tindakan yang harus diambil pada kondisi tertentu.
6		<i>Line Connector</i>	Digunakan untuk menghubungkan satu simbol dengan yang simbol lain.




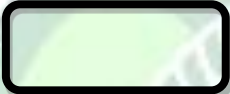


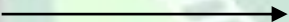
**Gambar 2.6 : Contoh Activity Diagram**

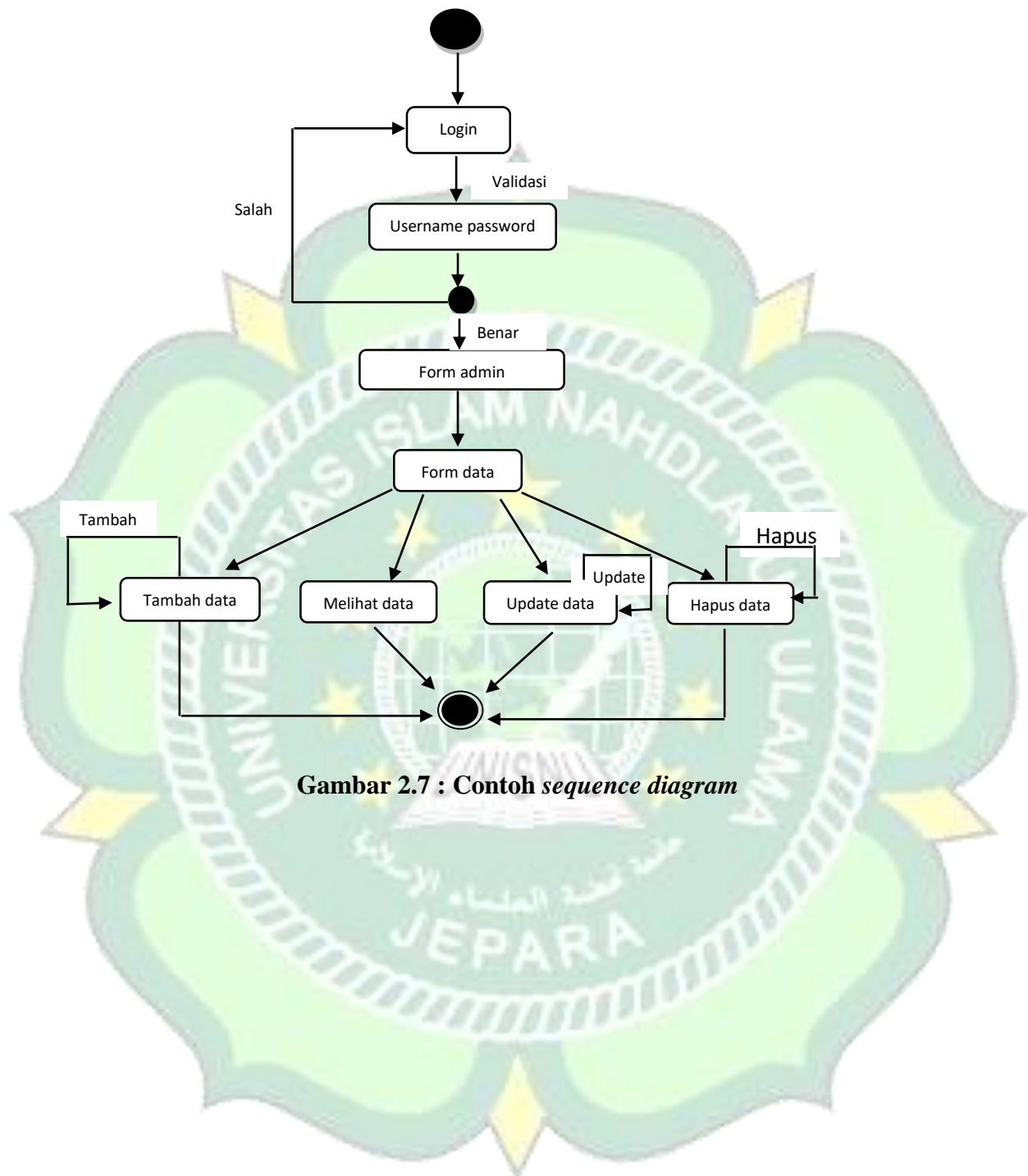
### 3) *Sequence Diagram*

*Sequence diagram* adalah sebuah diagram yang menjelaskan interaksi Objek yang disusun dalam suatu urutan waktu. Diagram ini secara khusus berasosiasi dalam *use case*. *Sequence diagram* memperlihatkan tahap demi tahap apa yang sebenarnya terjadi untuk menghasilkan sesuatu di dalam *use case*. Diagram *sequence* sebaiknya digunakan diawal tahap desain atau analisis karena kesederhanaan dan mudah untuk dimengerti[11].



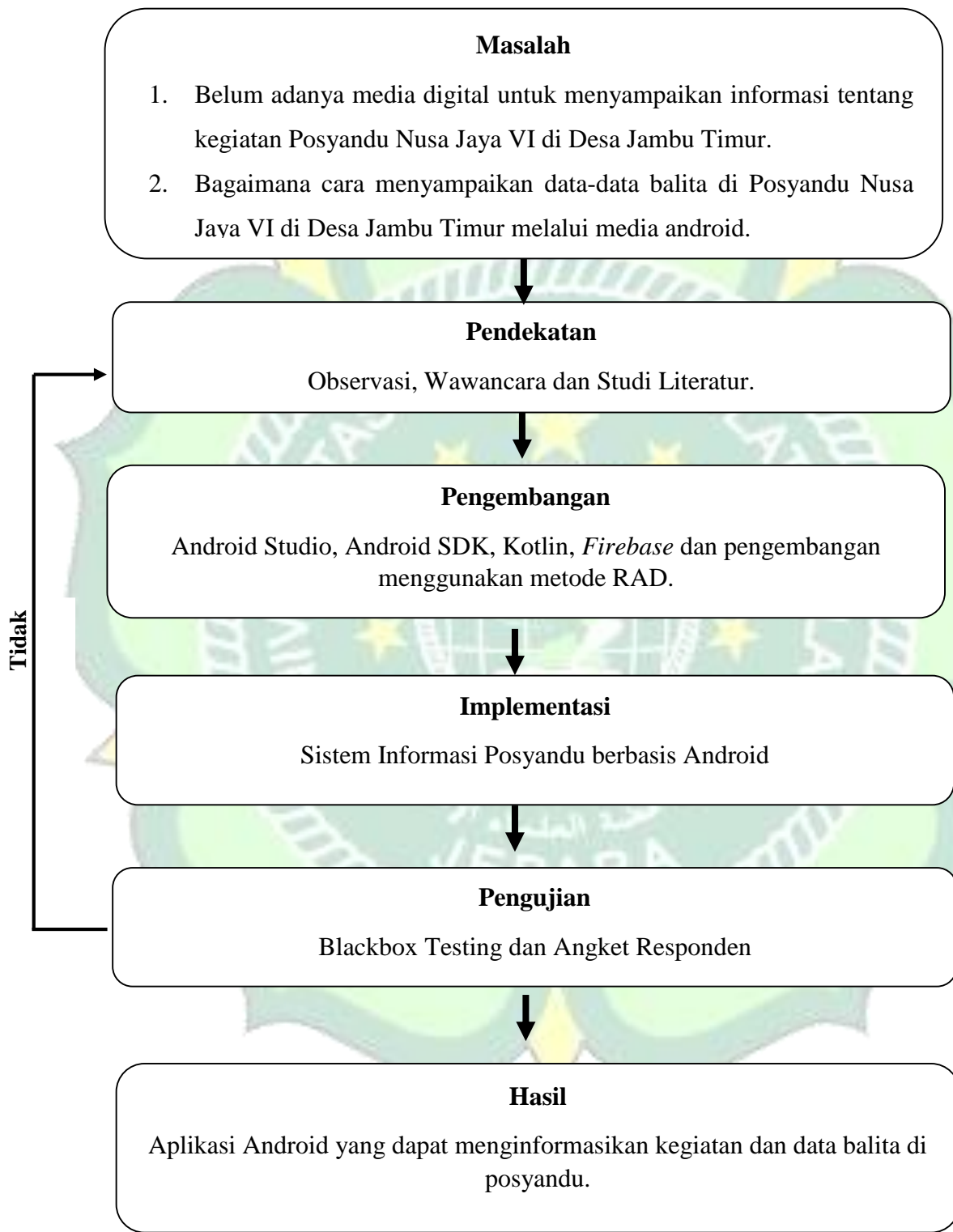
Tabel 2.6 : Simbol-simbol *Sequence diagram*

No	Simbol	Nama	keterangan
1.		<i>Start State</i>	Titik awal siklus hidup suatu elemen
2.		<i>State</i>	Menggunakan kondisi suatu elemen
3.		<i>Final State</i>	Titik akhir yang menjadi kondisi suatu elemen
4.		<i>Antion State</i>	Menggambarkan keadaan elemen dalam suatu aliran aktifitas
5.		<i>Flow Control</i>	Menggambarkan aliran aktifitas dari satu elemen ke elemen lain



**Gambar 2.7 : Contoh sequence diagram**

### 1.3. Kerangka Pemikiran



**Gambar 2.8 Kerangka Pemikiran**