

**APLIKASI ANDROID MANAJEMEN PASIEN
DI PRAKTIK MANDIRI BIDAN (PMB)**

SKRIPSI

**Diajukan sebagai salah satu syarat untuk memperoleh kelulusan
Jenjang Strata Satu (S1)
Pada Program Studi Sistem Informasi**

Oleh

Tupan Trisyanto

351742004



**SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER
INDONESIA MANDIRI
BANDUNG
2020**

LEMBAR PENGESAHAN
APLIKASI ANDROID MANAJEMEN PASIEN
DI PRAKTIK MANDIRI BIDAN (PMB)

Oleh
Tupan Trisyanto
351742004

Tugas Akhir ini telah diterima dan disahkan untuk
memenuhi persyaratan mencapai gelar

SARJANA KOMPUTER

Pada
PROGRAM STUDI SISTEM INFORMASI
SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER
INDONESIA MANDIRI

Bandung, 30 September 2020
Disahkan oleh



Ketua Program Studi,

Moch. Al Ramdhani, S.T., M.Kom
NIDN 0403097701

Dosen Pembimbing,

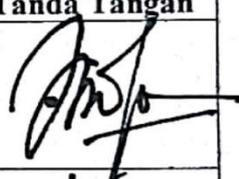
Hendra Gunawan, S.T., M.Kom
NIDN 0423037202

LEMBAR PERSETUJUAN REVISI
APLIKASI ANDROID MANAJEMEN PASIEN
DI PRAKTIK MANDIRI BIDAN (PMB)

Oleh
Tupan Trisyanto
351742004

Telah melakukan sidang tugas akhir dan telah melakukan revisi sesuai dengan perubahan dan perbaikan yang diminta pada saat sidang tugas akhir.

Bandung, Oktober 2020
Menyetujui

No	Nama Dosen	Keterangan	Tanda Tangan
1.	Hendra Gunawan, S.T., M.Kom	Pembimbing	
2.	Moch. Ali Ramdhani, S.T., M.Kom	Penguji 1	
3.	Novi Rukhviyanti, S.T., M.Si	Penguji 2	

SURAT PERNYATAAN

Dengan ini saya menyatakan bahwa:

- 1) Naskah Skripsi ini adalah asli dan belum pernah diajukan untuk mendapatkan gelar akademik, baik di Sekolah Tinggi Manajemen Informatika dan Komputer Indonesia Mandiri maupun perguruan tinggi lainnya.
- 2) Skripsi ini murni merupakan karya penelitian saya sendiri dan tidak menjiplak karya pihak lain. Dalam hal ada bantuan atau arahan dari pihak lain maka telah saya sebutkan identitas dan jenis bantuannya di dalam lembar ucapan terima kasih.
- 3) Seandainya ada karya pihak lain yang ternyata memiliki kemiripan dengan karya saya ini, maka hal ini adalah di luar pengetahuan saya dan terjadi tanpa kesengajaan dari pihak saya.

Pernyataan ini saya buat dengan sesungguhnya dan apabila di kemudian hari terbukti adanya kebohongan dalam pernyataan ini, maka saya bersedia menerima sanksi akademik sesuai norma yang berlaku di Sekolah Tinggi Manajemen Informatika dan Komputer Indonesia Mandiri.

Bandung, September 2020

Yang membuat pernyataan



Tupan Trisvanto
351742004

ABSTRAK

APLIKASI ANDROID MANAJEMEN PASIEN DI PRAKTIK MANDIRI BIDAN (PMB)

Oleh

Tupan Trisyanto

351742004

Kemudahan, kecepatan, dan keakuratan dari teknologi informasi semakin memudahkan pengguna dan meminimalisir tingkat kesalahan. Termasuk kemudahan pasien sebagai konsumen untuk dapat mengakses pelayanan kesehatan berupa pemeriksaan maupun pelayanan bidan tanpa harus melalui antrian. Tujuan pada penelitian ini yaitu membuat aplikasi manajemen pasien di praktik mandiri bidan yang berbasis Android untuk memudahkan pelayanan kepada pasien. Pada penelitian ini, penulis mencoba untuk membuat sebuah rancangan sistem teknologi informasi berbasis Android dengan memanfaatkan fitur GPS yang dapat memudahkan pasien dan calon pasien untuk melakukan pemesanan dan konsultasi kehamilan melalui *smartphone*. Sistem ini dapat mengeliminasi waktu antrian pasien baik dalam menunggu pemeriksaan bidan. Sistem ini juga dilengkapi dengan fitur tambahan berupa database rekam medis yang dapat diakses in time oleh pasien. Selain untuk pasien, fitur ini juga mampu memperkecil kemungkinan adanya human error pada saat proses entry data rekam medis yang semula dilakukan secara manual. Sistem ini menggunakan metode penelitian *waterfall* yaitu proses rancangan dimulai dengan analisis, desain, pengodean dan pengujian. Adapun pengumpulan data dilakukan melalui studi kepustakaan, studi lapangan, dan studi literature. Sistem teknologi informasi ini secara keseluruhan diharapkan mampu meningkatkan efektivitas dan efisiensi bagi pasien untuk mendapatkan pelayanan kesehatan maupun bagi petugas medis dalam memberikan pelayanan kesehatan.

Kata kunci: *bidan, smartphone, pasien*

ABSTRACT

ANDROID BASED PATIENT MANAGEMENT APPLICATION IN PRIVATE MATERNITY CLINIC

by

Tupan Trisyanto

351742004

Information and technology accessibility, accuracy and speed help its user to reduce errors. Access to healthcare service such as general health check or midwifery service without having to be physically waiting in line is one of patient accessibility. The purpose of this research is to build an android patient management application for private maternity clinic in giving access to its patient as healthcare service. In this research, the researcher was trying to design an android information and technology system using a GPS feature to give the patient and possible patient virtual queue and maternity consultation through a smartphone. This system can reduce queue time for the patient while waiting for their maternity check. This system also equipped with additional feature such as patient's medical record which can be accessed real-time by the patient. Not only for the patient, but this feature can reduce healthcare provider errors on medical record data entry, done manually before. This system using waterfall research method, a design process started with analysis, designing, coding, and testing. The data collection technique used literature review, field study and literature study. Overall, Information and technology system expected to improve effectivity and efficiency for the patient to get healthcare service and for the medical officer in giving healthcare service.

Keyword: midwife, patient, smartphone

KATA PENGANTAR

Bismillahirrahmanirrahim, Puji dan Syukur penulis panjatkan kepada Allah SWT karena atas rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan skripsi yang berjudul “**APLIKASI ANDROID MANAJEMEN PASIEN DI PRAKTIK MANDIRI BIDAN (PMB)**”. Skripsi ini disusun sebagai syarat untuk menyelesaikan jenjang pendidikan Strata Satu (S1) di Program Sistem Informasi Sekolah Tinggi Manajemen Informatika dan Komputer Indonesia Mandiri. Selama pembuatan skripsi ini, penulis banyak mendapatkan bantuan dan dorongan dari berbagai pihak, pada kesempatan kali ini penulis ingin menyampaikan penghargaan ucapan terima kasih sebesar-besarnya kepada:

1. Bapak Hendra Gunawan, S.T., M.Kom selaku pembimbing yang telah memberikan banyak saran dan arahan kepada penulis.
2. Bapak Dr. Chairuddin, M.M., M.T selaku Ketua Sekolah Tinggi Manajemen Informatika dan Komputer Indonesia Mandiri.
3. Bapak Moch. Ali Ramdhani, S.T., M.Kom selaku Ketua Prodi Sistem Informasi Sekolah Tinggi Manajemen Informatika dan Komputer Indonesia Mandiri.

Penulis menyadari dalam penulisan Skripsi ini, baik bentuk maupun isinya masih terdapat banyak kekurangan, oleh karena itu penulis mengharapkan kritik dan saran yang membangun untuk kemajuan penulis dikemudian hari. Penulis berharap skripsi ini bisa menjadi salah satu sumber yang dapat bermanfaat.

Bandung, September 2020
Penulis

DAFTAR ISI

SURAT PERNYATAAN	i
ABSTRAK	ii
ABSTRACT	iii
KATA PENGANTAR.....	iv
DAFTAR ISI.....	v
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang	1
1.2. Identifikasi Masalah	3
1.3. Tujuan Penelitian.....	3
1.4. Batasan Masalah.....	4
1.5. Metode Penelitian.....	4
1.6. Sistematika Penulisan.....	7
BAB II LANDASAN TEORI.....	9
2.1. Definisi Bidan	9
2.2. Kompetensi Bidan	11
2.3. Definisi Pasien.....	12
2.4. Metode <i>Waterfall</i>	13
2.5. Pemrograman Berorientasi Objek	14
2.6. <i>Unifed Modeling Language</i>	15
2.7. Basis Data (<i>Database</i>).....	25
2.8. MySQL (<i>Structured Query Language</i>)	26
2.9. <i>Entity Relationship Diagram</i>	27
2.10. <i>Black Box Testing</i>	32
2.11. <i>Android</i>	33
2.12. <i>Virtual Private Server</i>	41
2.13. <i>PHP</i>	42
2.14. <i>Framework</i>	43
2.15. <i>Laravel</i>	43

BAB III	ANALISA MASALAH DAN PERANCANGAN PROGRAM44
3.1.	Analisis.....	44
3.2.	Desain.....	50
BAB IV	IMPLEMENTASI DAN UJI COBA82
4.1.	<i>User Interface</i>	82
4.2.	Pengujian Sistem.....	90
BAB V	PENUTUP92
5.1.	Simpulan.....	92
5.2.	Saran.....	92
DAFTAR PUSTAKA	93

DAFTAR TABEL

Tabel 2.1.	Simbol-simbol <i>Use Diagram</i>	16
Tabel 2.2.	Simbol-Simbol <i>Activity Diagram</i>	17
Tabel 2.3.	Simbol-Simbol <i>Activity Diagram</i>	19
Tabel 2.4.	Simbol-simbol <i>Class Diagram</i>	21
Tabel 2.5.	Simbol-simbol <i>Statechart Diagram</i>	23
Tabel 2.6.	Simbol-simbol <i>Deployment Diagram</i>	25
Tabel 2.7.	Kardinalitas Pada ERD Versi James Martin.....	28
Tabel 2.8.	Simbol-Simbol <i>Flowchart</i>	29
Tabel 3.1.	Deskripsi Perancangan <i>Use Case</i>	51
Tabel 3.2.	Skenario <i>Use Case</i> Melihat Jadwal Bidan	53
Tabel 3.3.	Skenario <i>Use Case</i> membuat <i>Appointment</i> Bidan	53
Tabel 3.4.	Skenario <i>Use Case</i> List <i>Appointment</i> Bidan	53
Tabel 3.5.	Skenario <i>Use Case</i> Manage Data Pribadi	54
Tabel 3.6.	Skenario <i>Use Case</i> Registrasi	54
Tabel 3.7.	Skenario <i>Use Case</i> Login.....	54
Tabel 3.8.	Skenario <i>Use Case</i> Melihat Rekam Medis	55
Tabel 3.9.	Skenario <i>Use Case</i> Manage Rekam Medis	55
Tabel 3.10.	Skenario <i>Use Case</i> Registrasi Pasien.....	55
Tabel 3.11.	Skenario <i>Use Case</i> Mangae Jadwal.....	55
Tabel 3.12.	Skenario <i>Use Case</i> Mangae Klinik.....	56
Tabel 4.1.	Test ID A. 001 Halaman <i>User</i>	90
Tabel 4.2.	Test ID A. 002 Halaman Admin	90

DAFTAR GAMBAR

Gambar 1.1.	Ilustrasi Model <i>Waterfall</i>	5
Gambar 2.1.	Metode <i>Waterfall</i>	13
Gambar 2.2.	Contoh <i>Use Case Diagram</i>	17
Gambar 2.3.	Contoh <i>ActivityDiagram</i>	19
Gambar 2.4.	Contoh <i>Sequence Diagram</i>	21
Gambar 2.5.	Contoh <i>Class Diagram</i>	23
Gambar 2.6.	Contoh <i>Statechart Diagram</i>	24
Gambar 2.7.	Contoh <i>Deployment Diagram</i>	25
Gambar 2.8.	Arsitektur Android	40
Gambar 3.1.	<i>Flowchart System Appointment</i> Bidan yang Sedang Berjalan.....	45
Gambar 3.2.	<i>Flowchart</i> Imunisasi yang Sedang Berjalan	47
Gambar 3.3.	<i>Flowchart</i> Rekap Medis yang Sedang Berjalan.....	48
Gambar 3.4.	<i>Use Case Diagram</i> Aplikasi <i>Appointmnet</i> Bidan.....	52
Gambar 3.5.	<i>Activity Diagram</i> Mencari Bidan	57
Gambar 3.6.	<i>Activity Diagram</i> Membuat <i>Appointmnet</i> Bidan	58
Gambar 3.7.	<i>Activity Diagram</i> <i>List Appointmnet</i>	59
Gambar 3.8.	<i>Activity Diagram</i> Melihat Data Pasien	59
Gambar 3.9.	<i>Activity Diagram</i> <i>Manage</i> Data Diri	70
Gambar 3.10.	<i>Activity Diagram</i> Register Pasien	61
Gambar 3.11.	<i>Activity Diagram</i> <i>Login</i>	62
Gambar 3.12.	<i>Activity Diagram</i> Melihat Daftar Rekap Medis	62
Gambar 3.13.	<i>Activity Diagram</i> Menambah Data Rekam Medis	63
Gambar 3.14.	<i>Activity Diagram</i> <i>Manage</i> Jadwal	64
Gambar 3.15.	<i>Activity Diagram</i> Tambah Data Jadwal	65
Gambar 3.16.	<i>Activity Diagram</i> Ubah Data Jadwal	66
Gambar 3.17.	<i>Activity Diagram</i> Hapus Data Jadwal	67
Gambar 3.18.	<i>Sequence Diagram</i> Mencari Bidan	68
Gambar 3.19.	<i>Sequence Diagram</i> Melihat Daftar <i>Appointment</i>	69
Gambar 3.20.	<i>Sequence Diagram</i> <i>Setting</i> Data Pasien	70

Gambar 3.21. <i>Sequence Diagram Register</i>	70
Gambar 3.22. <i>Sequence Diagram Login</i>	71
Gambar 3.23. <i>Sequence Diagram</i> Melihat Rekap Medis.....	71
Gambar 3.24. <i>Sequence Diagram</i> Membuat Rekap Medis.....	72
Gambar 3.25. <i>Sequence Diagram</i> Manage Jadwal.....	73
Gambar 3.26. <i>Statechart Diagram</i> Melihat List Appointment	73
Gambar 3.27. <i>Statechart Diagram</i> Setting Data Pasien.....	74
Gambar 3.28. <i>Statechart Diagram Register</i>	74
Gambar 3.29. <i>Statechart Diagram Login</i>	75
Gambar 3.30. <i>Statechart Diagram</i> Tambah Data Jadwal	75
Gambar 3.31. <i>Statechart Diagram</i> Ubah Data Jadwal	75
Gambar 3.32. <i>Statechart Diagram</i> Hapus Data Jadwal	76
Gambar 3.33. Perancangan <i>Interface Login</i> Pasien	77
Gambar 3.34. Perancangan <i>Register</i> Pasien	78
Gambar 3.35. Perancangan <i>Dashboard</i>	79
Gambar 3.36. <i>List Order</i>	80
Gambar 3.37. Perancangan <i>Buat Appointment</i>	81
Gambar 4.1. <i>Login</i> Bidan.....	82
Gambar 4.2. Halaman <i>Slider</i>	83
Gambar 4.3. Halaman <i>Berita</i>	83
Gambar 4.4. Halaman <i>User/Pasien</i>	84
Gambar 4.5. Halaman <i>Order</i> Pasien	84
Gambar 4.6. Halaman <i>Login</i> Pasien	85
Gambar 4.7. Halaman <i>Register</i> Pasien.....	86
Gambar 4.8. Halaman <i>Dashboard</i>	87
Gambar 4.9. Halaman <i>Register</i>	88
Gambar 4.10. Halaman <i>List Order</i>	89

BAB I

PENDAHULUAN

1.1. Latar Belakang

Perkembangan dan pemanfaatan teknologi informasi dan komunikasi (TIK) saat ini sangat pesat, tidak terkecuali bidang layanan kesehatan. Saat ini semakin banyak alat untuk menyampaikan dan menerima sebuah informasi, salah satunya dengan memanfaatkan *smartphone*. Salah satu *mobile device* yang banyak digunakan saat ini adalah *smartphone* berbasis android. Android adalah sistem operasi yang berbasis Linux untuk telepon seluler seperti *smartphone* dan komputer tablet. Android banyak digunakan karena menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh berbagai macam peranti bergerak. Pada perkembangannya android merupakan subset perangkat lunak yang meliputi sistem operasi, *middleware*, dan aplikasi kunci yang dirilis oleh *google*. (Marjuni et al., 2017)

Praktik Mandiri Bidan (PMB) merupakan bentuk pelayanan kesehatan di bidang kesehatan dasar. Praktik bidan adalah serangkaian kegiatan pelayanan kesehatan yang diberikan oleh bidan kepada pasien (individu, keluarga, dan masyarakat) sesuai dengan kewenangan dan kemampuannya. Bidan yang menjalankan praktik harus memiliki Surat Izin Praktik Bidan (SIPB) sehingga dapat menjalankan praktik pada saran kesehatan atau program (*PERATURAN MENTERI KESEHATAN REPUBLIK INDONESIA*, n.d.).

Praktik pelayanan bidan mandiri merupakan penyedia layanan kesehatan, yang memiliki kontribusi cukup besar dalam memberikan pelayanan, khususnya dalam meningkatkan kesejahteraan ibu dan anak. Supaya masyarakat pengguna jasa layanan bidan memperoleh akses pelayanan yang bermutu, perlu adanya regulasi pelayanan praktik bidan secara jelas persiapan sebelum bidan melaksanakan pelayanan praktik seperti perizinan, tempat, ruangan, peralatan praktik, dan kelengkapan administrasi semuanya harus sesuai dengan standar.

Selama ini Praktik Mandiri Bidan (PMB) belum menerapkan manajemen dengan menggunakan aplikasi untuk memudahkan dalam memberikan pelayanan kepada pasien. Walaupun pasien PMB tidak sebanyak di Klinik maupun di Rumah Sakit (RS), akan tetapi apabila manajemen pelayanan menggunakan sistem informasi online maka pasien tidak perlu mengantri atau kecewa karena Bidan sedang tidak ada di tempat, dan memberi kemudahan kepada pasien untuk mengetahui jadwal kunjungna berikutnya karena pada sistem ini akan dibuat pengingat untuk jadwal kunjungan berikutnya. Penelitian oleh Laeliah (2017), Torry (2016), dan Bustani (2015) menyatakan bahwa waktu tunggu antrian pada sistem konvensional sangat berpengaruh pada tingkat kepuasan pelayanan pasien. Oleh karena itu, dibuatlah sebuah rancangan dan implementasi dari suatu sistem antrian yang membantu antrian untuk menjadi terorganisir dan menyenangkan dan dapat diakses dari *mobile phone*. (Laeliah & Subekti, 2017) (Torry et al., 2016) (Bustani et al., 2015).

Beberapa penelitian menggunakan teknologi pada *mobile phone* sebagai media pendaftaran online, ada yang menggunakan media SMS (*short message*

service) untuk manajemen antrian (Daimi et al., 2018), ada pula yang menggunakan teknologi *Near Field Communication* atau NFC untuk *check in* antrian (Symey et al., 2013). Sebagian besar menggunakan aplikasi dengan *platform* android karena tergolong mudah dalam pengembangan.

Sistem yang akan dibangun adalah sistem yang dapat memudahkan pasien atau calon pasien untuk berhubungan langsung dengan Bidan melalui aplikasi yang akan berfungsi untuk pendaftaran kunjungan tanpa harus mengantri, pengingat kapan jadwal kunjungan berikutnya, maupun untuk pendaftaran pasien baru. Berkas rekam medis akan langsung bisa diketahui di database dan pasien tidak perlu menunggu berkasnya dicarikan terlebih dahulu. Dilihat dari kondisi tersebut dibutuhkan sebuah sistem informasi yang dapat membantu pelayanan kepada pasien sehingga dapat berjalan dengan efisien. Berdasarkan hal tersebut peneliti memilih judul “**APLIKASI ANDROID MANAJEMEN PASIEN DI PRAKTIK MANDIRI BIDAN (PMB)**”.

1.2. Identifikasi Masalah

Sesuai dengan latar belakang yang telah diuraikan sebelumnya, identifikasi masalah dari penelitian ini adalah: Belum adanya aplikasi manajemen pasien di bidan praktik mandiri bidan yang berbasis Android

1.3. Tujuan Penelitian

Sesuai dengan identifikasi masalah yang telah diuraikan sebelumnya, maka tujuan penelitian ini adalah: Membuat aplikasi manajemen pasien di praktik mandiri bidan yang berbasis Android untuk memudahkan pelayanan kepada pasien.

1.4. Batasan Masalah

Di dalam melakukan suatu penelitian di perlukan adanya batasan suatu batasan masalah supaya penelitian tersebut lebih terarah dan memudahkan dalam pembahasan sehingga tujuan penelitian akan tercapai. Beberapa batasan masalah dalam penelitian ini adalah sebagai berikut:

1. Aplikasi ini mencatat rekam medis pasien
2. Penggunaan aplikasi ini hanya bidan dan pasien
3. Penelitian ini dilakukan hanya sampai dengan tahap pengujian.

1.5. Metode Penelitian

1.5.1. Metode Pengumpulan Data

Adapun metode pengumpulan data yang dilakukan untuk mendapatkan data-data dan referensi yang dibutuhkan dalam menyusun penelitian ini, meliputi:

1. Studi Kepustakaan

Dilakukan dengan cara membaca dan mempelajari buku-buku, referensi yang mendukung dengan topik yang akan dibahas dalam penyusunan penelitian ini, selain itu penulis juga mengumpulkan data dari situs-situs internet yang berhubungan dengan penelitian ini.

2. Studi Lapangan

- a. Observasi

Melakukan observasi ke klinik bidan untuk mendapatkan data-data yang diperlukan.

b. Wawancara

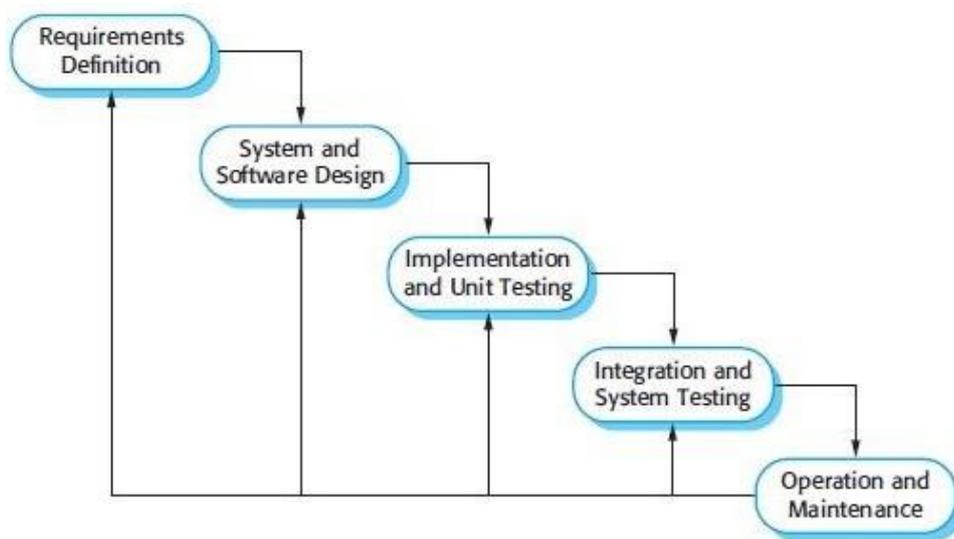
Melakukan wawancara kepada pihak klinik untuk mendapatkan data yang diperlukan dalam penelitian.

3. Studi Literatur

Penelitian melakukan suatu kajian dari hasil penelitian yang sudah ada sebagai suatu perbandingan dan sumber referensi dalam penelitian.

1.5.2. Metode Pengembangan Sistem

Kerangka berpikir dalam penelitian ini menggunakan metode pengembangan (SDLC) dengan Model *Waterfall*. Model ini melakukan pendekatan secara sistematis dan berurutan. Metode *Waterfall* mempunyai tahapan-tahapan yang digambarkan pada Gambar 1.1.



Gambar 1.1 Ilustrasi *Model Waterfall* (Ian Sommerville, 2011)

Berikut ini adalah penjelasan dari tahapan-tahapan yang dilakukan di dalam

Model *Waterfall*:

1. *Requirements analysis and definition*

Layanan sistem, kendala, dan tujuan ditetapkan oleh hasil konsultasi dengan pengguna yang kemudian didefinisikan secara rinci dan berfungsi sebagai spesifikasi sistem.

2. *System and software design*

Tahapan perancangan sistem mengalokasikan kebutuhan-kebutuhan sistem baik perangkat keras maupun perangkat lunak dengan membentuk arsitektur sistem secara keseluruhan. Perancangan perangkat lunak melibatkan identifikasi dan penggambaran abstraksi sistem dasar perangkat lunak dan hubungannya.

3. *Implementation and unit testing*

Pada tahap ini, perancangan perangkat lunak direalisasikan sebagai serangkaian program atau unit program. Pengujian melibatkan verifikasi bahwa setiap unit memenuhi spesifikasinya.

4. *Integration and system testing*

Unit-unit individu program atau program digabung dan diuji sebagai sebuah sistem lengkap untuk memastikan apakah sesuai dengan kebutuhan perangkat lunak atau tidak. Setelah pengujian, perangkat lunak dapat dikirimkan ke *customer*.

5. *Operation and maintenance*

Biasanya (walaupun tidak selalu), tahapan ini merupakan tahapan yang paling panjang. Sistem dipasang dan digunakan secara nyata. *Maintenance*

melibatkan pembetulan kesalahan yang tidak ditemukan pada tahap-tahapan sebelumnya, meningkatkan implementasi dari unit sistem, dan meningkatkan layanan sistem sebagai kebutuhan baru.

1.6. Sistematika Penulisan

Sistematika penulisan tugas akhir ini disusun untuk memberikan gambaran umum tentang penelitian yang dijalankan. Sistematika penulisan tugas akhir ini adalah sebagai berikut:

BAB I PENDAHULUAN

Bab ini menguraikan tentang latar belakang permasalahan, mencoba merumuskan inti permasalahan yang dihadapi, menentukan tujuan dan kegunaan penelitian, yang kemudian diikuti dengan pembatasan masalah, asumsi, serta sistematika penulisan.

BAB II LANDASAN TEORI

Bab ini membahas berbagai konsep dasar dan teori-teori yang berkaitan dengan topik penelitian yang dilakukan dan hal-hal yang berguna dalam proses analisis permasalahan serta tinjauan terhadap penelitian-penelitian serupa yang telah pernah dilakukan sebelumnya termasuk sintesisnya. Terdapat landasan teori yang mendasari pembangunan sistem informasi seperti pengertian sistem, informasi, sistem informasi pemodelan data, analisis sistem, kamus data, pengolahan data, dan *tools – tools* yang akan digunakan.

BAB III ANALISA MASALAH DAN PERANCANGAN PROGRAM

Bab ini membahas tentang analisis kebutuhan dalam membangun perangkat lunak, analisis sistem yang akan berjalan sesuai dengan metode pembangunan perangkat lunak yang digunakan. Terdapat pula perancangan antarmuka untuk aplikasi yang

akan dibangun sesuai dengan hasil analisis yang telah dibuat. Model dalam perancangan yang akan digunakan adalah pemodelan terstruktur.

BAB IV IMPLEMENTASI DAN UJI COBA

Bab ini juga membahas hasil implementasi dari hasil analisis dan perancangan yang telah dibuat disertai juga dengan hasil pengujian sehingga diketahui apakah sistem yang dibangun sudah memenuhi syarat sebagai aplikasi yang mudah digunakan.

BAB V PENUTUP

Bab ini berisi kesimpulan dan saran yang diperoleh dari hasil penelitian skripsi yang telah dibuat. Kesimpulan didapatkan dari penelitian yang telah dilakukan sampai pembuatan laporan serta saran agar manajemen pasien di praktik mandiri bidan nyeri susanti bisa lebih baik lagi, dan untuk selanjutnya bisa di pakai sebagai aplikasi pada bidan-bidan lainnya.

BAB II

LANDASAN TEORI

2.1. Definisi Bidan

Bidan adalah seorang perempuan yang lulus dari pendidikan Bidan yang diakui pemerintah dan organisasi profesi di wilayah Negara Republik Indonesia serta memiliki kompetensi dan kualifikasi untuk diregister, sertifikasi dan atau secara sah mendapat lisensi untuk menjalankan praktik kebidanan (Ikatan Bidan Indonesia, 2016).

Bidan adalah tenaga profesional yang bertanggung-jawab dan akuntabel, yang bekerja sebagai mitra perempuan untuk memberikan dukungan, asuhan dan nasehat selama masa hamil, masa persalinan dan masa nifas, memfasilitasi dan memimpin persalinan atas tanggung jawab sendiri dan memberikan asuhan kepada bayi baru lahir, dan bayi. Asuhan ini mencakup upaya pencegahan, promosi persalinan normal, deteksi komplikasi pada ibu dan anak, dan akses bantuan medis atau bantuan lain yang sesuai, serta melaksanakan tindakan kegawat-daruratan. Bidan mempunyai tugas penting dalam konseling dan pendidikan kesehatan, tidak hanya kepada perempuan, tetapi juga kepada keluarga dan masyarakat. Kegiatan ini mencakup pendidikan antenatal dan persiapan menjadi orang tua serta dapat meluas pada kesehatan perempuan, kesehatan seksual atau kesehatan reproduksi dan asuhan anak. Bidan dapat praktik diberbagai tatanan pelayanan: termasuk di rumah, masyarakat, Rumah Sakit, klinik atau unit kesehatan lainnya. (Ikatan Bidan Indonesia, 2016)

Definisi bidan menurut Ikatan Bidan Indonesia atau IBI (2006) adalah seorang wanita yang telah mengikuti dan menyelesaikan pendidikan yang telah diakui pemerintah dan lulus ujian sesuai dengan persyaratan yang berlaku dan diberi izin secara sah untuk melaksanakan praktik. Dalam melaksanakan pelayanan kesehatan dan kebidanan di masyarakat, bidan diberi wewenang oleh pemerintah sesuai dengan wilayah pelayanan yang diberikan. Wewenang tersebut berdasarkan peraturan Menkes RI.Nomor 900/Menkes ISK/VII/2002 tentang registrasi dan praktek bidan. (Kemenkes RI, 2016)

Federation of International Gynaecologist and Obstetritian atau FIGO (1991) dan *World Health Organization* atau WHO (1992) mendefinisikan bidan adalah seseorang yang telah menyelesaikan program pendidikan bidan yang diakui oleh negara serta memperoleh kualifikasi dan diberi izin untuk menjalankan praktik kebidanan di Negeri itu. Dia harus mampu memberikan supervisi, asuhan dan memberikan nasehat yang dibutuhkan kepada wanita selama masa hamil, persalinan dan masa pasca persalinan, memimpin persalinan atas tanggung jawabnya sendiri serta asuhan pada bayi baru lahir dan anak. (Kemenkes RI, 2016)

Bidan mempunyai tugas penting dalam konsultasi dan pendidikan kesehatan baik bagi wanita sebagai pusat keluarga maupun masyarakat pada umumnya, tugas ini meliputi antenatal, intranatal, postnatal, asuhan bayi baru lahir, persiapan menjadi orangtua, gangguan kehamilan dan reproduksi serta keluarga keluarga berencana. Bidan juga dapat melakukan praktik kebidanan pada Puskesmas, Rumah sakit, klinik bersalin dan unit-unit kesehatan lainnya di masyarakat. (Undang-Undang RI No 4, 2019)

Menurut Estiwidani.D, dkk (2008) peran, fungsi bidan dalam pelayanan kebidanan adalah sebagai: pelaksana, pengelola, pendidik, dan peneliti. Sedangkan tanggung jawab bidan meliputi pelayanan konseling, pelayanan kebidanan normal, pelayanan kebidanan abnormal, pelayanan kebidanan pada anak, pelayanan KB, dan pelayanan kesehatan masyarakat. Sedemikian kompleksnya peran, fungsi, dan tanggung jawab seorang bidan dalam melaksanakan tugasnya memberikan pelayanan kebidanan yang terbaik dan professional kepada masyarakat maka untuk keberhasilan dalam mencapai tujuan tersebut diperlukan landasan yang kuat berupa kompetensi bidan. [11]

2.2. Kompetensi Bidan

Kompetensi merupakan pengetahuan, keterampilan dan nilai-nilai dasar yang direfleksikan dalam kebiasaan berfikir dan bertindak secara konsisten dan terus menerus yang memungkinkan seseorang menjadi kompeten, dalam arti memiliki pengetahuan, keterampilan, nilai-nilai dan sikap dasar untuk melakukan sesuatu. Kebiasaan berfikir dan bertindak itu didasari oleh budi pekerti luhur baik dalam kehidupan pribadi, sosial, kemasyarakatan, keberagamaan, dan kehidupan berbangsa dan bernegara. Kompetensi dibagi atas 2 kategori, yaitu (1) Kompetensi Inti atau Dasar; yaitu kompetensi minimal yang mutlak dimiliki oleh pelaku profesi, dan (2) Kompetensi Tambahan atau Lanjutan; yaitu pengembangan dari pengetahuan dan keterampilan dasar untuk mendukung tugas suatu profesi dalam memenuhi tuntutan/kebutuhan masyarakat yang sangat dinamis serta perkembangan IPTEK. (*PERATURAN MENTERI KESEHATAN REPUBLIK INDONESIA, n.d.*)

Komptensi inti bidan sering bervariasi menurut pedoman-pedoman nasional dan regional, kode etik profesi, adat istiadat dan keyakinan kultural, mutu pendidikan pelatihan kebidanan, serta kerjasama dari tim kesehatan. Bidan harus memiliki kompetensi dan bidang pengetahuan, keterampilan dan perilaku dalam melaksanakan praktik kebidanan secara aman dan bertanggungjawab dalam berbagai tatanan pelayanan kesehatan. Kompetensi bidan dikelompokkan dalam dua kategori yaitu inti/dasar dan kompetensi tambahan/lanjutan. Kompetensi inti merupakan kompetensi minimal yang mutlak dimiliki oleh bidan. Kompetensi lanjutan merupakan pengembangan dari pengetahuan dan keterampilan dasar untuk mendukung tugas bidan dalam memenuhi tuntutan dan kebutuhan masyarakat serta perkembangan ilmu pengetahuan dan teknologi. (Undang-Undang RI No 4, 2019)

Lahirnya kompetensi bidan di Indonesia tidak terlepas dari Permenkes 572 Tahun 1996 tentang Registrasi Praktik Bidan, kompetensi bidan yang disusun oleh ICM pada Februari 1999, kompetensi bidan Indonesia yang disahkan pada KONAS IBI XII di Denpasar Bali, Peraturan Kepmenkes RI No. 900/Menkes/SK/II/2002 tentang kewenangan praktik bidan dan Keputusan Menteri Kesehatan Republik Indonesia Nomor: 369/Menkes/SK/III/2007 Tentang Standar Profesi Bidan.(Kemenkes RI, 2016)

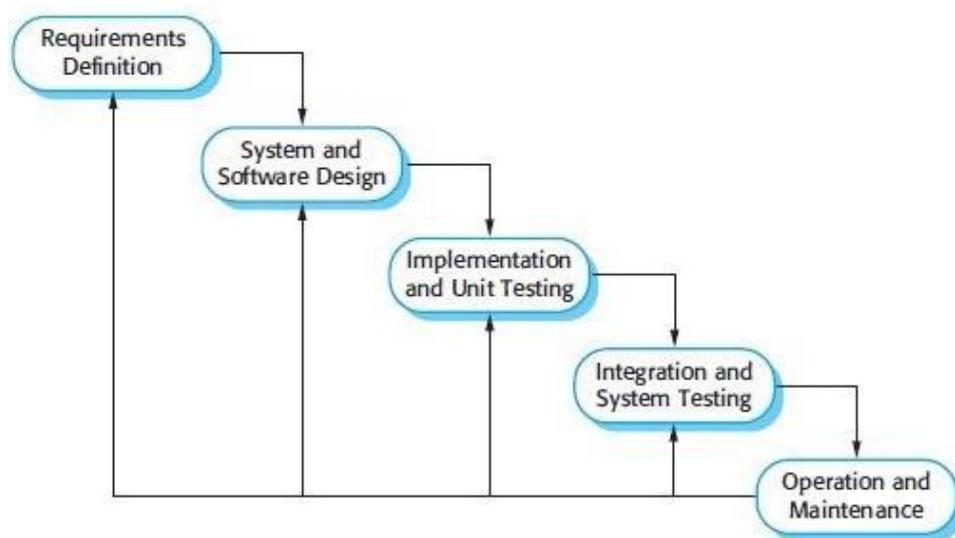
2.3. Definisi Pasien

Pasien adalah setiap orang yang melakukan konsultasi masalah kesehatannya untuk memperoleh pelayanan kesehatan yang diperlukan, baik secara langsung maupun tidak langsung di Rumah Sakit. Pasien merupakan pelanggan layanan kesehatan, tetapi pasien dalam hal ini hanya merupakan salah satu jenis

pelanggan. Pelanggan layanan kesehatan merupakan semua orang yang sehari-harinya melakukan kontak dengan layanan kesehatan. (Pohan, 2015)

2.4. Metode *Waterfall*

Metode yang digunakan dalam pengembangan sistem pada penelitian menggunakan siklus hidup pengembangan sistem SDLC (*System Development Life Cycle*) dengan Model *Waterfall*. Model ini melakukan pendekatan secara sistematis dan berurutan. Metode *Waterfall* mempunyai tahapan-tahapan yang digambarkan pada Gambar 2.1.



Gambar 2.1 Metode *Waterfall* (Ian Sommerville, 2011:30)

Menurut (Sommerville, 2011) menyatakan bahwa model *waterfall* diuraikan dengan tahap-tahap sebagai berikut:

1. *Requirement Analysis and definition*

Merupakan tahapan penetapan fitur, analisa kendala dan tujuan sistem melalui konsultasi dengan pengguna sistem. Semua tahapan tersebut akan ditetapkan secara rinci dan berfungsi sebagai spesifikasi sistem.

2. *System and Software Design*

Merupakan tahapan pembentukan arsitektur sistem berdasarkan persyaratan yang telah ditetapkan pada tahap sebelumnya. Pada tahap ini juga mengidentifikasi dan menggambarkan abstraksi dasar sistem perangkat lunak yang akan dibuat serta hubungan- hubungannya.

3. *Implementation and unit testing*

Merupakan tahapan hasil dari desain perangkat lunak untuk direalisasikan sebagai satu set program atau unit program. Setiap unit akan diuji apakah sudah memenuhi spesifikasinya.

4. *Integration and system testing*

Merupakan tahapan pengintegrasian setiap unit program satu sama lain dan diuji sebagai satu sistem yang utuh untuk memastikan sistem sudah memenuhi persyaratan yang ada. Setelah itu sistem akan dikirim ke pengguna sistem.

5. *Operation and Maintenance*

Merupakan tahapan penginstalasian dan penerapan sistem. Pada tahap ini juga dilakukan pengujian pada saat sistem dijalankan untuk menemukan dan memperbaiki error yang tidak ditemukan pada tahap pembuatan. Dalam tahap ini juga dilakukan pengembangan sistem seperti penambahan fitur dan fungsi baru.

2.5. Pemrograman Berorientasi Objek

Metodologi berorientasi objek adalah suatu strategi pembangunan perangkat lunak yang mengorganisasikan perangkat lunak sebagai kumpulan objek

yang berisi data dan operasi yang diberlakukan terhadapnya. Metodologi berorientasi objek merupakan suatu cara bagaimana sistem perangkat lunak dibangun melalui pendekatan objek secara sistematis. Metode berorientasi objek didasarkan pada penerapan prinsip-prinsip pengelolaan kompleksitas. Saat ini sudah banyak Bahasa pemrograman berorientasi objek. Banyak orang yang berfikir bahwa pemrograman berorientasi objek identic dengan basaha Java. Memang Bahasa java merupakan Bahasa yang paling konsisten dalam menimplementasikan paradigma pemrograman berorientasi objek. Namun sebenarnya Bahasa pemrograman yang mendukung pemrograman berorientasi objek tidak hanya Bahasa java. Contohnya: C++, PHP. (A.S. & Shalahudin, 2015)

2.6. *Unified Modeling Languange*

Rosa dan Shalahudin (2016), menjelaskan bahwa *UML (Unified Modeling Language)* adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis dan *design*, serta menggambarkan arsitektur dalam pemrograman berorientasi objek. *UML* menyediakan serangkaian gambar dan diagram yang sangat baik. Beberapa diagram memfokuskan diri pada ketangguhan teori *object oriented* dan sebagian lagi memfokuskan pada detail rancangan dan konstruksi. Semua dimaksudkan sebagai sarana komunikasi antar *team programmer* maupun dengan pengguna. (A.S. & Shalahudin, 2015)

UML mempunyai sejumlah elemen grafis yang bisa dikombinasikan menjadi diagram. *UML* memiliki sejumlah aturan untuk menggabungkan/ mengkombinasikan elemen-elemen tersebut.

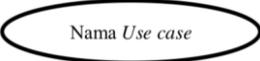
Berikut adalah diagram yang terdapat di dalam *UML*:

1. *Use Case Diagram*

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

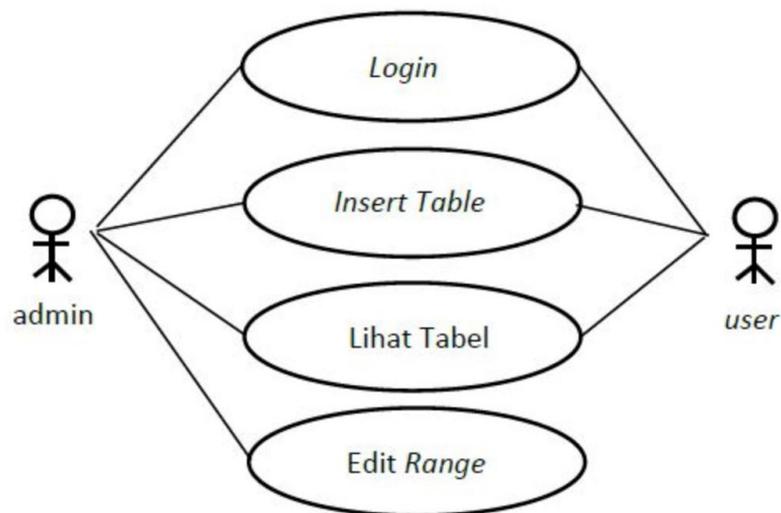
Simbol-simbol yang digunakan dalam *use case* Diagram yaitu:

Tabel 2.1 Simbol-simbol *Use Case Diagram* (A.S. & Shalahudin, 2015)

<p><i>Use Case</i></p> 	<p>Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor. Biasanya dinyatakan dengan menggunakan kata kerja di awal di awal frase nama <i>use case</i>.</p>
<p>Aktor / <i>Actor</i></p> 	<p>Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.</p>
<p>Asosiasi / <i>Association</i></p> 	<p>Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.</p>
<p>Ekstensi / <i>Extend</i></p> 	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu. Mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek.</p>
<p>Generalisasi / <i>Generalization</i></p> 	<p>Hubungan generalisasi dan spesialisasi (umum – khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.</p>

<p>Menggunakan / <i>include</i></p> <p style="text-align: center;">-- <i><<include>></i> --></p>	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> di mana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini.</p>
-----------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Contoh *use case diagram* sederhana:



Gambar 2.2 Contoh Use Case Diagram

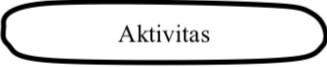
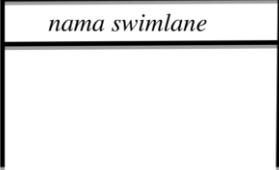
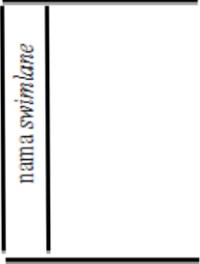
2. Activity Diagram

Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor.

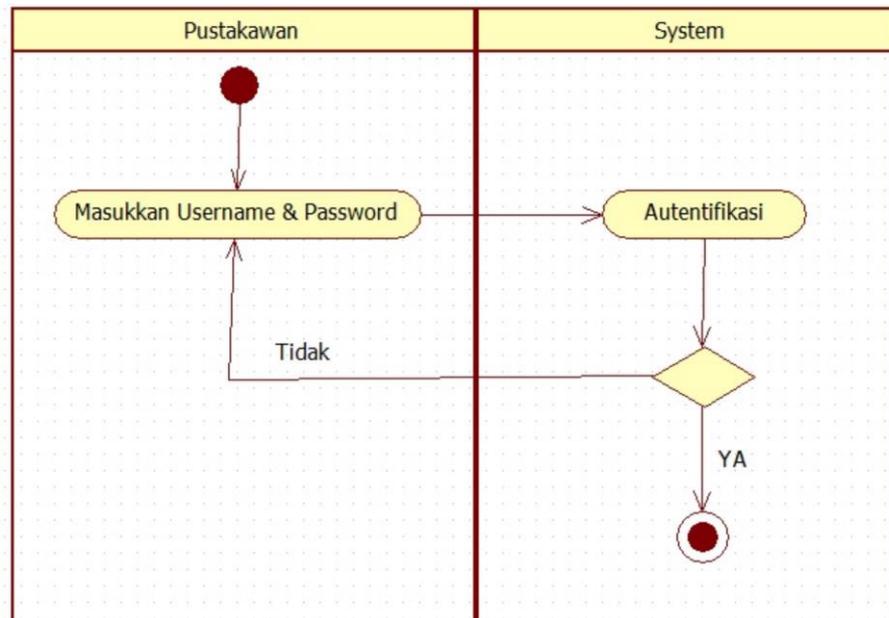
Simbol-simbol yang digunakan dalam *Activity Diagram* yaitu:

Tabel 2.2 Simbol-simbol Activity Diagram(A.S. & Shalahudin, 2015)

<p>Status awal</p> <p style="text-align: center;">●</p>	<p>Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.</p>
---------------------------------------------------------	--------------------------------------------------------------------------------------------

<p>Aktivitas</p> 	<p>Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.</p>
<p>Percabangan / <i>decision</i></p> 	<p>Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.</p>
<p>Penggabungan / <i>join</i></p> 	<p>Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.</p>
<p>Status akhir</p> 	<p>Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.</p>
<p><i>Swimlane</i></p>  <p>atau</p> 	<p>Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.</p>

Contoh *Activity Diagram* sederhana:



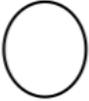
Gambar 2.3 Contoh *Activity Diagram*

3. *Sequence Diagram*

Diagram sequence menggambarkan kelakuan objek pada *use case* dengan medeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek.

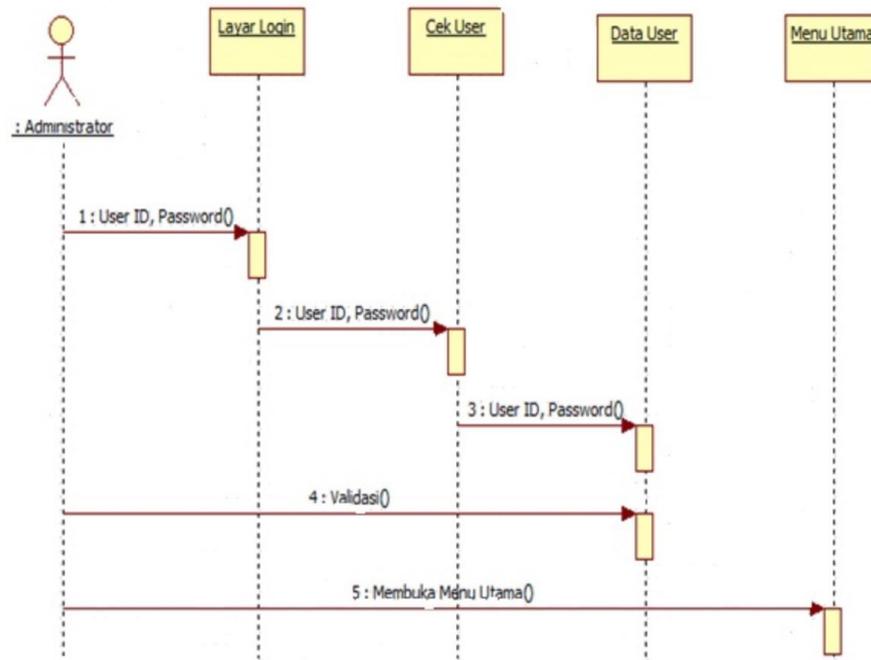
Simbol-simbol yang digunakan dalam *Sequence Diagram* yaitu:

Tabel 2.3 Simbol-simbol *Activity Diagram* (A.S. & Shalahudin, 2015)

<p><i>Entity Class</i></p> 	<p><i>Entity class</i>, merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.</p>
<p><i>Boundary Class</i></p> 	<p><i>Boundary class</i>, berisi kumpulan kelas yang menjadi <i>interface</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan <i>form entry</i> dan <i>form cetak</i>.</p>

<p><i>Control Class</i></p> 	<p><i>Control class</i>, suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek.</p>
<p><i>Message</i></p> 	<p><i>Message</i>, simbol mengirim pesan antar <i>class</i>.</p>
<p><i>Recursive</i></p> 	<p><i>Recursive</i>, menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.</p>
<p><i>Activation</i></p> 	<p><i>Activation</i> mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbandingan lurus dengan durasi aktivitas sebuah operasi.</p>
<p><i>Lifeline</i></p> 	<p><i>Lifeline</i>, garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i>.</p>

Contoh *sequence diagram* sederhana:



Gambar 2.4 Contoh *Sequence Diagram*

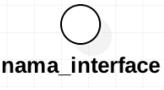
4. *Class Diagram*

Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi.

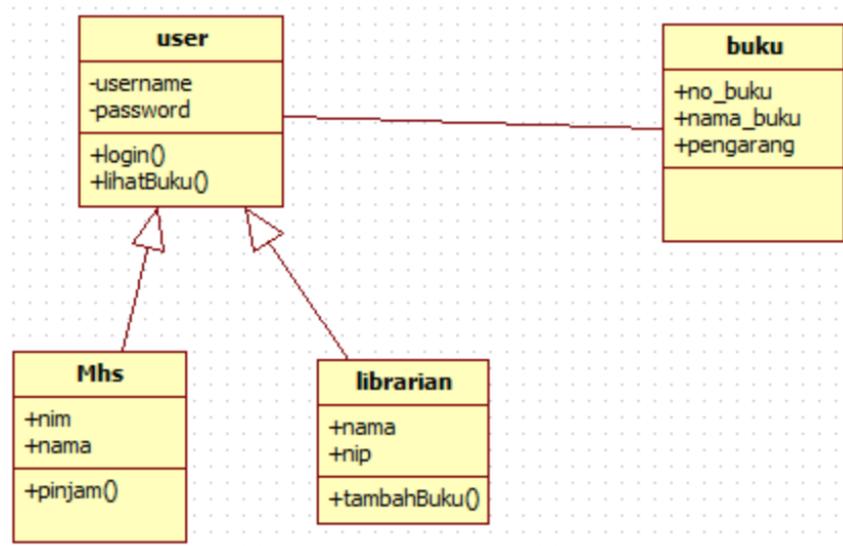
Simbol-simbol yang digunakan dalam *Class Diagram* yaitu:

Tabel 2.4 Simbol-simbol *Class Diagram* (A.S. & Shalahudin, 2015)

<p>Kelas</p> <div style="border: 1px solid black; padding: 5px; margin: 10px auto; width: fit-content;"> <p>nama_kelas</p> <p>+Attribute</p> <p>+Operation()</p> </div>	<p>Kelas pada struktur sistem</p>
<p>Antarmuka / <i>interface</i></p>	<p>Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek</p>

 <p>nama_interface</p>	
<p>Asosiasi / <i>association</i></p> 	<p>Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>.</p>
<p>Asosiasi berarah / <i>directed association</i></p> 	<p>Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>.</p>
<p>Generalisasi</p> 	<p>Relasi antarkelas dengan makna generalisasi-spesialisasi (umum khusus)</p>
<p>Kebergantungan / <i>dependency</i></p> 	<p>Relasi antarkelas dengan makna kebergantungan antarkelas</p>
<p>Agregasi / <i>aggregation</i></p> 	<p>Relasi antarkelas dengan makna semua-bagian (<i>whole-part</i>)</p>

Contoh *class diagram* sederhana:



Gambar 2.5 Contoh *Class Diagram*

5. *Statechart Diagram*

Statechart diagram digunakan untuk menggambarkan perubahan status atau transisi status dari sebuah mesin atau sistem atau objek. Jika diagram sekuen digunakan untuk interaksi antar objek. Perubahan tersebut digambarkan dalam suatu graf berarah.

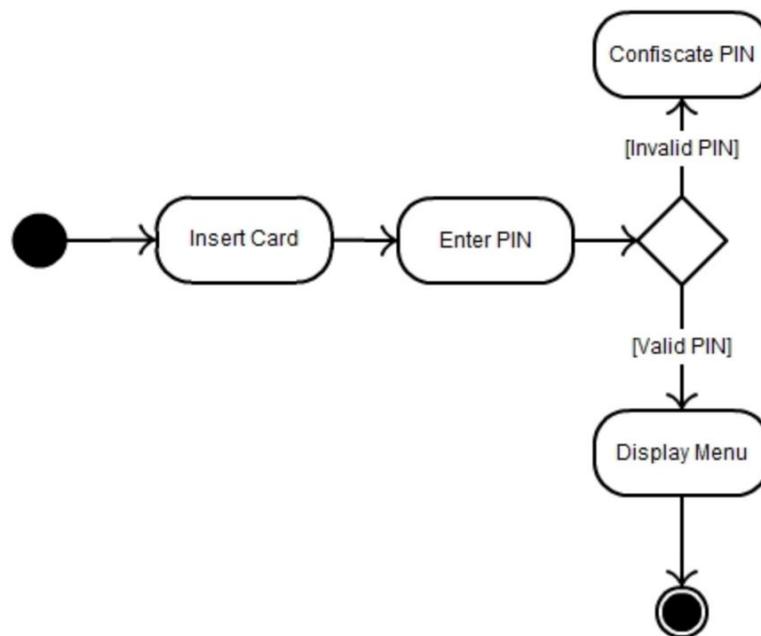
Simbol-simbol yang digunakan dalam *Statechart Diagram* yaitu:

Tabel 2.5 Simbol-simbol *Statechart Diagram* (A.S. & Shalahudin, 2015)

<p><i>Start / Status Awal (Initial State)</i></p> 	<p><i>Start</i> atau <i>initial state</i> adalah <i>state</i> atau keadaan awal pada saat sistem mulai hidup.</p>
<p><i>End / Status Akhir (Final State)</i></p> 	<p><i>End</i> atau <i>final state</i> adalah <i>state</i> keadaan akhir dari daur hidup suatu sistem.</p>

<p><i>Event</i></p> 	<p><i>Event</i> adalah kegiatan yang menyebabkan berubahnya status mesin.</p>
<p><i>State</i></p> 	<p>Sistem pada waktu tertentu. <i>State</i> dapat beubah jika ada event tertentu yang memicu perubahan tersebut.</p>

Contoh *statechart diagram* sederhana:



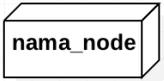
Gambar 2.6 Contoh *Statechart Diagram*

6. *Deployment Diagram*

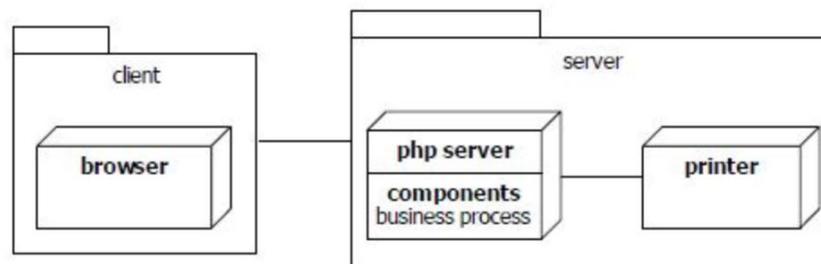
Diagram deployment atau *deployment diagram* menunjukkan konfigurasi komponen dalam proses eksekusi aplikasi.

Simbol-simbol yang digunakan dalam *Deployment Diagram* yaitu:

Tabel 2.6 Simbol-simbol *Deployment Diagram* (A.S. & Shalahudin, 2015)

<i>Package</i> 	<i>Package</i> merupakan sebuah bungkusan dari satu atau lebih node.
<i>Node</i> 	Biasanya mengacu pada perangkat keras (<i>hardware</i>), perangkat lunak tidak dibuat sendiri (<i>software</i>).
Kebergantungan / <i>dependency</i> 	Kebergantungan antar <i>node</i> , arah panah mengarah pada <i>node</i> yang dipakai.
<i>Link</i> 	Relasi antar <i>node</i>

Contoh *Deployment Diagram* sederhana:

**Gambar 2.7 Contoh *Deployment Diagram***

2.7. Basis Data (*Database*)

Menurut Fathansyah (2012) Basis data terdiri dari 2 kata, yaitu Basis dan Data. Basis kurang lebih dapat diartikan sebagai markas atau gudang, tempat bersarang/berkumpul. Data adalah represtasi fakta nyata yang mewakili suatu objek seperti manusia (pegawai, siswa, pembeli, pelanggan), barang, hewan yang

diwujudkan dalam bentuk angka, huruf, simbol, teks, gambar, bunyi, atau kombinasinya. Sebagai satu kesatuan istilah, Basis Data (*Database*) sendiri dapat didefinisikan, himpunan kelompok data (arsip) yang saling berhubungan yang diorganisasi sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah. (Fathansyah, 2015)

Berikut merupakan nama beberapa aplikasi *database* yang dikategorikan menurut komersi dan nonkomersil:

1. Aplikasi database komersil seperti: *DB2, Microsoft SQL Server, Oracle, Sybase* dan *Teradata*.
2. Aplikasi database nonkomersil seperti: *MySQL, Firebird, PostgreSQL*.

2.8. *MySQL (Structured Query Language)*

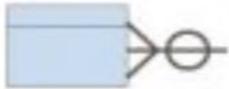
MySQL adalah sebuah *server database open source* yang terkenal yang digunakan berbagai aplikasi terutama untuk server atau membuat *webservice*. *MySQL* berfungsi sebagai *Structured Query Language (SQL)* yang dimiliki sendiri dan sudah diperluas oleh *MySQL* yang umumnya digunakan bersamaan dengan *PHP* untuk membuat aplikasi *server* yang dinamis dan *powerfull*. Kehandalan suatu sistem basisdata (*DBMS*) dapat diketahui dari cara kerja dalam melakukan proses perintah-perintah *SQL* yang dibuat oleh pengguna maupun program-program aplikasi yang memanfaatkannya. Sebagai pengoperasi basisdata, *MySQL* mendukung operasi basisdata transaksional maupun operasi basisdata nontransaksional. Pada modus operasi nontransaksional, *MySQL* dapat dikatakan unggul dalam hal untuk kerja dibandingkan perangkat lunak pengoperasi basisdata competitor lainnya. Pada modus nontransaksional tidak ada jaminan atas reliabilitas

terhadap data yang tersimpan, karenanya modus non-transaksional hanya cocok untuk jenis aplikasi yang tidak membutuhkan reliabilitas data seperti aplikasi *blogging* berbasis *web* (*wordpress*), *CMS*, dan sejenisnya. Kebutuhan sistem yang ditujukan untuk bisnis sangat disarankan untuk menggunakan modus basisdata transaksional, hanya saja sebagai konsekuensinya unjuk kerja *MySQL* pada modus transaksional tida secepat unjuk kerja pada modus non-transaksional. (Irsan, 2015)

2.9. Entity Relationship Diagram (ERD)

Menurut (Cerry et al., 2017) *ERD* adalah diagram yang menggambarkan keterhubungan antar data secara konseptual. Penggambaran keterhubungan antar data ini didasarkan pada anggapan bahwa dunia nyata dari kumpulan objek yang disebut entitas (*entity*), dan hubungan yang terjadi diantaranya yang disebut relasi (*relationship*).

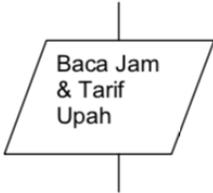
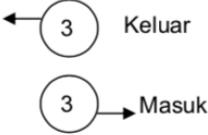
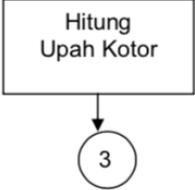
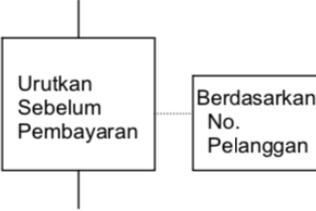
Tabel 2.7 Kardinalitas Pada *ERD* Versi James Martin

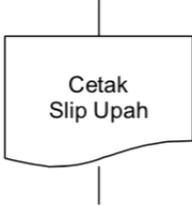
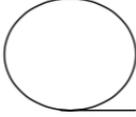
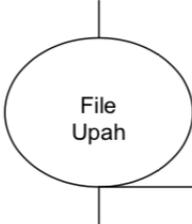
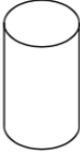
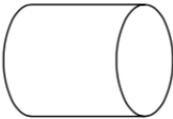
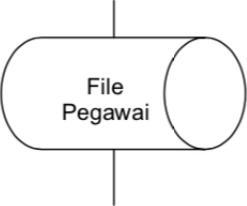
Simbol	Arti
	Satu dan hanya satu (One and only one)
	Satu atau lebih (One or more)
	Tidak ada atau lebih (Zero or more)
	Tidak ada atau satu (Zero or one)

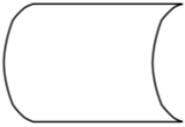
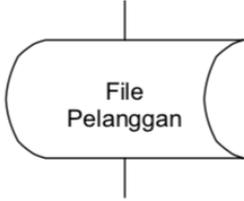
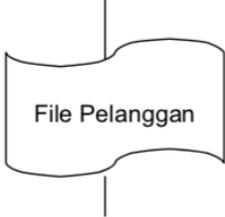
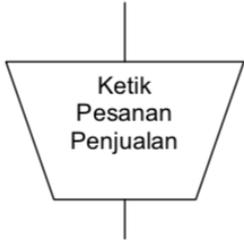
Flowchart

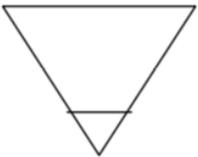
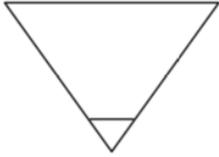
Menurut (Iswandy et al., 2015) bahwa *flowchart* merupakan urutan-urutan langkah kerja suatu proses yang digambarkan dengan menggunakan simbol-simbol yang disusun secara sistematis. Simbol-simbol yang digunakan untuk menggambarkan *flowchart* dapat di lihat seperti pada gambar 2.8:

Tabel 2.8 Simbol - Simbol *Flowchart*

SIMBOL	ARTI	CONTOH
<p>Input / Output</p> 	<p>Merepresentasikan Input data atau Output data yang diproses atau Informasi.</p>	
<p>Proses</p> 	<p>Mempresentasikan operasi</p>	
<p>Penghubung</p> 	<p>Keluar ke atau masuk dari bagian lain flowchart khususnya halaman yang sama</p>	
<p>Anak Panah</p> 	<p>Merepresentasikan alur kerja</p>	
<p>Penjelasan</p> 	<p>Digunakan untuk komentar tambahan</p>	

SIMBOL	ARTI	CONTOH
<p>Dokumen</p> 	<p>I/O dalam format yang dicetak</p>	
<p>Magnetic Tape</p> 	<p>I/O yang menggunakan pita magnetik</p>	
<p>Magnetic Disk</p> 	<p>I/O yang menggunakan disk magnetik</p>	
<p>Magnetic Drum</p> 	<p>I/O yang menggunakan drum magnetik</p>	

SIMBOL	ARTI	CONTOH
<p>On-line Storage</p> 	<p>I/O yang menggunakan penyimpanan akses langsung</p>	
<p>Punched Tape</p> 	<p>I/O yang menggunakan pita kertas berlubang</p>	
<p>Manual Input</p> 	<p>Input yang dimasukkan secara manual dari keyboard</p>	
<p>Display</p> 	<p>Output yang ditampilkan pada terminal</p>	
<p>Manual Operation</p> 	<p>Operasi Manual</p>	

SIMBOL	ARTI	CONTOH
<p data-bbox="347 443 568 510">Communication Link</p> 	<p data-bbox="619 477 971 577">Transmisi data melalui channel komunikasi, seperti telepon</p>	<p data-bbox="994 539 1107 595">Terminal Komputer</p> 
<p data-bbox="347 736 568 770">Off-line Storage</p> 	<p data-bbox="619 770 971 871">Penyimpanan yang tidak dapat diakses oleh komputer secara langsung</p>	

2.10. *Black Box Testing*

Black box atau uji fungsional, kondisi uji dikembangkan atas program atau fungsi sistem yaitu tester membutuhkan informasi tentang data *input* dan *output* yang diamati, tetapi tidak tahu bagaimana program atau sistem bekerja. Tester berfokus pada pengujian fungsi program terhadap spesifikasi tersebut. Pada pengujian *black-box*, tester memandang program sebagai kotak hitam dan benar-benar tidak peduli dengan struktur internal dari program atau sistem (Luthfi, 2017).

2.11. *Android*

2.11.1. *Sejarah Android*

Android adalah sebuah sistem operasi untuk perangkat mobile berbasis linux yang mencakup system operasi, *middleware* dan aplikasi. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi. (Utara, 2013) Menurut Jubilee Enterprise, android adalah sebuah sistem operasi yang memberi kemudahan dalam berkirim email melalui fasilitas Gmail Android merupakan sistem operasi mobile berbasis kernel Linux yang dikembangkan oleh Android Inc dan kemudian diakui sisi oleh Google.

Google ponsel masuk ke pasar ponsel bukan sebagai produsen handset, tapi dengan meluncurkan platform mobile yang disebut sebagai “Android” untuk perangkat mobile seperti smartphone. Google memiliki visi yaitu ponsel yang berbasis android memiliki fungsi yang seperti tersedia pada PC. Google meluncurkan *Open Handset Alliance*. Google memperkenalkan android sebagai OS (*Operating System*) yang menjalankan aplikasi yang menarik dan memberikan banyak pilihan pada pengguna untuk memilih aplikasi yang mereka sukai. Android mendapat dukungan dari arsitektur *hardware*. Maka dua aspek android yaitu bersifat open source dan model arsitektur. *Open source* dimaksudkan agar android mudah dianalisis, dipahami untuk diperbaiki bagian yang *bug* dan *port hardware* baru. Disisi lain model arsitektur nya pun menambah penggunaan pada fitur-fitur. Aplikasi android berbasis Java, dan faktor ini memerlukan penggunaan mesin virtual VM yang disebut Dalvik, yaitu menafsirkan dan mengeksekusi Java bergaya

portable kode byte, yang beroperasi pada platform mobile. (Gandhewar & Sheikh, 2010)

Awalnya, Google Inc. membeli android Inc. yang merupakan pendatang baru yang membuat piranti lunak untuk ponsel/smartphone. Kemudian untuk mengembangkan android dibentuklah *Open Handset Alliance*, konsorsium dari 34 perusahaan piranti keras, piranti lunak, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-mobile, dan Nvidia. Android SDK dirilis pertama kali pada 12 November 2007 dan para pengembang memiliki kesempatan untuk memberikan umpan balik dari pengembang SDK tersebut. Pada bulan September 2008, T-mobile memperkenalkan ketersediaan T-mobile G1 yang merupakan smartphone pertama berbasis platform android. Beberapa hari kemudian Google merilis android SDK 1.0 Google membuat *source code* dari platform android menjadi tersedia dibawah lisensi *Apanche's Open Source Google* merilis perangkat genggam (disebut Android Dev Phone 1) yang dapat menjalankan aplikasi Android tanpa terikat oleh berbagai jaringan *provider* telepon seluler pada akhir 2008.

Tujuan dari perangkat ini adalah memungkinkan pengembang untuk melakukan percobaan dengan perangkat sebenarnya yang dapat menjalankan Android OS tanpa berbagai kontrak. Google juga merilis versi 1.1 dari sistem operasi Android pada waktu yang tidak lama. Versi 1.1 dari Android tidak mendukung adanya *soft keyboards* dan membutuhkan perangkat yang memiliki *keyboard* secara fisik. (Utara, 2013)

Android menyelesaikan masalah ini dengan merilis versi 1.5 pada bulan April 2009 dengan sejumlah tambahan fitur seperti kemampuan perekaman media, *widgets*, dan *live folders*. Versi 1.6 dari Android OS dirilis pada bulan September 2009 dan hanya dalam waktu satu bulan versi Android 2.0 dirilis dan membanjiri seluruh perangkat Android. Versi ini memiliki kemampuan *advanced search*, *text to speech*, *gestures*, dan *multi touch*. Android 2.0 memperkenalkan kemampuan untuk menggunakan HTML karena didukung oleh HTML 5. Semakin banyak aplikasi berbasis Android setiap harinya yang terdapat pada *application store* secara *online* atau dikenal sebagai *Android Market*.

Secara garis besar, arsitektur android dapat dijelaskan sebagai berikut: (Murtiwiyati & Lauren, 2013)

- *Applications dan Widgets Applications*, adalah layer yang berhubungan dengan aplikasi saja, biasanya download aplikasi dijalankan kemudian dilakukan instalasi dan jalankan aplikasi tersebut.
- *Applications Frameworks*, adalah layer para pembuat aplikasi melakukan pengembangan/pembuatan aplikasi yang akan dijalankan pada sistem operasi android, karena pada layer inilah aplikasi dapat dirancang dan dibuat, seperti *contact-provider* yang berupa sms dan panggilan telepon.
- *Libraries*, adalah layer untuk fitur-fitur android berada, biasanya para pembuat aplikasi mengakses libraries untuk menjalankan aplikasinya. Berjalan diatas kernel, layer ini meliputi berbagai library C/C++ inti seperti *Libe* dan *SSL*.

- *Android Run Time* yang membuat aplikasi android dapat dijalankan dalam prosesnya menggunakan implementasi Linux.
- *Linux kernel*, adalah layer inti dari operating sistem android itu berada. Berisi file-file system yang mengatur sistem *processing*, *memory*, *resource*, *drivers*, dan sistem-sistem operasi android lainnya. Linux kernel yang digunakan android adalah linux kernel 2.6.

Layanan aplikasi berikut adalah pilar arsitektur semua aplikasi android, memberikan kerangka kerja yang akan digunakan oleh pengembang. Berikut penjelasannya : (Esmaeel, 2015)

- 1) *Activity Manager Controls*, yaitu siklus hidup kegiatan, termasuk pengelolaan kegiatan stack.
- 2) *View*, digunakan untuk membangun *user interface* untuk membuat aplikasi.
- 3) *Notification Manager Providers*, merupakan mekanisme yang konsisten dan non-intrusif untuk signaling pengguna.
- 4) *Content Provides Lets*, merupakan data pengguna aplikasi untuk berbagai aplikasi.

2.11.2. Versi Android

1. *Android 1.5 Cupcake*

Cupcake dirilis 30 April 2009. *Cupcake* menjadi versi android pertama yang menggunakan nama makanan. Konon katanya versi ini seharusnya versi 1.2, namun *Google* memutuskan untuk membuat revisi besar dan membuatnya menjadi versi 1.5 *Cupcake* adalah kue kecil yang dipanggang dalam cetakan berbentuk cup.

2. *Android 1.6 Donut*

Android 1.6, codename *Donut*, dirilis pada 15 September 2009. Pada versi ini diperbaiki beberapa kesalahan *reboot*, perubahan fitur foto dan video dan integrasi pencarian yang lebih baik. Donat merupakan panganan berbentuk cincin bulat bolong tengah. Adonan donat dimasak dengan cara digoreng dan biasanya disajikan dengan topping di atasnya.

3. *Android 2.0/2.1 Éclair*

Android 2.0/2.1 Eclair Dirilis 26 Oktober 2009. *Eclair* adalah makanan penutup yakni kue yang biasanya berbentuk persegi panjang yang dibuat dengan krim di tengah dan lapisan cokelat di atasnya.

4. *Android 2.2 Froyo*

Dirilis 20 Mei 2010. Menggunakan *codename Froyo*, yang merupakan makan penutup yang nama merek sebuah produk yang terbuat dari *Yoghurt*. *Froyo* singkatan dari *Frozen Yoghurt*, *Froyo* adalah *yoghurt* yang telah mengalami proses pendinginan, sehingga secara terlihat sama seperti es krim.

5. *Android 2.3 Gingerbread*

Android versi 2.3 *Gingerbread* dirilis resmi tanggal 6 Desember 2010. *Gingerbread* merupakan jenis kue kering yang dengan rasa jahe. Kue jahe biasanya dibuat pada perayaan hari libur akhir tahun di Amerika. Biasanya cemilan kering ini dicetak berbentuk tubuh manusia.

6. *Android 3.0 Honeycomb*

Dirilis tanggal 22 February 2011. *Honeycomb* adalah sereal sarapan manis

yang sudah dibuat tahun 1965 oleh *Posting Sereal*. Seperti namanya, *Honeycomb*/sarang lebah, sereal ini terbuat dari potongan jagung berbentuk sarang lebah dengan rasa madu.

7. *Android 4.0 Ice Cream Sandwich*

Android 4.0-4.0.2 API Level 14 dan *4.0.3-4.0.4 API Level 15* pertama dirilis 19 Oktober 2011. Dinamai *Ice Cream Sandwich*. *Ice Cream Sandwich* adalah lapisan es krim, biasanya rasa vanilla yang terjepit di antara dua kue coklat, dan biasanya berbentuk persegi panjang.

8. *Android 4.1 Jelly bean*

Android Jelly Bean diluncurkan pertama kali pada Juli 2012, dengan berbasis *Linux Kernel 3.0.31*. Terdiri dari *Android 4.1 API Level 16*, *Android 4.2 API Level 17*, *Android 4.3 API Level 18*. Penamaan *Jelly Bean* mengadaptasi nama sejenis permen dalam beraneka macam rasa buah. Ukurannya sebesar kacang merah. Permen ini keras di luar tapi lunak di dalam serta lengket bila digigit.

9. *Android 4.4 KitKat*

Android 4.4 Kitkat API level 19. *Google* mengumumkan *Android KitKat* (dinamai dengan izin dari *Nestle* dan *Hershey*) pada 3 september 2013. Dengan tanggal rilis 31 Oktober 2013. *KitKat* merupakan merk sebuah coklat yang dikeluarkan oleh *Nestle*. Rilis berikutnya setelah nama *KitKat* diperkirakan banyak pengamat akan diberi nomor 5.0 dan dinamai '*Key Lime Pie*'.

10. *Android 5.0 Lollipop*

Android 5.0 atau yang lebih dikenal dengan nama *Lollipop* pertama kali diperkenalkan pada 25 juni 2014. Bertepatan dengan momen tersebut, beragam hasil produk teknologi dari *Google* juga diperkenalkan. Antara lain *Android TV* dan *platform* pelacakan kesehatan *Google Fit*.

11. *Android 6.0 Marshmallow*

Sistem proteksi *Android* dengan metode sidik jari pertama sekali dirancang pada OS *Android Marshmallow*. Penggunaan proteksi sidik jari pada *Android M* ini bisa digunakan untuk proses otentikasi *Play Store* dan pembelian dengan sistem *Android Pay*. Disamping itu, sidik jari banyak digunakan oleh pengguna *Marshmallow* untuk membuka kode kunci dengan cepat tanpa harus menyetikkan *password* pada layar *smartphone*.

12. *Android 7.0 Nougat*

Saya termasuk salah satu pengguna *Nougat*, sejauh ini saya merasa sangat puas dengan versi *Android* ini. Hal yang belum pernah saya temukan pada *OS Android* sebelumnya. Mode multitasking membuat para pengguna bisa menjalankan 2 aplikasi dalam 1 layar. Sehingga kamu tidak perlu menghabiskan banyak waktu untuk menjalankan aplikasi yang kamu butuhkan. Selain itu, kecepatan untuk moving dari satu layar ke layar lainnya sangat *smooth* dan responsif bagi pengguna. Jika kamu belum pake *Nougat*, segera *upgrade* sekarang dan rasakan kecanggihannya.

13. *Android 8.0 Oreo*

Meskipun sampai sekarang belum resmi dirilis oleh *Google*, *Android* versi 8.0 diprediksi akan dilahirkan dengan nama *Android Oreo*. Mengingat

google akan meluncurkan *android* berikutnya dengan awalan huruf O (berdasarkan huruf abjad). Disamping itu, muncul suatu bocoran beberapa waktu lalu dari Senior *VP Android*, Hiroshi Lockheimer, lewat tweet nya @lockheimer mengunggah GIF dengan gambar kue dari susunan Oreo. Namun ini hanya rumor yang beredar di dunia maya. Bisa jadi *Google* akan meluncurkan *Android 8.0* dengan nama yang tidak pernah diprediksi sebelumnya. Kita tunggu saja gebrakan hebat dari raksasa *Google* dalam memperjuangkan *Android* sebagai *OS* terpopuler di dunia versi *smartphone*.

2.11.3. *Android studio*

Android Studio merupakan *Integrated Development Environment (IDE)* resmi untuk pengembangan aplikasi *Android*, berdasarkan *IntelliJ IDEA*. *Android* berubah menjadi platform yang begitu cepat dalam melakukan inovasi. Hal ini tidak lepas dari pengembangan utama dibelakangnya yaitu *Google*, *Google* mengakuisisi *Android* dan kemudian membuatkan sebuah platform.

Platform *android* terdiri dari Sistem Operasi berbasis *Linux*, sebuah *GUI (Graphic User Interface)*, sebuah web browser dan Aplikasi *Studio End-User* yang dapat di download dan juga para pengembang bisa dengan leluasa berkarya serta menciptakan aplikasi yang terbaik dan terbuka untuk digunakan oleh berbagai macam perangkat.

2.11.4. *Arsitektur Android*



Gambar 2.8 Arsitektur Android

Arsitektur android ditunjukkan pada gambar 2.8 yang terdiri dari Aplikasi, *Framework Aplikasi*, *Libraries*, *Android Runtime*, dan Linux Kernel. Aplikasi adalah lapisan paling atas yang menyediakan satu set aplikasi termasuk email, program SMS, Kalender, peta, browser, kontak, dan lain-lain. Semua aplikasi yang ditulis menggunakan bahasa pemrograman Java. Aplikasi dapat digunakan dalam waktu yang bersamaan, contoh ketika akan mengirim pesan lewat email ataupun SMS, aplikasi android pun dapat memutar music diwaktu yang bersamaan. (Gandhewar & Sheikh, 2010)

Framework aplikasi adalah sebuah kerangka kerja perangkat lunak yang digunakan untuk menerapkan struktur standar dari sebuah aplikasi untuk sistem operasi tertentu. Melalui bantuan *Managers*, penyediaan konten dan layanan aplikasi lain dapat berfungsi di bawah aplikasi yang sudah ada, yaitu aplikasi kerangka yang terdiri dari dua bagian sebagai libraries yang semua ditulis dalam bahasa C/C++, ini disebut sebagai antarmuka Java. Termasuk pada Manager, 2D, dan grafis 3D, Codec media seperti MPEG-4 dan MP3, SQL, SQLite database dan Engine browser web WebKit. (Gandhewar & Sheikh, 2010)

Bagian lapisan kedua adalah *Runtime android*, yaitu yang mencakup satu *set libraries* inti yang menyediakan sebagian besar fungsi yang tersedia di *libraries* inti dari pemrograman Java. Setiap aplikasi android dapat berjalan sendiri. Contoh dari mesin *Dalvik Virtual*. Dalvik VM mengeksekusi file dalam *Dalvik executable (Dex)* format yang dioptimalkan untuk meminimalisir jejak memori. Lapisan terakhir adalah Linux Kernel, android pada dasarnya bergantung pada linux versi 2.6 untuk layanan system inti, seperti keamanan, manajemen memori, proses manajemen, jaringan stack, dan driver model kernel juga bertindak sebagai lapisan abstraksi antara hardware dan software stack. (Gandhewar & Sheikh, 2010)

2.12. Virtual Private Server

VPS atau *Virtual Private Server* atau *Virtual Dedicated Server* adalah sebuah server yang dibagi menjadi beberapa *virtual server* yang dapat diinstall OS dan berbagai aplikasinya sendiri. VPS atau VDS itu sendiri merupakan teknologi yang memungkinkan sebuah komputer (*server*) dengan kapasitas *resource hardware* yang sangat besar dapat dibagi-bagi menjadi beberapa *virtual* komputer yang mandiri.

VPS dapat berjalan layaknya sebuah *Dedicated Server* dan juga dapat diinstall sistem operasi (OS) tersendiri serta dapat mengatur virtual komputernya tanpa mengganggu virtual komputer yang lain. VPS atau VDS juga memiliki banyak kegunaan, diantaranya yaitu:

1. Hosting atau Webservice

VPS seringkali digunakan sebagai sebuah *webservice* untuk hosting ataupun untuk berbagai jenis *webservice*, seperti Web Komersil, Institusi

Pemerintahan, maupun Organisasi. Karena, *hosting* atau *webserver* pada umumnya menggunakan sebuah *resource* yang besar dan dengan menggunakan VPS/VDS ini mampu untuk mencukupi *resource* tersebut.

2. *File Server*

Jadikan VPS sebagai pusat data untuk bisnis Anda. Selalu tersedia kapanpun Anda ingin menempatkan dan mengambilnya. *Cloud VPS Cloudmatika* memberikan *unlimited bandwidth* untuk semua jenis VPS yang membuat pengiriman *file* dan menerima *file* menjadi lebih nyaman.

3. *Game Server*

Dengan *latency* yang rendah membuat kecepatan internet menjadi lebih kencang. *VPS Cloudmatika* menyediakan VPS dengan *latency* yang rendah dan sangat cocok sebagai *Game Server*. Sehingga para pengguna *game* yang Anda buat menjadi lebih nyaman

2.13. *PHP*

Script PHP adalah bahasa pemrograman yang berjalan pada sebuah *webserver*, atau sering disebut *server-side*. Oleh karena itu, *PHP* dapat melakukan apa saja yang bisa dilakukan program *CGI (Common Gateway Interface)* lain, yaitu mengolah data dengan tipe apapun, menciptakan halaman web yang dinamis, serta menerima dan menciptakan *cookies*, dan bahkan *PHP* bias melakukan lebih dari itu. Pengertian *script server-side* adalah, agar dapat menjalankan *script* ini dibutuhkan tiga program utama, yaitu *webserver* (dapat berupa *IIS* dari *windows* atau *apache*), modul *PHP* dan juga *web browser*. *PHP* dapat berjalan pada semua jenis *system operasi*, antara lain pada *Linux* dan varian *Unix* (*HP-UX*, *Solaris* dan *OpenBSD*), pada *Ms Windows*, *Mac* dan masih banyak lagi. Selain itu *PHP* juga

dapat berjalan pada beberapa jenis *webserver* antara lain *Apache*, *Microsoft IIS*, *personal webserver*, *Netscape* dan *Iplanet Server*, *Caudium*, *Xitami*, *Omnihttpd* dan masih banyak lagi. (Irsan, 2015)

2.14. Framework

Menurut Octafian (2015:2) bahwa *framework* dapat diartikan sebagai koleksi atau kumpulan potongan-potongan program yang disusun atau diorganisasikan sedemikian rupa, sehingga dapat digunakan untuk membantu membuat aplikasi tanpa harus membuat semua kodenya dari awal. Saat ini ada banyak *framework PHP*, diantaranya: *Zend*, *Cake PHP*, *Codeigniter*, *Laravel* dan sebagainya. (Octafian, 2015)

2.15. Laravel

Laravel adalah sebuah *MVC web development framework* yang didesain untuk meningkatkan kualitas perangkat lunak dengan mengurangi biaya pengembangan dan perbaikan serta meningkatkan produktifitas pekerjaan dengan sintak yang bersih dan fungsional yang dapat mengurangi banyak waktu untuk *implementasi*. (Luthfi, 2017)

BAB III

ANALISA MASALAH DAN PERANCANGAN PROGRAM

3.1 Analisis

Analisis ini bertujuan untuk mengidentifikasi permasalahan-permasalahan yang terdapat pada sistem yang akan dibangun. Adapun sistem yang akan dibangun ini adalah sebuah sistem *appointment* bidan berbasis *android*, aplikasi ini dibuat sebagai sarana untuk memudahkan masyarakat atau calon pasien dalam berkomunikasi untuk *appointment* pemeriksaan bidan, imunisasi maupun melahirkan, selain itu pasien dapat mengetahui daftar rekap medis pasien pribadi.

3.1.1 Analisis Masalah

Analisis masalah merupakan langkah awal dari analisis sistem, langkah ini diperlukan untuk mengetahui permasalahan apa saja yang terjadi di dalam sistem yang telah berjalan. Adapun identifikasi masalah pada penelitian ini adalah bagaimana cara meningkatkan kualitas pelayanan bidan praktik agar dapat memudahkan masyarakat dalam berkomunikasi atau mengetahui jadwal dan layanan bidan.

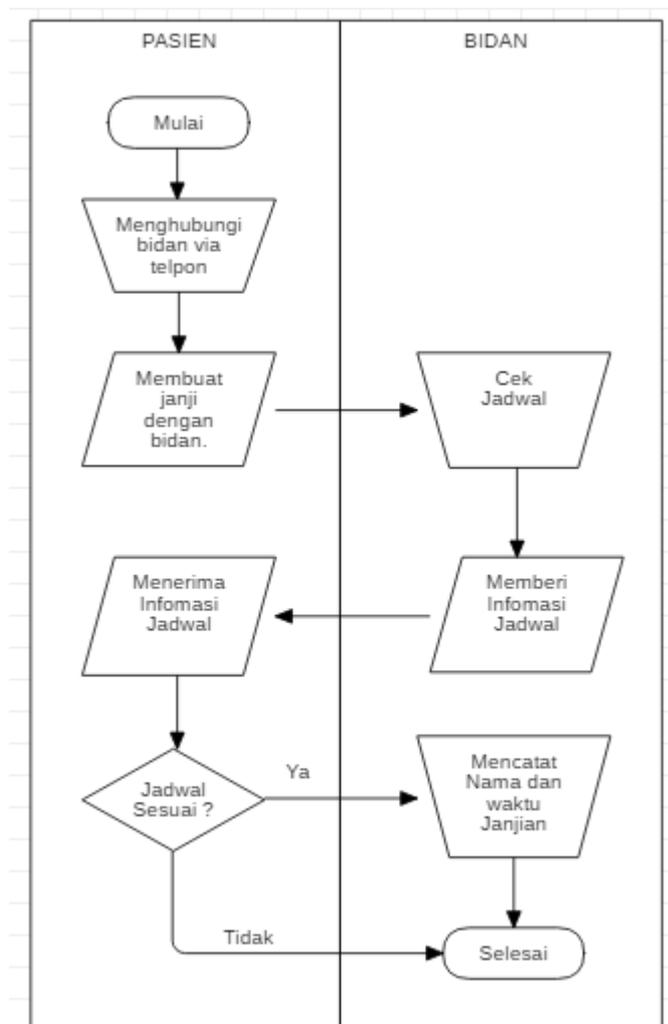
3.1.2 Analisis Sistem yang Sedang Berjalan

Analisis terhadap sistem yang berjalan bertujuan untuk mengetahui lebih jelas bagaimana kerja suatu sistem dan mengetahui masalah yang dihadapi sistem untuk dapat dijadikan landasan di sarankan perancangan sistem. Tahap analisis

sistem perlu dilakukan terlebih dahulu sebelum dilakukan perancangan sistem. Tahap analisis ini merupakan tahapan yang penting di dalam merancang suatu sistem. Analisa sistem yang berjalan dilakukan untuk mengetahui masalah apa yang sedang dihadapi oleh pihak bidan.

3.1.2.1 Analisis Prosedur *Appointment* Bidan

Analisis prosedur *appointment* bidan merupakan tahapan untuk melakukan *appointment* dengan bidan di klinik yang sedang berjalan saat ini.



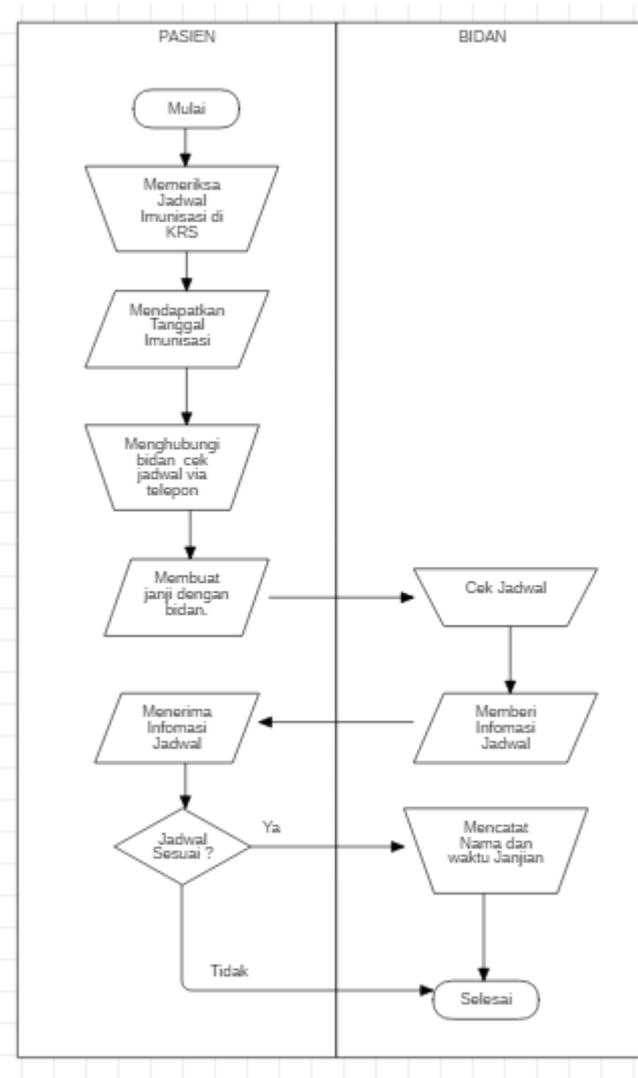
Gambar 3.1 *Flowchart* Sistem *Appointment* Bidan yang Sedang Berjalan

Pasien akan menghubungi bidan melalui telepon dan membuat janji.

1. Pasien mendapatkan informasi dari bidan diantaranya:
 - a. Bidan akan menyampaikan ketersediaan jadwal sesuai keinginan pasien.
 - b. Bidan akan memberikan saran apabila jadwal tidak sesuai dengan keinginan pasien.
 - c. Bidan akan mencatat informasi data pasien apabila jadwal sesuai dengan pasien.

3.1.2.2 Analisis Prosedur Imunisasi

Analisis prosedur imunisasi merupakan tahapan untuk melakukan imunisasi yang sedang berjalan saat ini.

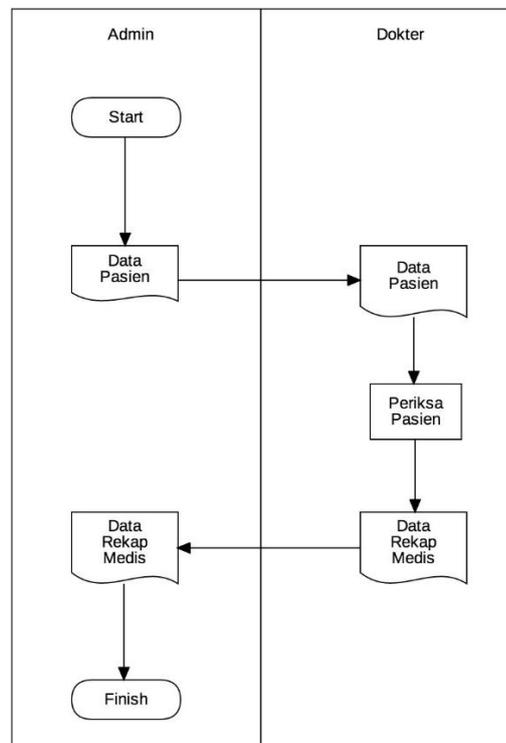


Gambar 3.2 Flowchart Sistem Imunisasi yang Sedang Berjalan

1. Pasien memeriksa kartu kunjungan kapan harus melakukan imunisasi.
2. Pasien menghubungi bidan melalui telepon dan menanyakan ketersediaan jadwal dan obat imunisasi yang akan diambil.
3. Pasien mendapatkan informasi dari bidan diantaranya:
 - a. Bidan akan menyampaikan ketersediaan obat imunisasi di kliniknya.
 - b. Apabila jadwal tersedia dan sesuai maka bidan akan mencatat informasi data pasien.

3.1.2.3 Analisis Prosedur Rekap Medis

Analisis prosedur rekap medis merupakan tahapan untuk melakukan pendataan rekap medis yang dilakukan oleh bidan.



Gambar 3.3 Flowchart Sistem Rekap Medis yang Sedang Berjalan

1. Admin memberikan informasi data pasien kepada bidan.
2. Bidan akan memeriksa data pasien dan melakukan pemeriksaan.
3. Bidan akan mencatat informasi rekap medis dan dikembalikan kembali kepada admin.

3.1.3 Analisis Pengguna

Analisis perangkat merupakan tahapan untuk menentukan perangkat yang akan digunakan saat membuat aplikasi dan menjalankan aplikasi agar berjalan

secara optimal. Perangkat yang dianalisis terdiri dari perangkat keras (*hardware*) dan perangkat lunak (*software*) yang harus memenuhi spesifikasi yang disarankan.

1. Perangkat Keras (*Hardware*)

Perangkat keras yang akan digunakan oleh pengguna ataupun *developer* harus memenuhi standar minimal dari aplikasi yang akan digunakan ataupun digunakan ataupun dikembangkan.

a. *Developer*

Perangkat kerja yang digunakan oleh *developer* dalam membuat aplikasi minimal harus memiliki spesifikasi sebagai berikut:

- 1) Processor : Intel® Core™ 2 Duo CPU
- 2) RAM : 4GB
- 3) Harddisk : 60 GB
- 4) VGA : Intel® HD Graphics 1 GB
- 5) Resolusi Monitor : 1366 x 768 (32bit/60Hz)

b. *Pengguna (User)*

Perangkat keras yang akan digunakan oleh pengguna (*User*) agar mendukung fitur aplikasi yang dibuat adalah *smartphone* dengan minimal spesifikasi sebagai berikut:

- 1) Processor : Quad-Core 1.3 GHz Cortex-A7
- 2) Layar : *Touchscreen*
- 3) Resolusi Layar : 480 x 800 pixels
- 4) RAM : 512 MB
- 5) ROM *Free Space* : 50 MB

2. Perangkat Lunak (*Software*)

Adapun spesifikasi dari perangkat lunak (*software*) yang digunakan oleh pengguna ataupun *developer* agar proses penggunaan ataupun pembuatan aplikasi berjalan secara optimal.

a. *Developer*

- 1) Sistem Operasi : Microsoft Windows 7
- 2) Perangkat Lunak Pendukung :
 - *Sublime Text 3*
 - XAMPP 7.2.7
 - *Android SDK*

b. *Pengguna (User)*

Pengguna (*User*) yang akan menggunakan aplikasi harus menggunakan *smartphone* dengan sistem operasi Android versi 5.0 (*Lollipop*) atau versi yang lebih tinggi.

3.2 Desain

Desain merupakan tahap membuat arsitektur program, gaya, tampilan dan kebutuhan material atau bahan untuk program. Tujuan pada tahap desain adalah untuk menentukan secara detail perancangan arsitektur, gaya dan semua material yang akan digunakan.

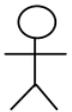
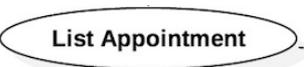
3.2.1 Perancangan Sistem Perangkat Lunak

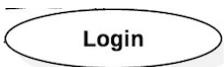
Perancangan sistem perangkat lunak memiliki tujuan memberikan gambaran dari sistem yang akan dibuat. Penulis menggunakan diagram *UML* diantaranya *use*

case diagram, activity diagram, class diagram, sequence diagram, statechart diagram dan deployment digram.

3.2.1.1 Deskripsi Perancangan *Use Case*

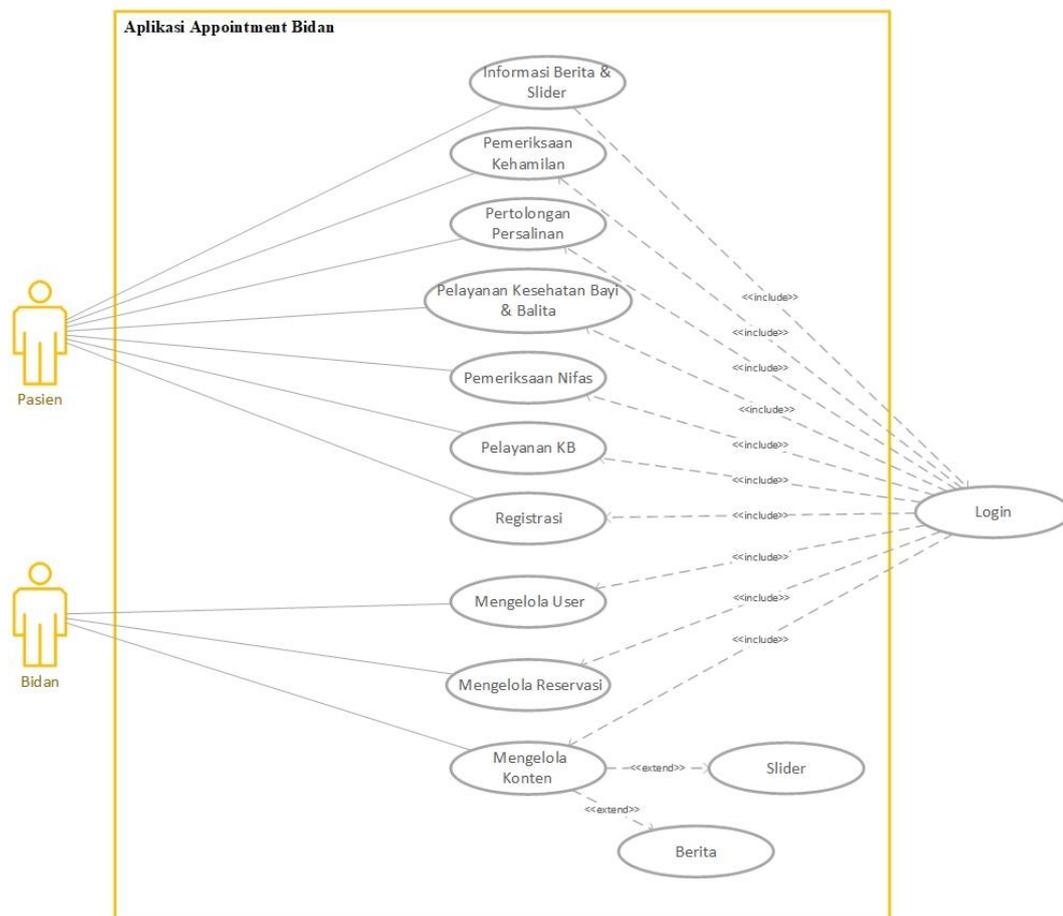
Tabel 3.1 Deskripsi Perancangan *Use Case*

No	Simbol	Deskripsi
1	 Pasien	Pihak yang mempunyai hak akses untuk melihat informasi pelayanan bidan yang tersedia dan dapat membuat <i>appointment</i> dengan bidan. Selain itu pasien dapat melihat riwayat data rekap medis pasien tersebut.
2	 Bidan	Pihak yang mempunyai hak akses untuk melihat informasi data pasien dan melakukan pendataan riwayat rekap medis untuk pasien.
3		Masuk kedalam menu untuk membuat <i>appointment</i> dengan bidan.
4		Masuk kedalam menu yang akan menampilkan daftar <i>appointment</i> yang sudah terdatar.
5		Masuk kedalam menu untuk menampilkan data profil user pasien.
6		Menampilkan halaman pendaftaran pasien. Sebelum melakukan <i>appointment</i> pasien harus melakukan pendaftaran terlebih dahulu.

7		Menampilkan halaman login aplikasi sebelum masuk ke halaman utama.
---	-----------------------------------------------------------------------------------	--------------------------------------------------------------------

3.2.1.2 Perancangan *Use Case Diagram*

Use case diagram digunakan untuk menggambarkan sistem dari sudut pandang pengguna sistem tersebut, seperti pada gambar 3.4. di bawah ini:



Gambar 3.4 *Use Case Diagram* Aplikasi *Appointment Bidan*

1. Use Case Skenario

Berdasarkan gambar 3.3 skenario dari aktivitas tersebut dapat dilihat pada Tabel 3.2 sampai dengan Tabel 3.17:

Tabel 3.2 Skenario Use Case Melihat Jadwal Bidan

Identifikasi	
<i>Use Case</i>	Melihat jadwal bidan
Aktor	Pasien
Deskripsi	Memilih bidan lalu Masuk kedalam menu jadwal untuk melihat jadwal bidan.
Kondisi Awal	Membuka menu utama
Aktivitas Aktor	Aktor menekan tombol menu jadwal bidan.
Aktivitas Sistem	Memproses <i>request</i> aktor.
Kondisi Akhir	Menampilkan informasi jadwal bidan.

Tabel 3.3 Skenario Use Case Membuat Appointment Bidan

Identifikasi	
<i>Use Case</i>	Membuat <i>Appointment</i>
Aktor	Pasien
Deskripsi	Masuk kedalam menu untuk membuat <i>appointment</i> dengan bidan.
Kondisi Awal	Membuka menu utama.
Aktivitas Aktor	Aktor menekan tombol menu membuat <i>appointment</i> .
Aktivitas Sistem	Memproses <i>request</i> aktor.
Kondisi Akhir	Menampilkan data daftar Apointment.

Tabel 3.4 Skenario Use Case Melihat List Appointment

Identifikasi	
<i>Use Case</i>	<i>List Appointment</i>
Aktor	Pasien
Deskripsi	Masuk kedalam menu yang akan menampilkan daftar <i>appointment</i> yang sudah terdatar.
Kondisi Awal	Membuka menu utama.
Aktivitas Aktor	Aktor menekan tombol menu <i>list appointment</i> .
Aktivitas Sistem	Memproses <i>request</i> aktor dan menampilkan data <i>list appointment</i> .
Kondisi Akhir	Masuk Menu <i>list appointment</i> .

Tabel 3.5 Skenario Use Case Skenario Manage data pribadi

Identifikasi	
<i>Use Case</i>	<i>Manage data pribadi</i>
Aktor	Pasien
Deskripsi	Masuk kedalam menu profil untuk menampilkan data profil user pasien.
Kondisi Awal	Membuka menu utama.
Aktivitas Aktor	Aktor menekan tombol menu <i>update data</i> .
Aktivitas Sistem	Memproses <i>request</i> aktor.
Kondisi Akhir	Menampilkan data profil user pasien.

Tabel 3.6 Skenario Use Case Registrasi

Identifikasi	
<i>Use Case</i>	Registrasi
Aktor	Pasien
Deskripsi	Menampilkan halaman pendaftaran pasien. Sebelum melakukan <i>appointment</i> pasien harus melakukan pendaftaran terlebih dahulu.
Kondisi Awal	Membuka halaman <i>login</i> .
Aktivitas Aktor	Aktor menekan tombol menu daftar.
Aktivitas Sistem	Memproses <i>request</i> aktor dan menyimpan data pasien.
Kondisi Akhir	Menampilkan pesan.

Tabel 3.7 Skenario Use Case Login

Identifikasi	
<i>Use Case</i>	<i>Login</i>
Aktor	Pasien, Bidan
Deskripsi	Menampilkan halaman login aplikasi sebelum masuk ke halaman utama.
Kondisi Awal	Membuka aplikasi <i>appointment</i> .
Aktivitas Aktor	Aktor memasukan email dan password.
Aktivitas Sistem	Memproses <i>request</i> aktor.
Kondisi Akhir	Jika berhasil menampilkan menu utama jika salah munculkan pesan.

Tabel 3.8 Skenario Use Case Melihat Rekam Medis

Identifikasi	
<i>Use Case</i>	Melihat rekam medis
Aktor	Pasien
Deskripsi	Masuk kedalam menu yang akan menampilkan daftar data rekap medis yang sudah dilakukan oleh pasien.
Kondisi Awal	Membuka menu utama.
Aktivitas Aktor	Aktor menekan tombol menu rekap medis.
Aktivitas Sistem	Memproses <i>request</i> aktor.
Kondisi Akhir	Masuk menu daftar rekam medis.

Tabel 3.9 Skenario Use Case Manage rekam Medis

Identifikasi	
<i>Use Case</i>	<i>Manage</i> Rekam Medis
Aktor	Bidan
Deskripsi	Masuk kedalam menu untuk menambahkan data rekam medis yang dapat dilakukan oleh bidan.
Kondisi Awal	Membuka menu utama.
Aktivitas Aktor	Aktor menekan tombol menu membuat rekam medis.
Aktivitas Sistem	Memproses <i>request</i> aktor.
Kondisi Akhir	Masuk ke <i>form</i> tambah rekam medis.

Tabel 3.10 Skenario Use Case Registrasi pasien

Identifikasi	
<i>Use Case</i>	<i>Registrasi pasien</i>
Aktor	Bidan
Deskripsi	Masuk kedalam menu untuk mengelola data pasien lalu menambahkan data pasien.
Kondisi Awal	Membuka menu utama.
Aktivitas Aktor	Aktor menekan tombol menu <i>manage</i> pasien.
Aktivitas Sistem	Memproses <i>request</i> aktor.
Kondisi Akhir	Masuk menu <i>manage</i> pasien.

Tabel 3.11 Skenario Use Case Manage Jadwal

Identifikasi	
<i>Use Case</i>	<i>Manage Jadwal</i>
Aktor	Bidan
Deskripsi	Masuk kedalam menu untuk mengelola data jadwal yang tersedia di setiap klinik yang di lakukan oleh Bidan klinik.
Kondisi Awal	Membuka menu utama.
Aktivitas Aktor	Aktor menekan tombol menu <i>manage</i> jadwal.
Aktivitas Sistem	Memproses <i>request</i> aktor.
Kondisi Akhir	Masuk menu <i>manage</i> jadwal.

Tabel 3.12 Skenario *Use Case Manage klinik*

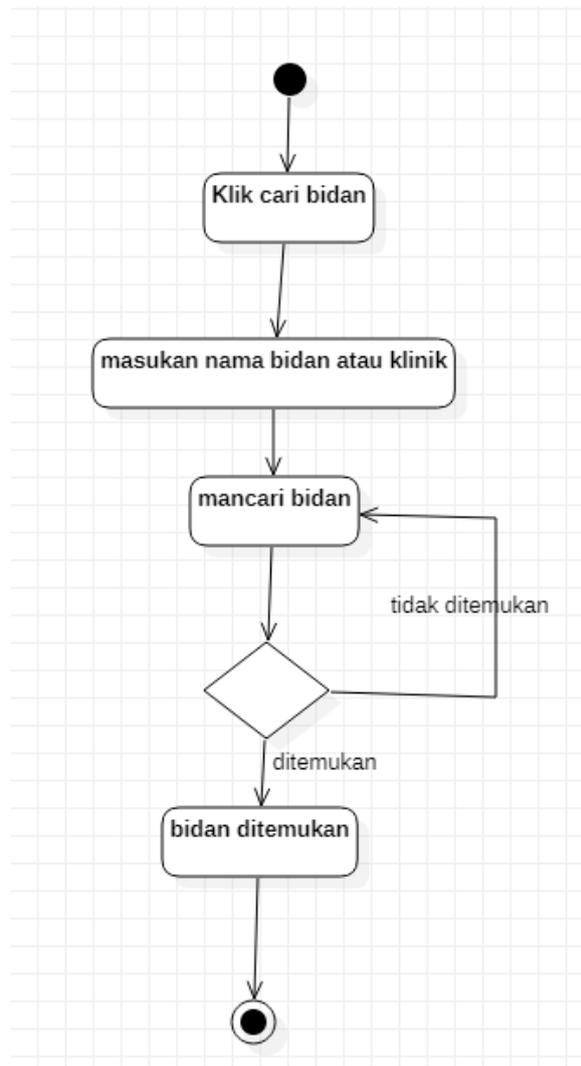
Identifikasi	
<i>Use Case</i>	Manage Klinik
Aktor	bidan
Deskripsi	Masuk kedalam menu untuk mengelola data klinik yang di lakukan oleh bidan.
Kondisi Awal	Membuka menu utama.
Aktivitas Aktor	Aktor menekan tombol menu <i>manage</i> klinik.
Aktivitas Sistem	Memproses <i>request</i> aktor.
Kondisi Akhir	Masuk menu <i>manage</i> klinik.

3.2.1.3 Activity Diagram

Activity diagram digunakan untuk menggambarkan *event-event* yang terjadi di dalam *use case diagram* aplikasi *appointment* klinik pada gambar 3.4.

1. Activity Diagram mencari bidan

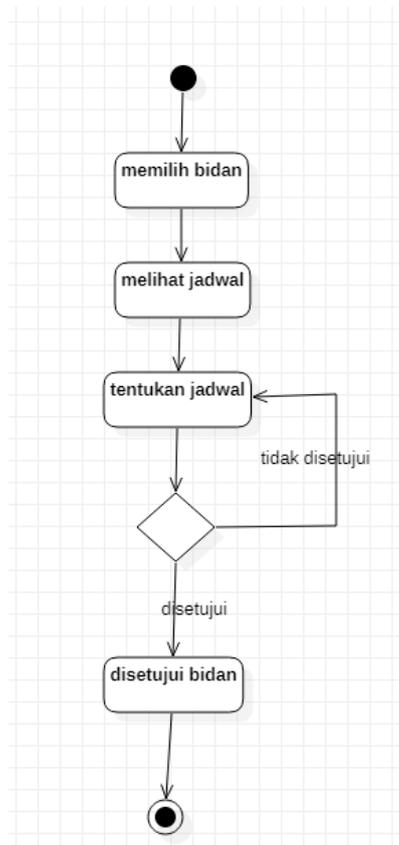
Pada gambar 3.5, *activity diagram* tersebut menggambarkan proses untuk mencari bidan.



Gambar 3.5 Activity Diagram Mencari Bidan

2. Activity Diagram Membuat Appointment bidan

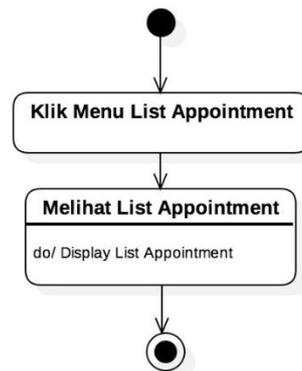
Pada gambar 3.6, *activity diagram* tersebut menggambarkan proses untuk membuat *appointment* bidan.



Gambar 3.6 Activity Diagram Membuat Appointment Bidan

3. *Activity Diagram Daftar Appointment*

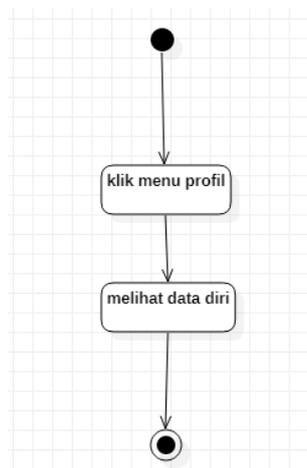
Pada gambar 3.7, *activity diagram* tersebut menggambarkan proses untuk menampilkan *list appointment* pasien.



Gambar 3.7 Activity Diagram List Appointment

4. *Activity Diagram melihat data diri*

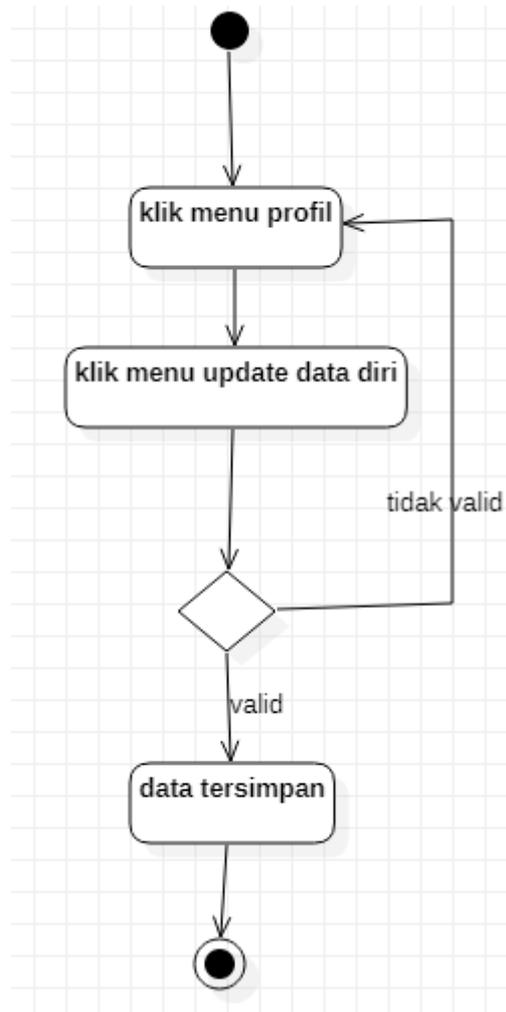
Pada gambar 3.8, *activity diagram* tersebut menggambarkan proses untuk menampilkan data pasien.



Gambar 3.8 Activity Diagram melihat Data Pasien

5. *Activity Diagram* manage data pribadi

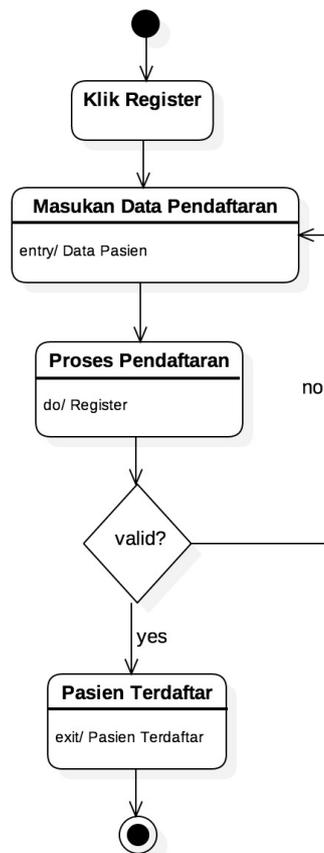
Pada gambar 3.9, *activity diagram* tersebut menggambarkan proses untuk mengubah data pasien.



Gambar 3.9 *Activity Diagram* Manage Data Diri

6. *Activity Diagram* Register

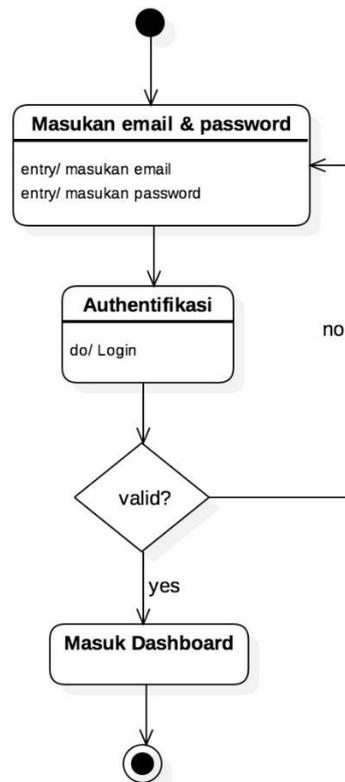
Pada gambar 3.10, *activity diagram* tersebut menggambarkan proses untuk daftar pasien kedalam aplikasi.



Gambar 3.10 *Activity Diagram Register Pasien*

7. *Activity Diagram Login*

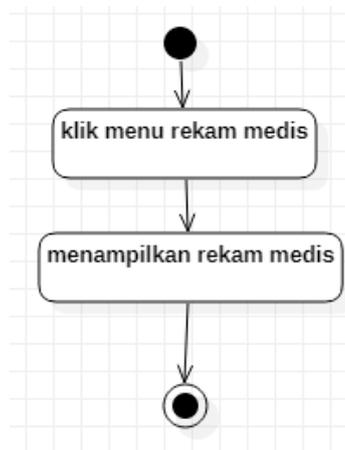
Pada gambar 3.11, *activity diagram* tersebut menggambarkan proses untuk *login* masuk ke dalam aplikasi.



Gambar 3.11 Activity Diagram Login

8. Activity Diagram Melihat Daftar Rekam Medis

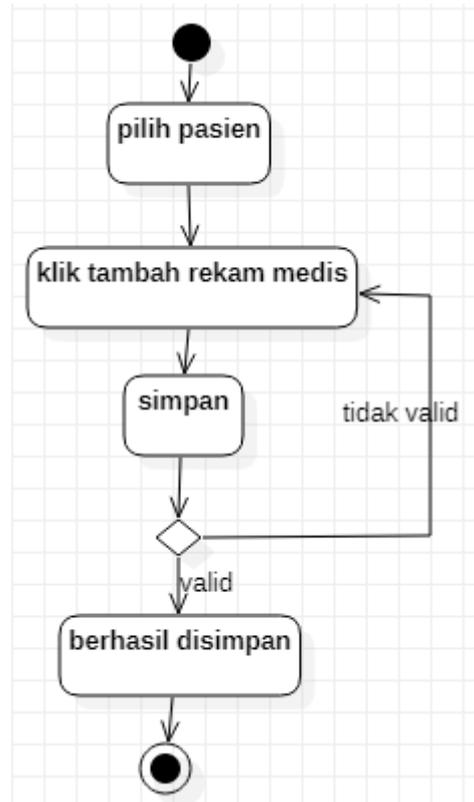
Pada gambar 3.12, *activity diagram* tersebut menggambarkan proses untuk melihat daftar rekap medis.



Gambar 3.12 Activity Diagram Melihat Daftar Rekap Medis

9. *Activity Diagram Menambah Data Rekam Medis*

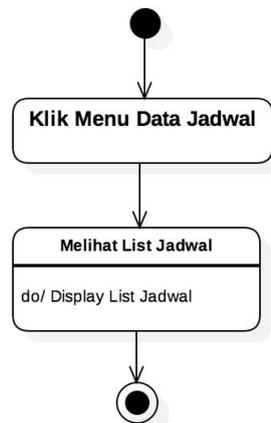
Pada gambar 3.13, *activity diagram* tersebut menggambarkan proses menambahkan data rekap medis yang dilakukan oleh bidan.



Gambar 3.13 *Activity Diagram Menambah Data Rekam Medis*

10. *Activity Diagram Manage Jadwal*

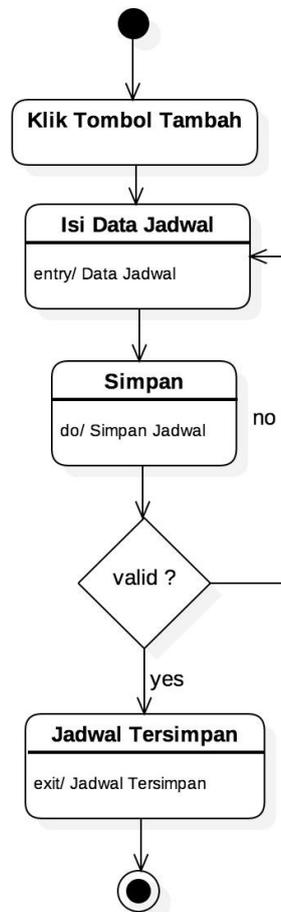
Pada gambar 3.18, *activity diagram* tersebut menggambarkan proses menampilkan daftar data jadwal.



Gambar 3.14 Activity Diagram Manage Jadwal

11. Activity Diagram Tambah Data Jadwal

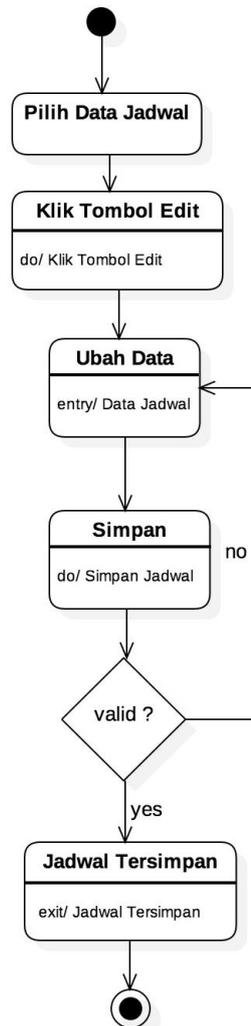
Pada gambar 3.15, *activity diagram* tersebut menggambarkan proses menambahkan data jadwal.



Gambar 3.15 Activity Diagram Tambah Data Jadwal

12. Activity Diagram Ubah Data Jadwal

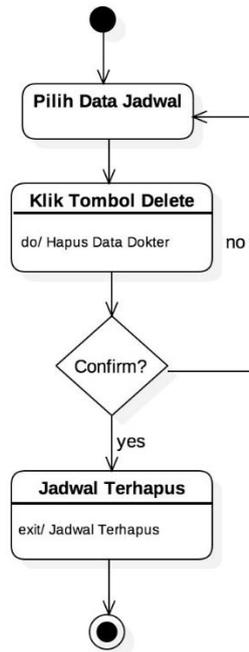
Pada gambar 3.16, *activity diagram* tersebut menggambarkan proses mengubah data jadwal.



Gambar 3.16 Activity Diagram Ubah Data Jadwal

13. Activity Diagram Hapus Data Jadwal

Pada gambar 3.21, *activity diagram* menggambarkan proses menghapus data jadwal.



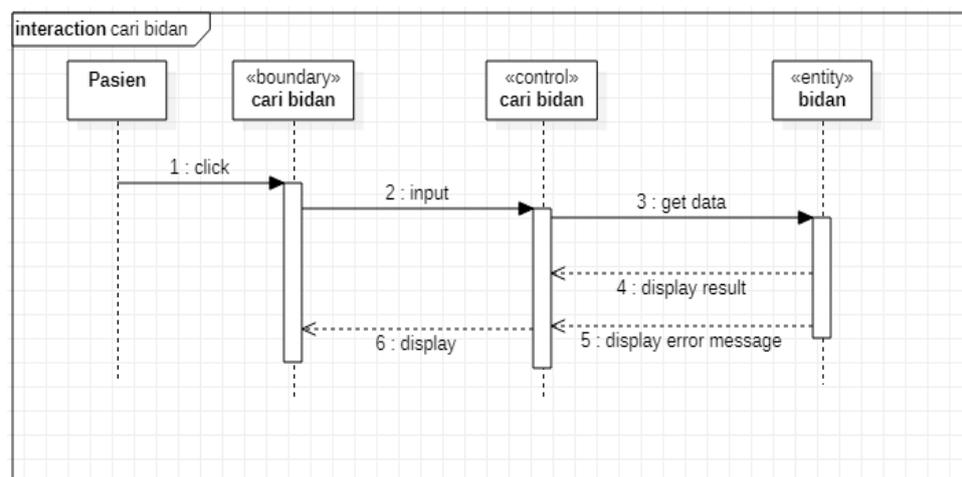
Gambar 3.17 *Activity Diagram* Hapus Data Jadwal

3.2.1.4 Sequence Diagram

Sequence diagram merupakan gambaran interaksi antara sejumlah objek dalam urutan waktu. Bertujuan untuk menunjukkan rangkain pesan yang dikirim antar objek.

1. *Sequence Diagram* cari bidan

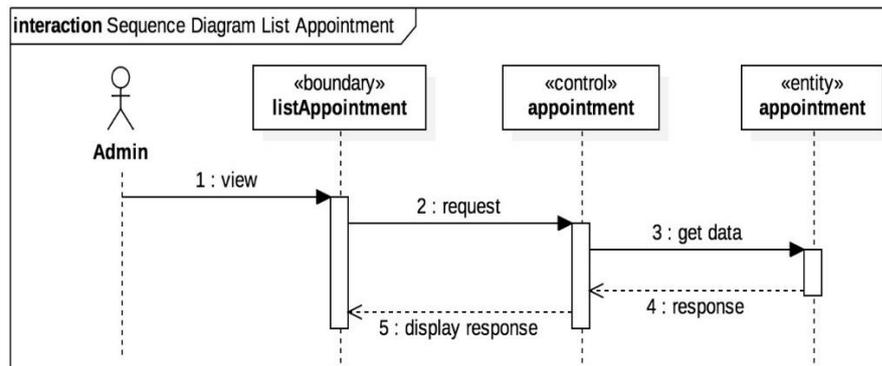
Sequence Diagram mencari bidan menggambarkan urutan interaksi pada saat pasien akan mencari bidan.



Gambar 3.18 *Sequence Diagram* Mencari bidan

2. *Sequence Diagram* Melihat Daftar *Appointment*

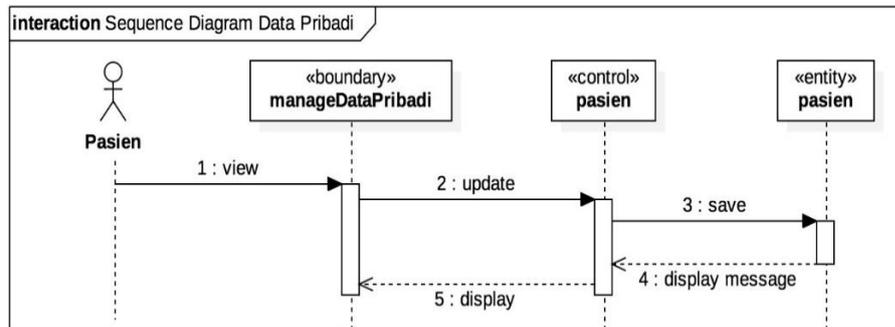
Sequence diagram melihat *appointment* menggambarkan urutan interaksi untuk menampilkan daftar *appointment* yang sudah dilakukan oleh pasien.



Gambar 3.19 *Sequence Diagram* Melihat Daftar *Appointment*

3. *Sequence Diagram manage data diri*

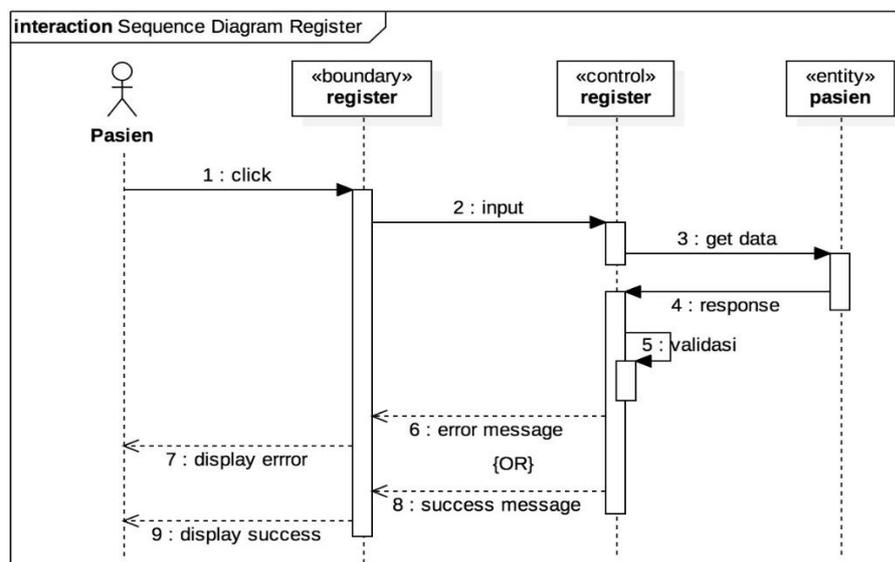
Sequence diagram setting data pasien menggambarkan urutan interaksi saat proses melakukan perubahan data pasien.



Gambar 3.20 *Sequence Diagram Setting Data Pasien*

4. *Sequence Diagram Register*

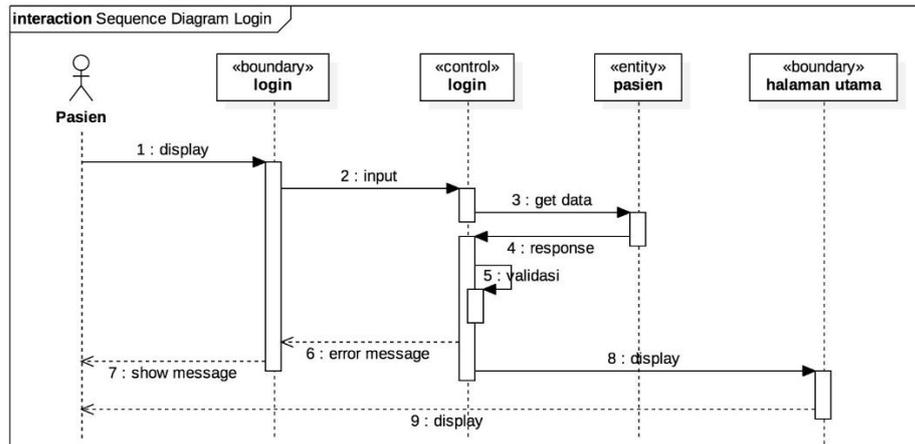
Sequence diagram register menggambarkan urutan interaksi saat proses saat pasien melakukan pendaftaran.



Gambar 3.21 *Sequence Diagram Register*

5. Sequence Diagram Login

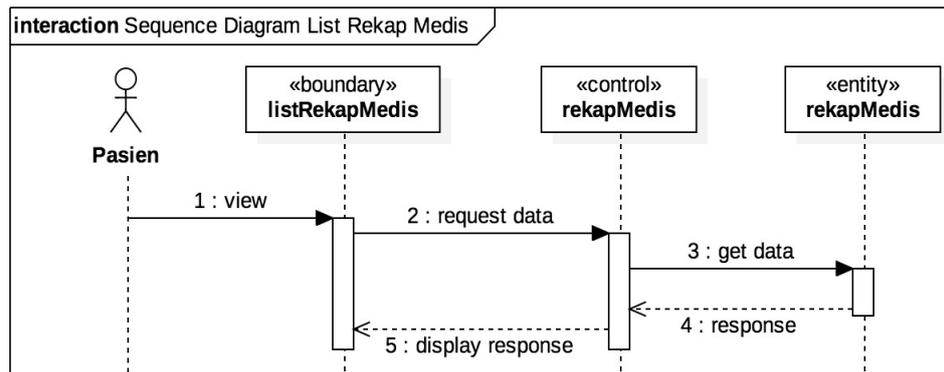
Sequence diagram login menggambarkan urutan interaksi saat proses saat pasien login masuk aplikasi.



Gambar 3.22 Sequence Diagram Login

6. Sequence Diagram Melihat Daftar Rekam Medis

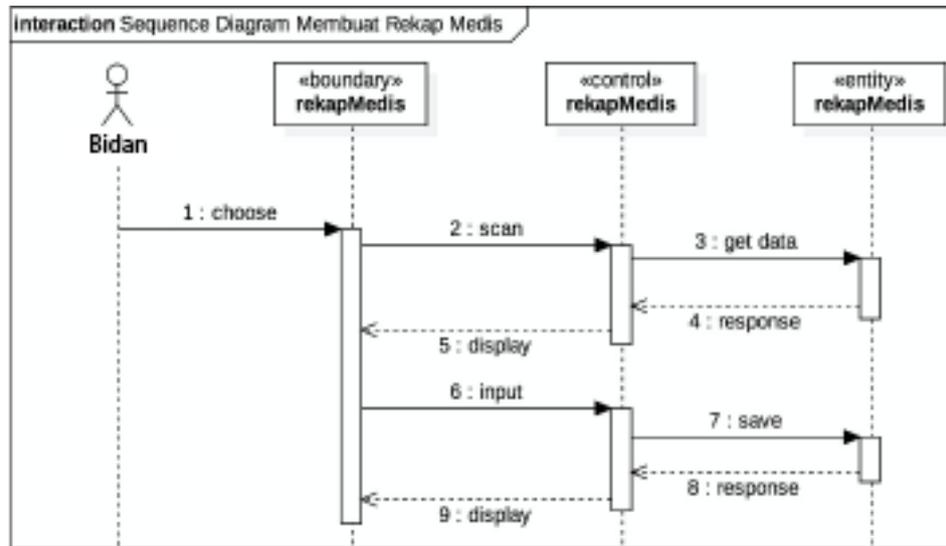
Sequence diagram melihat rekap medis menggambarkan urutan interaksi untuk menampilkan daftar rekap medis.



Gambar 3.23 Sequence Diagram Melihat Rekap Medis

7. *Sequence Diagram Membuat Rekam Medis*

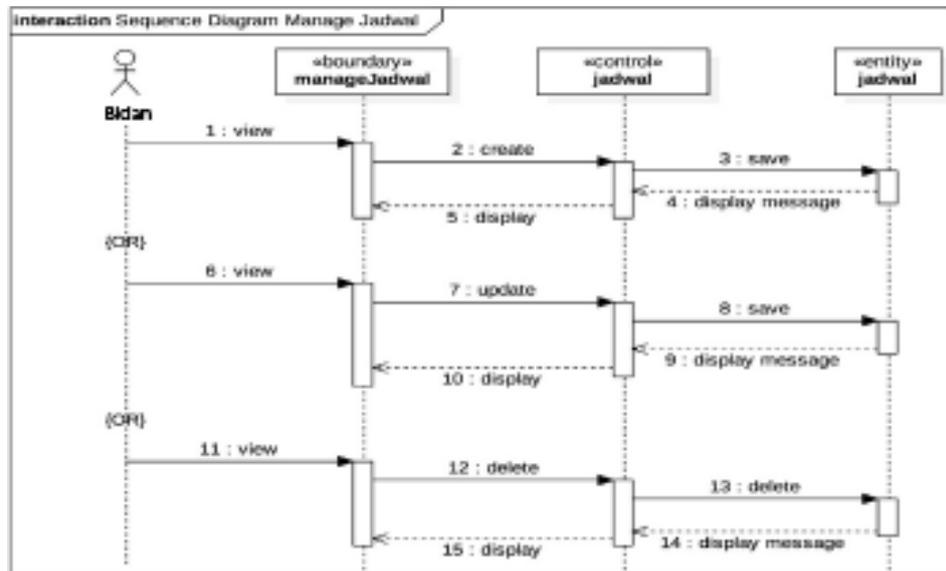
Sequence diagram membuat rekam medis menggambarkan urutan interaksi proses bidan dalam menambahkan data rekam medis.



Gambar 3.24 *Sequence Diagram Membuat Rekap Medis*

8. *Sequence Diagram Manage Jadwal*

Sequence Diagram manage jadwal menggambarkan urutan interaksi proses dalam memmanage data jadwal.



Gambar 3.25 *Sequence Diagram Manage Jadwal*

3.2.1.5 State Chart Diagram

Statechart diagram menggambarkan transisi dan perubahan keadaan (dari satu *state* ke *state* lainnya) pada suatu objek pada sistem sebagai sebuah gambaran.

1. *Statechart Diagram Melihat Daftar Appointment*

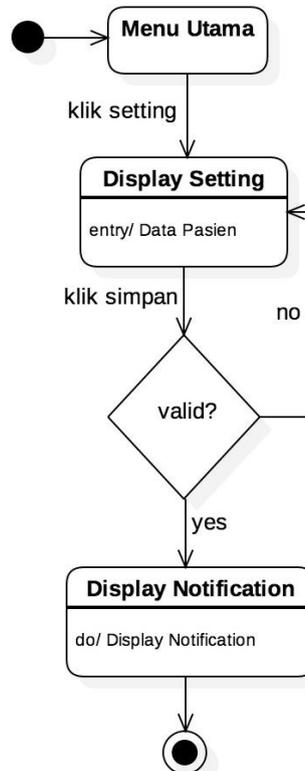
Statechart diagram melihat daftar *appointment* menggambarkan transisi dan perubahan keadaan untuk melihat daftar *appointment* yang dilakukan oleh pasien.



Gambar 0.26 *Statechart Diagram Melihat List Appointment*

2. *Statechart Diagram Setting Data Pasien*

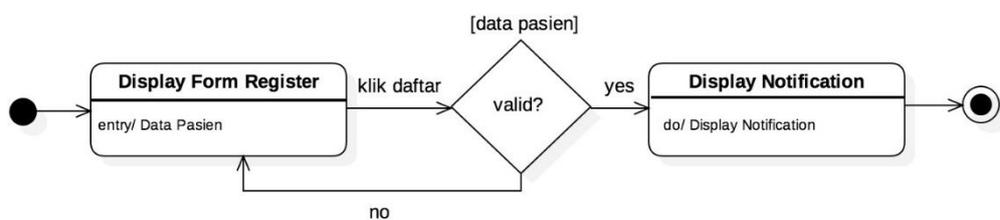
Statechart diagram setting data pasien menggambarkan transisi dan perubahan keadaan untuk pada saat me-*manage* informasi data pribadi pasien.



Gambar 3.27 *Statechart Diagram Setting Data Pasien*

3. *Statechart Diagram Register*

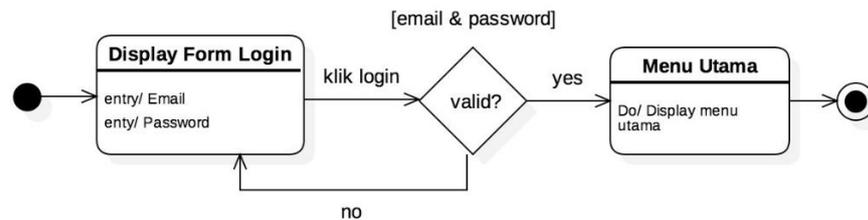
Statechart diagram register menggambarkan transisi dan perubahan keadaan pada saat pasien melakukan registrasi.



Gambar 3.28 *Statechart Diagram Register*

4. *Statechart Diagram Login*

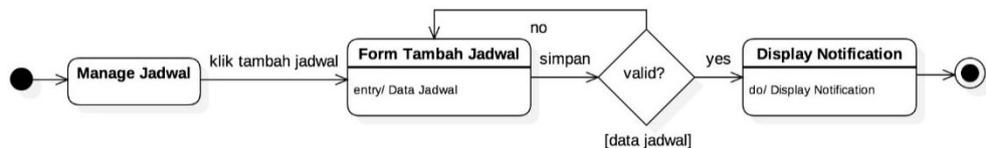
Statechart diagram login menggambarkan transisi dan perubahan keadaan pada saat pasien melakukan *login*.



Gambar 3.29 *Statechart Diagram Login*

5. *Statechart Diagram Tambah Data Jadwal*

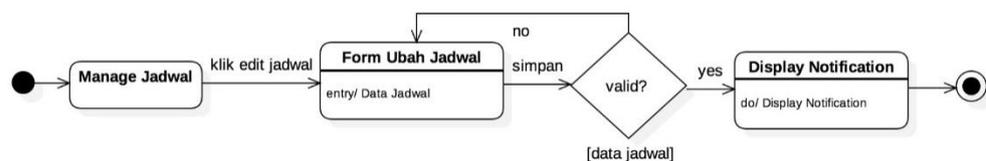
Statechart diagram tambah data jadwal menggambarkan transisi dan perubahan keadaan pada saat admin klinik menambahkan data jadwal.



Gambar 3.30 *Statechart Diagram Tambah Data Jadwal*

6. *Statechart Diagram Ubah Data Jadwal*

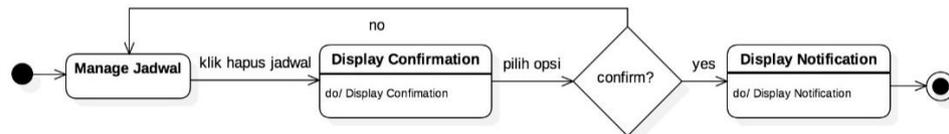
Statechart diagram ubah data jadwal menggambarkan transisi dan perubahan keadaan pada saat admin klinik mengubah data jadwal.



Gambar 3.31 *Statechart Diagram Ubah Data Jadwal*

7. *Statechart Diagram Hapus Data Jadwal*

Statechart diagram hapus data jadwal menggambarkan transisi dan perubahan keadaan pada saat admin klinik menghapus data jadwal.



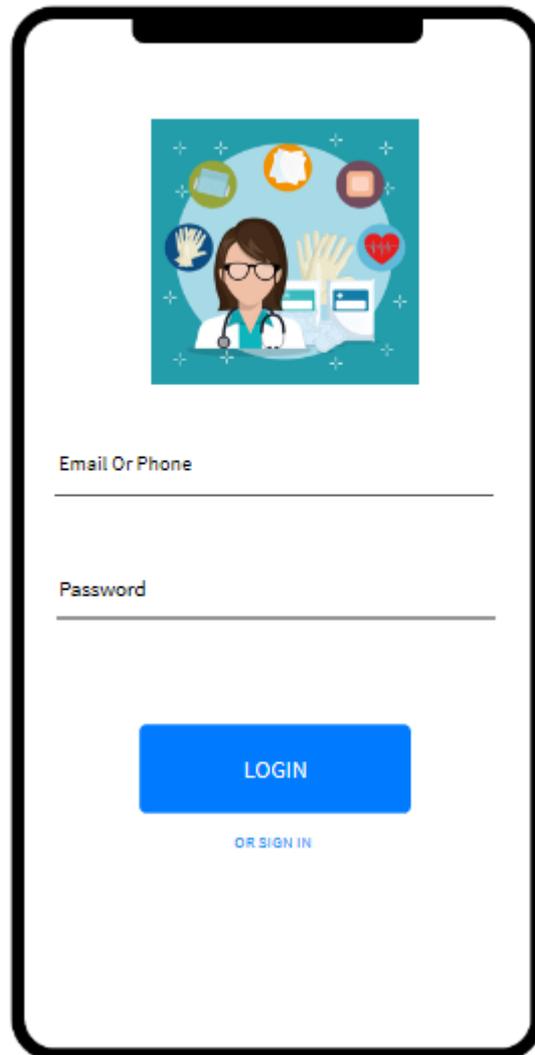
Gambar 3.32 *Statechart Diagram Hapus Data Jadwal*

3.2.2 Perancangan *Interface*

Setelah melakukan perancangan sistem perangkat lunak, tahap selanjutnya adalah melakukan analisis terhadap perancangan *interface* yang akan dibangun terdiri aplikasi *web base* dan *mobile android*.

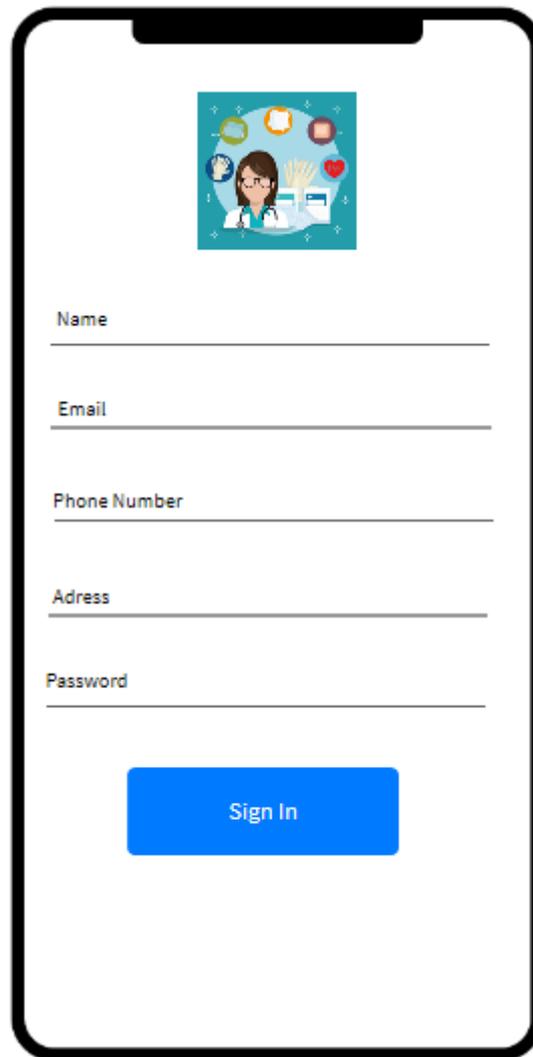
3.2.2.1 Perancangan *Interface Mobile*

1. Perancangan *Interface Login Pasien (mobile)*



Gambar 3.33 Perancangan *Interface Login Pasien*

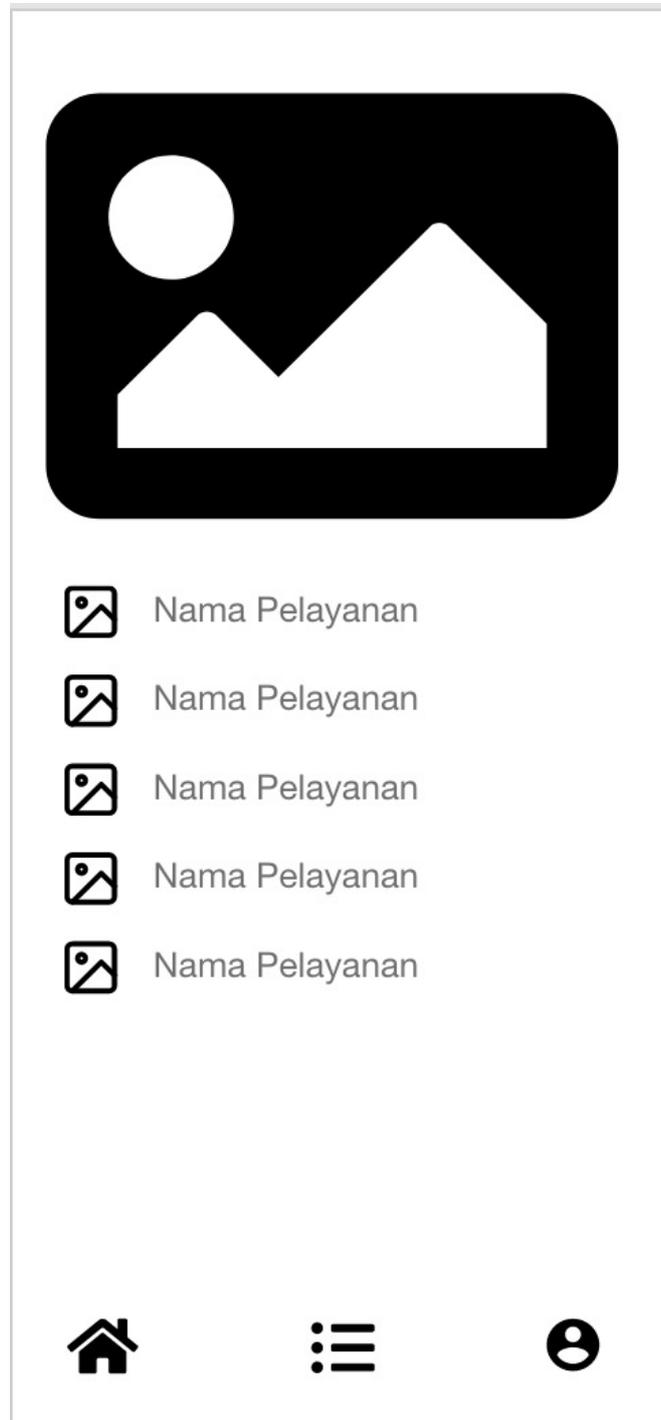
2. Perancangan *Interface Register Pasien (mobile)*



The image shows a mobile registration form interface. At the top, there is a circular icon featuring a female doctor with a stethoscope, surrounded by various medical symbols like a heart, a hand, a microscope, and a pill. Below the icon, there are five input fields, each with a label and a horizontal line for text entry: "Name", "Email", "Phone Number", "Adress", and "Password". At the bottom of the form, there is a prominent blue rectangular button with the text "Sign In" in white.

Gambar 3.34 Perancangan Register Pasien

3. Perancangan *Interface Dashboard (mobile)*



Gambar 3.35 Perancangan *Dashboard*

4. Perancangan *Interface* cari bidan (*mobile*)

Nama Pasien Alamat Tanggal, Waktu
  

Gambar 3.36 *List Order*

5. Perancangan *Interface* Buat apointment (*mobile*)

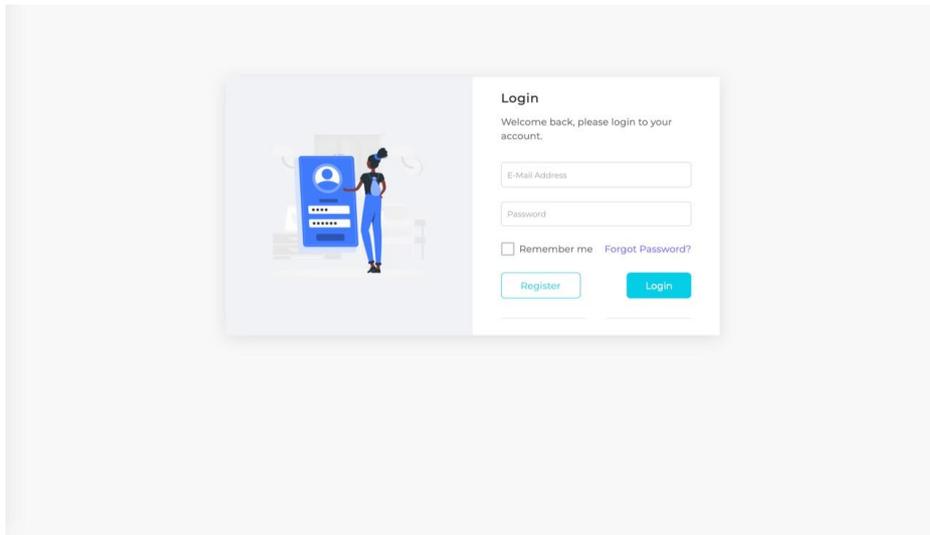
The image shows a mobile interface for creating an appointment. It consists of a vertical list of input fields, each with a label above it: 'Nama', 'Alamat', 'Tanggal', 'Waktu', 'Catatan', and 'Nama'. Each input field is a rounded rectangle. At the bottom of the screen, there is a navigation bar with three icons: a house icon, a hamburger menu icon, and a person profile icon.

Gambar 3.37 Perancangan Buat *Appointment*

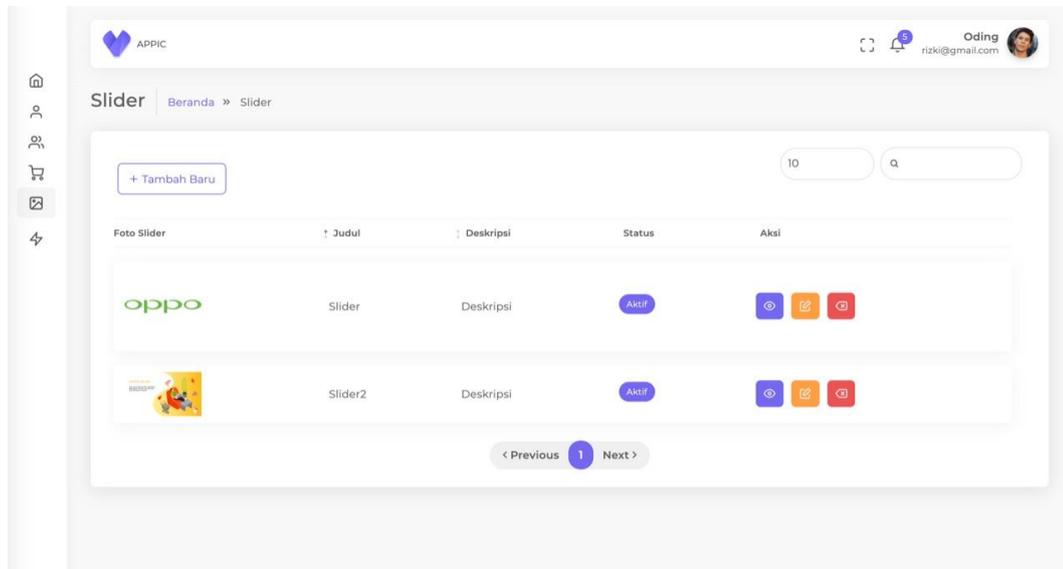
BAB IV

IMPLEMENTASI DAN UJI COBA

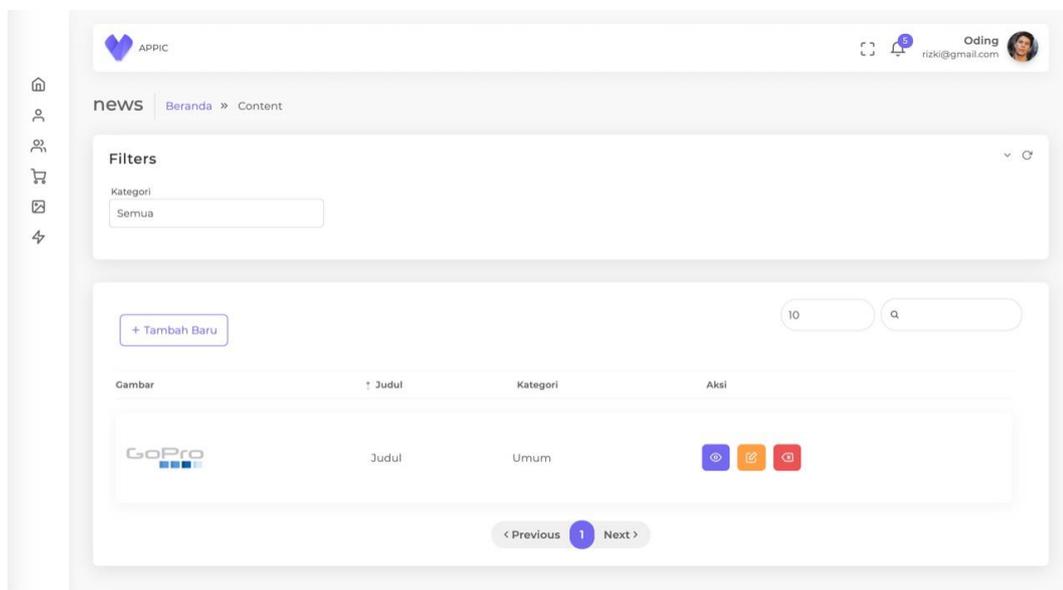
4.1. *User Interface*



Gambar 4.1 *Login Bidan*



Gambar 4.2 Halaman Slider



Gambar 4.3 Halaman Berita

User Beranda » Member

Filters

Jenis Kelamin: Semua Status: Semua

Aksi: + Tambah Baru

Nama	Email	No HP	JK	TTL	Aktif	Aksi
Rifa	rifa@gmail.com	0822403862829	L	Bandung 1994-10-10	Aktif	
Rima	rima@gmail.com	0822403862822	L	Bandung 1994-10-10	Aktif	
Dina	ema@gmail.com	081231232133	L	Majalengka 1994-06-06	Aktif	

Gambar 4.4 Halaman *User / Pasien*

Order Beranda » Order

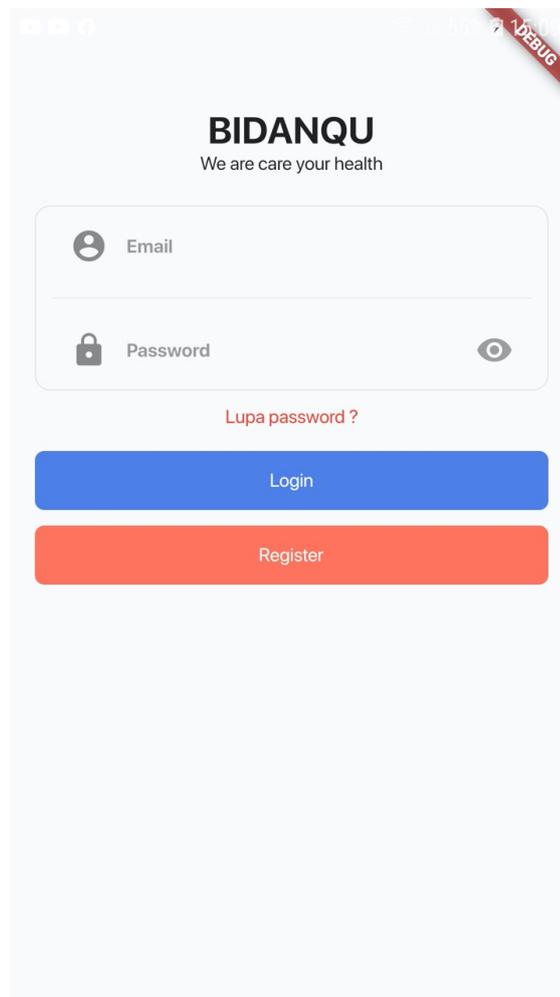
Aksi: + Tambah Baru

Pelayanan	Nama Pasien	Alamat	Tanggal	Waktu	Status	Aksi
Pelayanan KB	ema	Bandung	2020-10-10	10:10:10	Done	
Pertolongan Persalinan	Rifa	Bandung	2020-10-10	10:10:10	Pending	
Pelayanan kesehatan pada bayi dan balita	ema	Bandung	2020-10-10	10:10:10	Pending	
Pemeriksaan kehamilan	Rifa	Bandung	2020-10-10	10:10:10	Done	

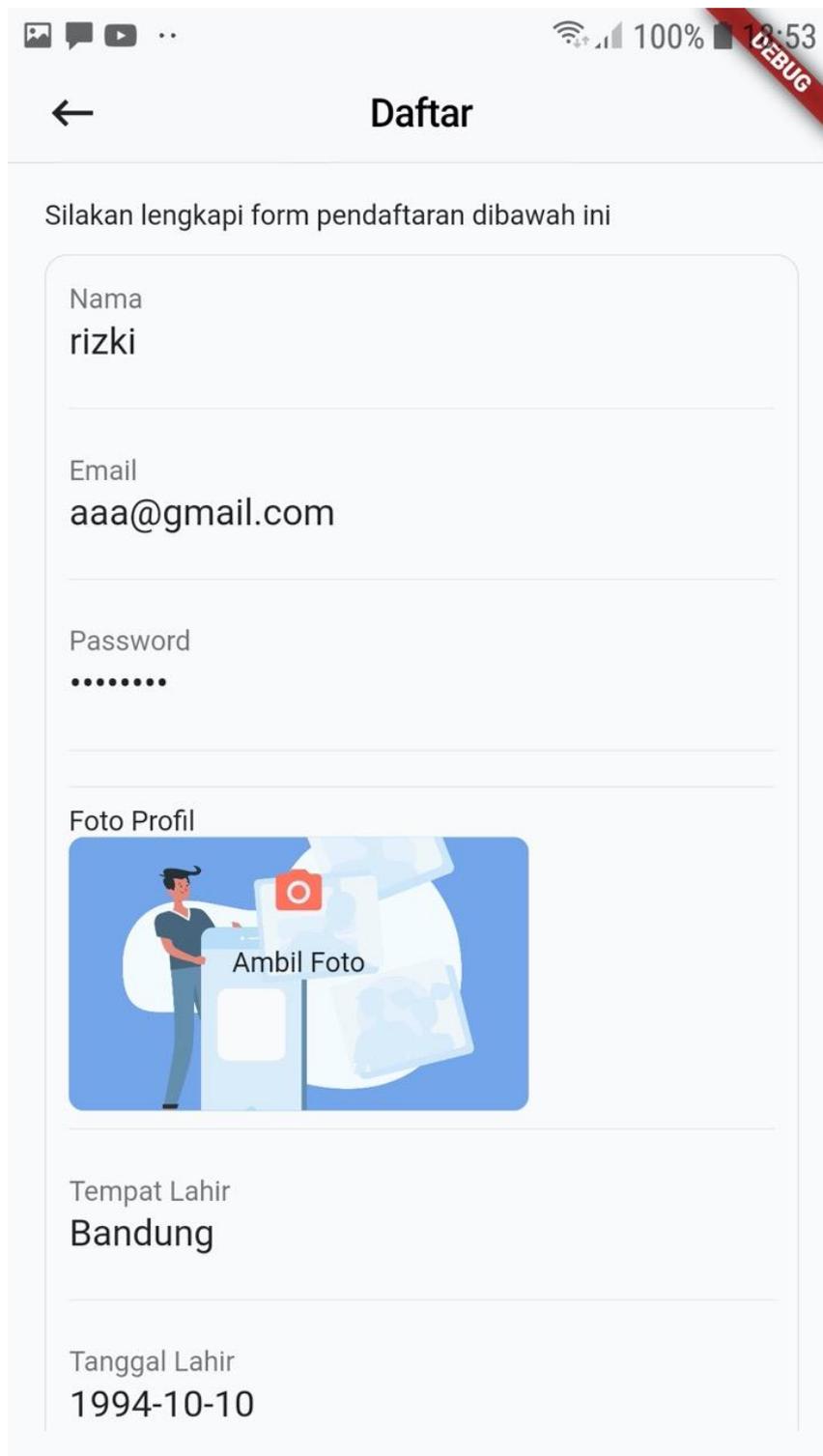
< Previous 1 Next >

COPYRIGHT © 2020 Pionir, All rights Reserved. Hand-crafted & Made with ❤️

Gambar 4.5 Halaman Order Pasien



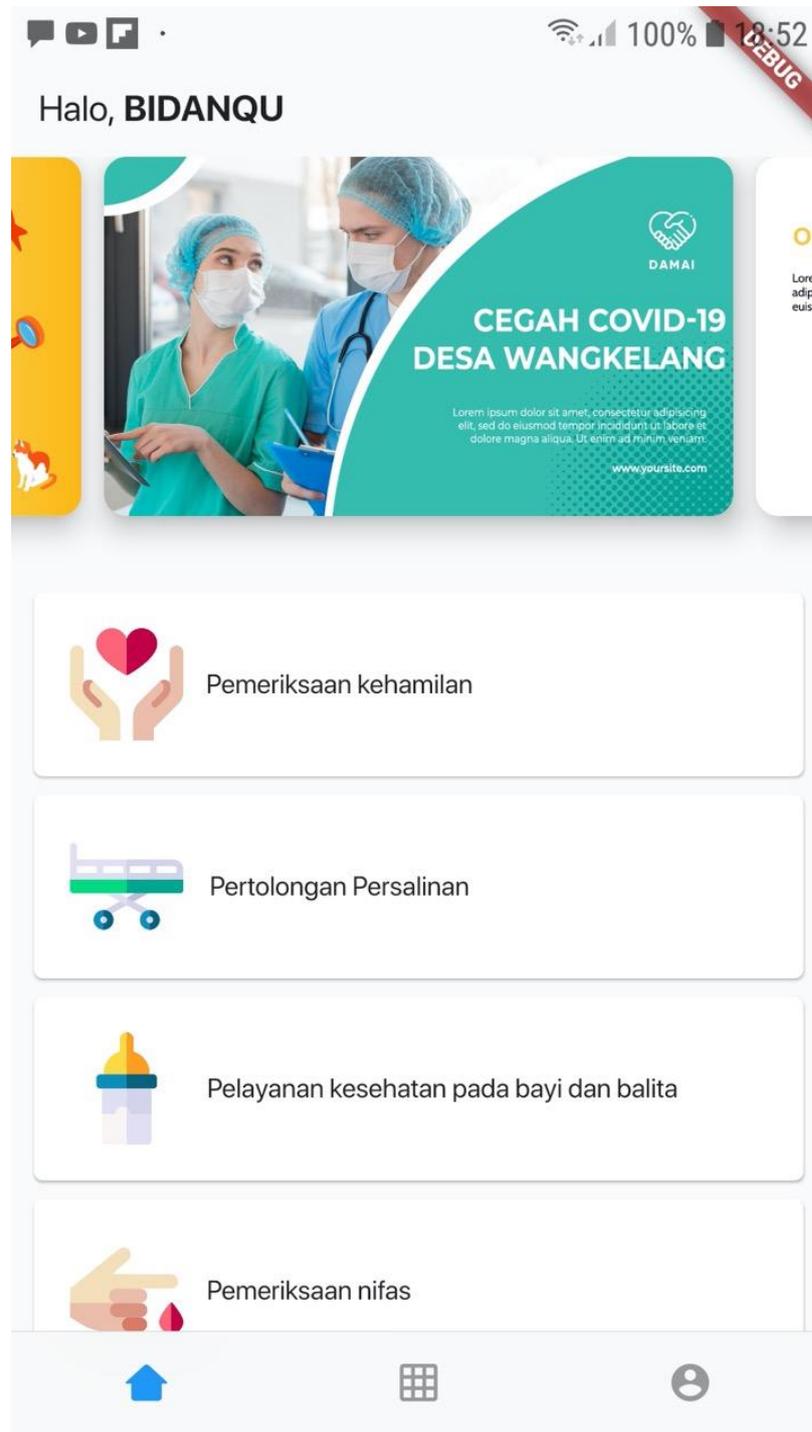
Gambar 4.6 Halaman *Login* Pasien



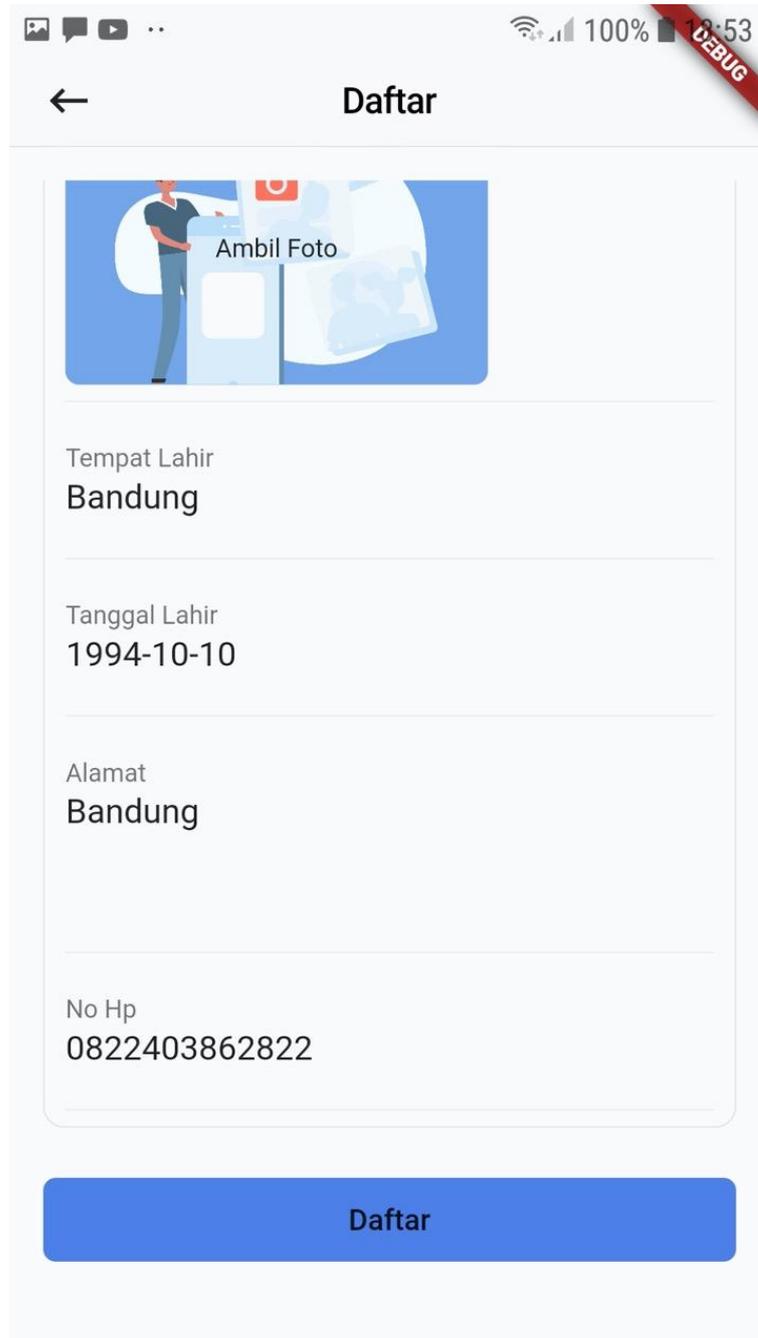
The image shows a mobile application interface for patient registration. At the top, there is a status bar with icons for camera, messages, and video, along with signal strength, 100% battery, and the time 10:53. A red 'DEBUG' banner is visible in the top right corner. Below the status bar is a navigation bar with a back arrow and the title 'Daftar'. The main content area contains the instruction 'Silakan lengkapi form pendaftaran dibawah ini' followed by a form with the following fields:

- Nama**: rizki
- Email**: aaa@gmail.com
- Password**: (masked)
- Foto Profil**: A blue button with an illustration of a person taking a photo and the text 'Ambil Foto'.
- Tempat Lahir**: Bandung
- Tanggal Lahir**: 1994-10-10

Gambar 4.7 Halaman Register Pasien



Gambar 4.8 Halaman *Dashboard*



The image shows a mobile application interface for a registration page. At the top, there is a status bar with icons for camera, messages, and video, along with signal strength, 100% battery, and the time 09:53. A red diagonal banner in the top right corner says 'DEBUG'. Below the status bar is a navigation bar with a back arrow and the title 'Daftar'. The main content area is a form with a blue header image containing a person at a photo booth and the text 'Ambil Foto'. The form fields are: 'Tempat Lahir' with the value 'Bandung', 'Tanggal Lahir' with the value '1994-10-10', 'Alamat' with the value 'Bandung', and 'No Hp' with the value '0822403862822'. At the bottom of the form is a large blue button labeled 'Daftar'.

← Daftar

Ambil Foto

Tempat Lahir
Bandung

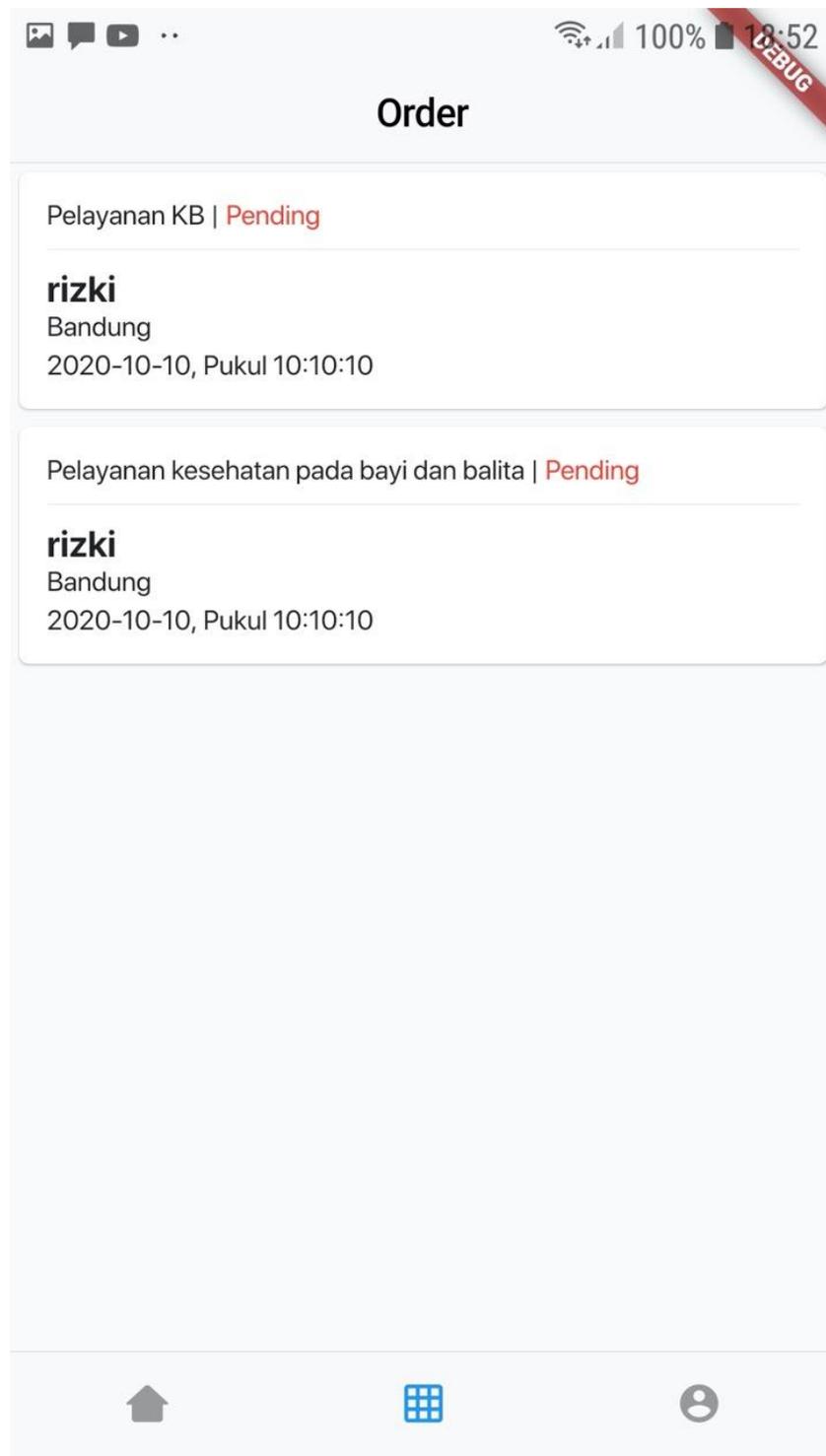
Tanggal Lahir
1994-10-10

Alamat
Bandung

No Hp
0822403862822

Daftar

Gambar 4.9 Halaman Register



Gambar 4.10 Halaman *List Order*

4.2 Pengujian Sistem

Tabel. 4.1 Test ID A.001 Halaman *User*

Test ID	Deskripsi Pengujian	Masukan	Hasil yang Diharapkan	Hasil yang Didapat
A.001.01	Menguji login pada sistem mobile	Obyek pengujian: mengetik email dan password dan mengklik tombol login	Muncul halaman utama aplikasi	Sesuai
A.001.02	Menguji Register / pendaftaran user	Mengetik pada kolom yang telah disediakan, kemudian menekan tombol daftar	User dapat melakukan pendaftaran	Sesuai
A.001.03	Menguji menu pelayanan	Menekan tombol menu pelayanan	Muncul halaman pelayanan sesuai dengan menu pelayanan yang diklik	Sesuai
A.001.04	Menguji pemesanan pelayanan	Mengetik pada kolom yang telah disediakan, kemudian menekan tombol pesan	Dapat memesan sesuai dengan menu pelayanan	Sesuai
A.001.04	Melihat history order	Menekan tombol menu order	Muncul tampilan history order pelayanan	Sesuai
A.001.05	Melihat profil	Menekan menu profil pada aplikasi	Muncul tampilan profile user	Sesuai
A.001.06	Menguji pada konten slider	Melihat tampilan slider	User dapat melihat konten slider	Sesuai
A.001.07	Menguji konten berita	Melihat tampilan berita pada aplikasi	User dapat melihat berita pada aplikasi	Sesuai

Tabel. 4.2 Test ID A.002 Halaman Admin

Test ID	Deskripsi Pengujian	Masukan	Hasil yang Diharapkan	Hasil yang Didapat
A.002.01	Menguji Login Pada Sistem Web admin	Obyek Pengujian: Mengetik Email dan Password dan mengklik tombol login	Muncul halaman utama aplikasi	Sesuai

A.002.02	Menguji order yang masuk	Mengklik tombol order dan menampilkan order yang masuk	Muncul tampilan order yang masuk	Sesuai
A.002.03	Menguji approval order	Mengklik tombol approve pada halamn order	Admin dapat menyetujui order	Sesuai
A.002.04	Menguji penambahan konten slider	Menambahkan konten slider	Slider dapat ditambahkan	Sesuai
A.002.05	Menguji penambahan konten berita	Menambahkan konten berita	Berita dapat ditambahkan	Sesuai

BAB V

PENUTUP

4.1. Simpulan

Berdasarkan hasil penelitian dan data temuan penelitian mengenai “Aplikasi Android Manajemen Pasien Di Praktik Mandiri Bidan (PMB)”, maka dapat disimpulkan bahwa sistem yang dibangun dapat berjalan dengan baik, dapat membantu untuk menangani pemesanan pelayanan bidan secara online, serta dapat membantu pasien dengan efektif dan efisien.

4.2. Saran

Berdasarkan penjelasan mengenai beberapa keterbatasan penelitian diatas maka terdapat saran untuk peneliti selanjutnya. Adapun saran bagi peneliti selanjutnya adalah sebagai berikut:

1. Bagi peneliti selanjutnya diharapkan dapat memesan bidan lebih dari satu, karena untuk saat ini pemesanan hanya dapat dilakukan untuk satu bidan.
2. Dapat menambahkan fitur chat bidan dan pasien

DAFTAR PUSTAKA

A.S., R., & Shalahudin, M. (2015). Rekayasa Perangkat Lunak dan Terstruktur dan Berorientasi Objek. In *Informatika Bandung*.

- Bustani, N. M., Rattu, A. J., & Saerang, J. S. M. (2015). Analisis Lama Waktu Tunggu Pelayanan Pasien Rawat Jalan Di Balai Kesehatan Mata Masyarakat Propinsi Sulawesi Utara. *Jurnal E-Biomedik*, 3(3). <https://doi.org/10.35790/ebm.3.3.2015.10456>
- Cerry, A., Rovita, N., Budiwa, S. D., & Suryan, A. (2017). Sistem Informasi Manajemen Kepegawaian (Studi Kasus : PT . Anugrah Kreasi Solusindo). 3, 329–336.
- Daimi, S. S., Rehman, U., & Maid, S. C. (2018). GSM Based Queue Management Device for OPD. 2864–2866.
- Esmael, H. R. (2015). Apply Android Studio (SDK) Tools. *International Journal of Advanced Research in Computer Science and Software Engineering*, 5(5), 88–93.
- Fathansyah. (2015). Basis Data. Revisi Kedua. In *Informatika, Bandung*. <https://doi.org/10.1136/jnnp.2006.100248>
- Gandhewar, N., & Sheikh, R. (2010). Google Android: An Emerging Software Platform For Mobile Devices. *International Journal on Computer Science and Engineering (IJCSE), Special Issue*, 6. <https://doi.org/10.1007/s003810050241>
- Ian Sommerville. (2011). *Software Engineering (Rekayasa Perangkat Lunak)* (6th ed.). Erlangga.
- Ikatan Bidan Indonesia. (2016). *Definisi Bidan*.
- Irsan, M. (2015). Rancang Bangun Aplikasi Mobile Notifikasi Berbasis Android Untuk Mendukung Kinerja Di Instansi Pemerintahan. *Jurnal Penelitian Teknik Informatika*, 1(1), 115–120. <http://jurnal.untan.ac.id/index.php/justin/article/view/9984/9752>
- Iswandy, E., Komputer, D. S. T. M. I., & Padang, S. J. (2015). Sistem Penunjang Keputusan Untuk Menentukan Dan Santunan Sosial Anak Nagari Dan Penyaluran Bagi Mahasiswa Dan Pelajar Kurang Mampu. *Jurnal TEKNOIF*, 3(2). <https://doi.org/2338-2724>
- Kemendes RI. (2016). *Modul Praktikum Konsep Kebidanan dan Etika Legal dalam Praktik Kebidanan*.
- Laelihyah, N., & Subekti, H. (2017). Waktu Tunggu Pelayanan Rawat Jalan dengan Kepuasan Pasien Terhadap Pelayanan di Rawat Jalan RSUD Kabupaten Indramayu. *Jurnal Kesehatan Vokasional*, 1(2), 102. <https://doi.org/10.22146/jkesvo.27576>
- Luthfi, F. (2017). Penggunaan Framework Laravel Dalam Rancang Bangun Modul Back-End Artikel Website Bisnisbisnis.ID. *JISKA (Jurnal Informatika Sunan Kalijaga)*, 2(1), 34. <https://doi.org/10.14421/jiska.2017.21-05>
- Marjuni, Kurniawan, E., Rizal, M., & Windarto, H. S. (2017). *Aplikasi Pendaftaran*

Online Di RSJD Dr. RM. Soedjarwadi Provinsi Jawa Tengah Berbasis Android.

Murtiwiyati, & Lauren, G. (2013). Rancang Bangun Aplikasi Pembelajaran Budaya Indonesia Untuk Anak Sekolah Dasar berbasis Android. *Jurnal Ilmiah*, 12, 2,3.

Octafian, D. T. (2015). Web Multi E-Commerce Berbasis Framework CodeIgniter. *Jurnal Teknologi Dan Informatika (TEKNOMATIKA)*.

PERATURAN MENTERI KESEHATAN REPUBLIK INDONESIA. (n.d.).

Pohan. (2015). Definisi Pasien. In *Kamus Kesehatan*.

Sommerville, I. (2011). Engenharia de Software. In *Pearson Brasil*.

Symey, Y., Sankaranarayanan, S., & Nurafifah, S. (2013). *hospital IOT*. 2(4).

Torry, T., Koeswo, M., & Sujianto, S. (2016). Faktor yang Mempengaruhi Waktu Tunggu Pelayanan Kesehatan kaitannya dengan Kepuasan Pasien Rawat Jalan Klinik penyakit dalam RSUD Dr. Iskak Tulungagung. *Jurnal Kedokteran Brawijaya*, 29(3), 252–257. <https://doi.org/10.21776/ub.jkb.2016.029.03.3>

Undang-Undang RI No 4. (2019). Undang - Undang RI No 4. *Undang-Undang Republik Indonesia Nomor 4 Tahun 2019, KEBIDANAN(004078)*. UNDANG-UNDANG REPUBLIK INDONESIA NOMOR 4 TAHUN 2019

Utara, U. S. (2013). *BAB 2 LANDASAN TEORI 2.1 Teori Umum Android® adalah sebuah kumpulan perangkat lunak untuk perangkat*. 7–110.