



**APLIKASI MONITORING KESEHATAN BALITA MELALUI
E-POSYANDU BERBASIS HYBRID PROGRAMMING**

Disusun dan Diajukan untuk Memenuhi Persyaratan Ujian Akhir Memperoleh
Gelar Sarjana Komputer pada Fakultas Sains dan Teknologi
Universitas Pembangunan Panca Budi
Medan

SKRIPSI

OLEH

NAMA : MUHAMMAD IKHSAN
NPM : 1614370644
PROGRAM STUDI : SISTEM KOMPUTER

**FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS PEMBANGUNAN PANCA BUDI
MEDAN
2020**

LEMBAR PENGESAHAN

**APLIKASI *MONITORING* KESEHATAN BALITA MELALUI
E-POSYANDU BERBASIS *HYBRID PROGRAMMING***

Disusun Oleh :

Nama : Muhammad Ikhlas

NPM : 1614370644

Program Studi : Sistem Komputer

Skripsi Telah Disetujui Oleh Dosen Pembimbing Skripsi

Pada Tanggal :

Dosen Pembimbing I



Khairul, S.Kom., M.Kom

Dosen Pembimbing II



Sri Wahyuni, S.Kom., M.Kom

Mengetahui,

Dekan Fakultas Sains Dan Teknologi



Hamdani, ST., M.T

Ketua Program Studi

Eko Hariyanto, S.Kom., M.Kom



DAFTAR ISI

ABSTRAK	i
KATA PENGANTAR.....	ii
DAFTAR ISI.....	iv
DAFTAR GAMBAR.....	vii
DAFTAR TABEL	xi
BAB II PENDAHULUAN	1
1.1 Latar Belakang Masalah	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah	3
1.4 Tujuan Penelitian	3
1.5 Manfaat Penelitian	3
BAB II LANDASAN TEORI	5
2.1 Posyandu	5
2.1.1 Tujuan Penyelenggaraan Posyandu	6
2.1.2 Sasaran Posyandu	6
2.1.3 Fungsi Posyandu	7
2.1.4 Kegiatan Posyandu	7
2.1.5 Pemantauan Kesehatan Anak	8
2.1.6 Imunisasi	8
2.1.7 Gizi	10
2.2 Aplikasi Berbasis <i>Hybrid</i>	11
2.3 <i>Progrresive Web Apps</i> (PWA)	13
2.4 HTML5 (<i>Hypertext Markup Language 5</i>)	15
2.5 CSS (<i>Cascading Style Sheet</i>)	16

2.6	<i>JavaScript</i>	17
2.7	<i>Bootstrap</i>	18
2.8	<i>Hypertext Processor (PHP)</i>	18
2.9	<i>MySQL</i>	19
2.10	<i>Xampp</i>	20
2.11	<i>Visual Studio Code</i>	21
2.12	<i>Adobe XD CC</i>	22
2.13	<i>Unified Modeling Language (UML)</i>	22
2.14	<i>Flowchart</i>	28
BAB III METODE PENELITIAN		31
3.1	Tahapan Penelitian	31
3.2	Metode Pengumpulan Data	35
3.3	Analisis Sistem Berjalan	35
3.4	Rancangan Penelitian	36
3.4.1	Analisis Sistem Usulan	37
3.5	Rancangan Secara Global	39
3.6	Rancangan Tampilan Antarmuka	52
BAB IV HASIL DAN PEMBAHASAN		75
4.1	Kebutuhan Spesifikasi Minimum <i>Hardware</i> dan <i>Software</i>	75
4.2	Hasil Tampilan Antarmuka	76
4.3	Pengujian Aplikasi dan Pembahasan	105
BAB V PENUTUP		111
5.1	Kesimpulan	111
5.2	Saran	111\

DAFTAR PUSTAKA

BIOGRAFI PENULIS

LAMPIRAN-LAMPIRAN

DAFTAR GAMBAR

Gambar 3.1 Tahapan Penelitian.....	31
Gambar 3.2 <i>Flow of Document</i> Sistem Usulan.....	38
Gambar 3.3 <i>Use Case Diagram</i> Pada Aplikasi E-Posyandu	40
Gambar 3.4 <i>Activity Diagram</i> Pada Admin Aplikasi E-Posyandu	42
Gambar 3.5 <i>Activity Diagram</i> Pada Bidan Aplikasi E-Posyandu.....	44
Gambar 3.6 <i>Activity Diagram</i> Pada Bidan Aplikasi E-Posyandu.....	46
Gambar 3.7 <i>Sequence Diagram</i> Pada Admin Aplikasi E-Posyandu	48
Gambar 3.8 <i>Sequence Diagram</i> Pada Bidan Aplikasi E-Posyandu.....	49
Gambar 3.9 <i>Sequence Diagram</i> Pada Ibu Bayi Aplikasi E-Posyandu.....	50
Gambar 3.10 <i>Class Diagram</i> Pada Aplikasi E-Posyandu.....	51
Gambar 3.11 Halaman Menu Utama Pada <i>Admin Di Website</i>	52
Gambar 3.12 Halaman Menu Utama Pada <i>Admin Di Mobile</i>	53
Gambar 3.13 Halaman Data Bidan Pada <i>Admin Pada Website</i>	53
Gambar 3.14 Halaman Data Bidan Pada <i>Admin Pada Mobile</i>	54
Gambar 3.15 Halaman Data Bayi Pada <i>Admin Di Website</i>	55
Gambar 3.16 Halaman Data Bayi Pada <i>Admin Di Mobile</i>	55
Gambar 3.17 Halaman Data Artikel Pada <i>Admin Di Website</i>	56
Gambar 3.18 Halaman Data Artikel Pada <i>Admin Di Mobile</i>	57
Gambar 3.19 Halaman Jadwal Vaksin Pada <i>Admin Di Website</i>	58
Gambar 3.20 Halaman Jadwal Vaksin Pada <i>Admin Di Mobile</i>	58
Gambar 3.21 Halaman Menu Utama Pada Bidan Di <i>Website</i>	59
Gambar 3.22 Halaman Menu Utama Pada Bidan Di <i>Mobile</i>	60
Gambar 3.23 Halaman Data Bayi Pada Bidan Di <i>Website</i>	61
Gambar 3.24 Halaman Data Bayi Pada Bidan Di <i>Mobile</i>	61
Gambar 3.25 Halaman Jadwal Vaksin Pada Bidan Di <i>Website</i>	62
Gambar 3.26 Halaman Jadwal Vaksin Pada Bidan Di <i>Mobile</i>	63

Gambar 3.27 Halaman Artikel Kesehatan Pada Bidan Di <i>Website</i>	64
Gambar 3.28 Halaman Artikel Kesehatan Pada Bidan Di <i>Mobile</i>	64
Gambar 3.29 Halaman Menu Utama Pada Ibu Bayi Di <i>Website</i>	65
Gambar 3.30 Halaman Menu Utama Pada Ibu Bayi Di <i>Mobile</i>	66
Gambar 3.31 Halaman Data Bayi Pada Ibu Bayi Di <i>Website</i>	67
Gambar 3.32 Halaman Data Bayi Pada Ibu Bayi Di <i>Mobile</i>	67
Gambar 3.33 Halaman Statistik Berat Badan Bayi Pada Ibu Bayi Di <i>Website</i>	68
Gambar 3.34 Halaman Statistik Berat Badan Bayi Pada Ibu Bayi Di <i>Mobile</i>	69
Gambar 3.35 Halaman Jadwal Vaksin Pada Ibu Bayi Di <i>Website</i>	70
Gambar 3.36 Halaman Jadwal Vaksin Pada Ibu Bayi Di <i>Mobile</i>	70
Gambar 3.37 Halaman Artikel Kesehatan Pada Ibu Bayi Di <i>Website</i>	71
Gambar 3.38 Halaman Artikel Kesehatan Pada Ibu Bayi Di <i>Mobile</i>	72
Gambar 3.39 Halaman <i>Profile</i> Bayi Pada Ibu Bayi Di <i>Website</i>	73
Gambar 3.40 Halaman <i>Profile</i> Bayi Pada Ibu Bayi Di <i>Mobile</i>	73
Gambar 4.1 Tampilan Menu Utama Pada <i>Admin</i> Di <i>Website</i>	76
Gambar 4.2 Tampilan Menu Utama Pada <i>Admin</i> Di <i>Mobile</i>	77
Gambar 4.3 Tampilan Data Bidan Pada <i>Admin</i> Di <i>Website</i>	77
Gambar 4.4 Tampilan Data Bidan Pada <i>Admin</i> Di <i>Mobile</i>	78
Gambar 4.5 Tampilan Tambah Data Bidan Pada <i>Admin</i> Di <i>Website</i>	78
Gambar 4.6 Tampilan Edit Data Bidan Pada <i>Admin</i> Di <i>Website</i>	79
Gambar 4.7 Tampilan Tambah dan Edit Data Bidan Pada <i>Admin</i> Di <i>Mobile</i>	79
Gambar 4.8 Tampilan Data Bayi Pada <i>Admin</i> Di <i>Website</i>	80
Gambar 4.9 Tampilan Data Bayi Pada <i>Admin</i> Di <i>Mobile</i>	80
Gambar 4.10 Tampilan Tambah Data Bayi Pada <i>Admin</i> Di <i>Website</i>	81
Gambar 4.11 Tampilan Edit Data Bayi Pada <i>Admin</i> Di <i>Website</i>	82
Gambar 4.12 Tampilan Tambah Data Bayi Pada <i>Admin</i> Di <i>Mobile</i>	82
Gambar 4.13 Tampilan Edit Data Bayi Pada <i>Admin</i> Di <i>Mobile</i>	83
Gambar 4.14 Tampilan Data Artikel Pada <i>Admin</i> Di <i>Website</i>	84

Gambar 4.15 Tampilan Data Artikel Pada <i>Admin Di Mobile</i>	84
Gambar 4.16 Tampilan Tambah Data Artikel Pada <i>Admin Di Website</i>	85
Gambar 4.17 Tampilan Tambah Data Artikel Pada <i>Admin Di Mobile</i>	85
Gambar 4.18 Tampilan Edit Data Artikel Pada <i>Admin Di Website</i>	86
Gambar 4.19 Tampilan Edit Data Artikel Pada <i>Admin Di Mobile</i>	86
Gambar 4.20 Tampilan Jadwal Vaksin Pada <i>Admin Di Website</i>	87
Gambar 4.21 Tampilan Jadwal Vaksin Pada <i>Admin Di Mobile</i>	88
Gambar 4.22 Tampilan Menu Utama Pada Bidan Di <i>Website</i>	88
Gambar 4.23 Tampilan Menu Utama Pada Bidan Di <i>Mobile</i>	89
Gambar 4.24 Tampilan Data Bayi Pada Bidan Di <i>Website</i>	89
Gambar 4.25 Tampilan Data Bayi Pada Bidan Di <i>Mobile</i>	90
Gambar 4.26 Tampilan Menu Utama Pada Bidan Di <i>Website</i>	90
Gambar 4.27 Tampilan Menu Utama Pada Bidan Di <i>Mobile</i>	91
Gambar 4.28 Tampilan Data Bayi Pada Bidan Di <i>Website</i>	91
Gambar 4.29 Tampilan Data Bayi Pada Bidan Di <i>Mobile</i>	92
Gambar 4.30 Tampilan Tambah Data Bayi Pada Bidan Di <i>Website</i>	93
Gambar 4.31 Tampilan Tambah Data Bayi Pada Bidan Di <i>Mobile</i>	93
Gambar 4.32 Tampilan Edit Data Bayi Pada Bidan Di <i>Website</i>	94
Gambar 4.33 Tampilan Edit Data Bayi Pada Bidan Di <i>Mobile</i>	94
Gambar 4.34 Tampilan Data Orang Tua Bayi Pada Bidan Di <i>Website</i>	95
Gambar 4.35 Tampilan Data Orang Tua Bayi Pada Bidan Di <i>Mobile</i>	95
Gambar 4.36 Tampilan Edit Data Orang Tua Bayi Pada Bidan Di <i>Website</i>	96
Gambar 3.37 Tampilan Edit Data Orang Tua Bayi Pada Bidan Di <i>Mobile</i>	96
Gambar 3.38 Tampilan Jadwal Vaksin Pada Bidan Di <i>Website</i>	97
Gambar 3.39 Tampilan Jadwal Vaksin Pada Bidan Di <i>Mobile</i>	97
Gambar 3.40 Tampilan Artikel Kesehatan Pada Bidan Di <i>Website</i>	98
Gambar 4.41 Tampilan Artikel Kesehatan Pada Bidan Di <i>Mobile</i>	98
Gambar 4.42 Tampilan Menu Utama Pada Ibu Bayi Di <i>Website</i>	99

Gambar 4.43 Tampilan Menu Utama Pada Ibu Bayi Di <i>Mobile</i>	99
Gambar 4.44 Tampilan Data Bayi Pada Ibu Bayi Di <i>Website</i>	100
Gambar 4.45 Tampilan Data Bayi Pada Ibu Bayi Di <i>Mobile</i>	100
Gambar 4.46 Tampilan Statistik Berat Badan Bayi Pada Ibu Bayi Di <i>Website</i>	101
Gambar 4.47 Tampilan Statistik Berat Badan Bayi Pada Ibu Bayi Di <i>Mobile</i>	101
Gambar 4.48 Tampilan Jadwal Vaksin Pada Ibu Bayi Di <i>Website</i>	102
Gambar 4.49 Tampilan Jadwal Vaksin Pada Ibu Bayi Di <i>Mobile</i>	102
Gambar 4.50 Tampilan Artikel Kesehatan Pada Ibu Bayi Di <i>Website</i>	103
Gambar 4.51 Tampilan Artikel Kesehatan Pada Ibu Bayi Di <i>Mobile</i>	103
Gambar 4.52 Tampilan Profile Bayi Pada Ibu Bayi Di <i>Website</i>	104
Gambar 4.53 Tampilan Profile Bayi Pada Ibu Bayi Di <i>Mobile</i>	104
Gambar 4.54 Hasil Input Data Bidan Ke Database Oleh <i>Admin</i>	105
Gambar 4.55 Hasil Input Data Ibu Bayi Ke Database Oleh <i>Admin</i>	106
Gambar 4.56 Hasil Input Data Ibu Bayi Ke Database Oleh Bidan.....	107
Gambar 4.57 Tampilan Form Login Users Pada <i>Website</i>	108
Gambar 4.58 Tampilan Form Login Users Pada <i>Mobile</i>	108
Gambar 4.59 Tampilan Data Bayi Pada <i>Website</i>	109
Gambar 4.60 Tampilan Data Bayi Pada <i>Mobile</i>	110
Gambar 4.61 Tampilan Data Password Bayi	110

DAFTAR TABEL

Tabel 2.1 Simbol <i>Use Case Diagram</i>	23
Tabel 2.2 Simbol <i>Sequence Diagram</i>	25
Tabel 2.3 Simbol <i>Activity Diagram</i>	26
Tabel 2.4 Simbol <i>Class Diagram</i>	28
Tabel 2.5 Simbol <i>Flowchart</i>	29

KATA PENGANTAR

Syukur Alhamdulillah kepada Allah SWT, karena berkat rahmat dan karunianya saya masih diberi kesehatan sampai saat ini sehingga penulis dapat menyelesaikan tugas akhir ini dengan baik, Shalawat dan salam senantiasa disampaikan kepada Rasulullah SAW yang menghantar manusia dari jaman yang gelap sampai jaman yang terang seperti saat ini. Penyusunan skripsi ini bertujuan untuk memenuhi syarat-syarat guna mencapai gelar Sarjana Komputer pada Fakultas Sains dan Teknologi di Universitas Pembangunan Panca Budi.

Tugas akhir ini disusun berdasarkan hasil penelitian yang dilaksanakan pada 18 Desember 2019 dengan judul “**APLIKASI *MONITORING KESEHATAN BALITA MELALUI E-POSYANDU BERBASIS *HYBRID PROGRAMMING****”.

Dalam kesempatan ini penulis mengucapkan terima kasih yang sebesar- besarnya kepada pihak yang telah membantu dalam penyelesaian penyusunan Tugas Akhir. Penulis mengucapkan terima kasih kepada:

1. Kepada ortang tua saya yang telah memberikan dukungan semangat dan menyelesaikan tugas akhir ini, dan penulis berharap semoga penyusunan skripsi ini dapat bermanfaat bagi pembaca.
2. Rektor Universitas Pembangunan Panca Budi, Bapak DR. H. Muhammad Isa Indrawan, SE, MM.
3. Rektor I, Bapak Ir. Bhakti Alamsyah, M.T., Ph.D.
4. Dekan Fakultas Sains dan Teknologi, Bapak Hamdani, ST., M.T

5. Ketua Program Studi Sistem Komputer, Bapak Eko Hariyanto, S.Kom.,
M.Kom
6. Dosen Pembimbing I, Bapak Khairul, S.Kom., M.Kom
7. Dosen Pembimbing II, Ibu Sri Wahyuni, S.Kom., M.Kom
8. Bapak/Ibu dosen beserta seluruh staf Universitas Pembangunan Panca Budi
Medan yang telah mendidik dan membimbing penulis selama mengikuti
perkuliahan.
9. Seluruh teman-teman penulis dari program studi Sistem Komputer,
Fakultas Sains dan Teknologi, Universitas Pembangunan Panca Budi,
Medan

Medan, 07 Maret 2020

Penulis,

Muhammad Ikhsan

1614370644

SURAT PERNYATAAN

Saya yang bertanda tangan dibawah ini:

Nama : Muhammad Ikhsan
NPM : 1614370644
Program Studi : Sistem Komputer
Jenjang : S-1 (Strata Satu)
Judul Skripsi : Aplikasi Monitoring Kesehatan Balita Melalui E-Posyandu Berbasis Hybrid Programming

Dengan ini menyatakan bahwa:

1. Skripsi ini merupakan hasil karya tulis saya sendiri dan bukan merupakan hasil karya orang lain.
2. Memberikan izin hak bebas Royalti Non-Eksklusif kepada Unpub untuk menyimpan, mengalih-media/formatkan, mengelola, mendistribusikan, dan mempublikasikan karya skripsi saya melalui internet atau media lain bagi kepentingan akademis.

Pernyataan ini saya buat dengan penuh tanggung jawab dan saya bersedia menerima konsekuensi apa pun sesuai dengan aturan yang berlaku apabila dikemudian hari diketahui bahwa pernyataan ini tidak benar.

Medan, 16 Desember 2020



(Muhammad Ikhsan)

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Posyandu merupakan kegiatan swadaya masyarakat yang bertujuan untuk memelihara kesehatan masyarakat khususnya para ibu hamil dan balita (Firda Dini Yulianti dan Yustina Meisella Kristania : 2019). Namun pada saat ini pelayanan posyandu mengalami banyak hambatan, salah satu masalahnya adalah pengolahan data tentang kegiatan-kegiatan yang sedang berlangsung di posyandu seputar laporan kunjungan imunisasi pada balita, peningkatan gizi, dan timbang berat badan yang masih manual sehingga pengolahan data laporan kegiatan lebih lama, banyak data yang berserakan serta data sering hilang seperti data-data kegiatan posyandu.

Pada saat ini ketika ibu bayi ingin melihat data bayi pada posyandu harus melewati beberapa tahapan sehingga hal proses ini menjadi tidak praktis dengan adanya sistem yang akan dibangun diharapkan menjadi solusi dalam proses melihat data. Dalam hal lain seperti pengolahan data pada ibu bayi yang dilakukan oleh bidan juga masih melewati beberapa tahapan hal ini membuat bidan dalam pengolahan data menjadi tidak efektif dengan adanya sistem yang akan dibangun proses ini nantinya akan menjadi efektif. Pada hasil kegiatan posyandu seperti *monitoring* kesehatan balita saat ini masih kurang efisien di karenakan masih manual dengan adanya sistem nantinya ketika ibu bayi ingin melihat dan hasil kegiatan pada posyandu nantinya dapat diakses pada sistem hal ini akan menjadi efisien.

Berdasarkan permasalahan tersebut, maka perlu dibangun suatu sistem yang dapat membantu seperti pengelolah data dan *monitoring* kesehatan balita sebagai solusi yang tepat untuk menyelesaikan berbagai masalah ini. Maka diperlukan aplikasi *monitoring* kesehatan balita melalui e-posyandu dengan adanya ini dapat memberikan kemudahan petugas posyandu dalam pengelolaan data maupun *monitoring* kesehatan balita yang sedang berlangsung pada posyandu serta dalam pembuatan laporan kegiatan. Banyak sekali perkembangan teknologi yang telah dikembangkan oleh para developer dunia dan khususnya berbasis *web*. Salah satu contohnya adalah *Progressive Web Apps* (PWA) yang diperkirakan mampu bersaing dengan aplikasi *native*. Aplikasi *native* merupakan suatu aplikasi yang berjalan pada perangkat *mobile* dan dapat di akses melalui *icon* yang ada pada layar perangkat tersebut aplikasi biasanya dikembangkan secara khusus untuk satu *platform*. Dengan adanya perkembangan teknologi *Progressive Web Apps* (PWA) dapat menjadi solusi dalam membangun aplikasi *monitoring* kesehatan balita melalui e-posyandu berbasis *hybrid programming* (Wijayanti Nurul Khotimah, Laurensius Adi dan Rizky Januar Akbar, 2017).

1.2 Rumusan Masalah

Bedasarkan latar belakang yang telah diuraikan sebelumnya, dapat dirumuskan suatu masalah yaitu sebagai berikut :

1. Bagaimana membuat aplikasi *monitoring* kesehatan balita melalui e-posyandu berbasis *Progressive Web Apps* (PWA).
2. Bagaimana menerapkan *hybrid programming* pada aplikasi *monitoring* kesehatan balita melalui e-posyandu berbasis *hybrid programming*.

1.3 Batasan Masalah

Adapun batasan-batasan masalah dalam penelitian ini :

1. Penerapan teknologi *Progressive Web Apps* pada aplikasi *monitoring* kesehatan balita pada e-posyandu.
2. Aplikasi yang dikembangkan adalah aplikasi berbasis *hybrid* yaitu aplikasi yang dapat dijalankan di *website* maupun *smartphone* yang berhubungan dengan layanan kesehatan balita di posyandu.
3. Aplikasi ini dikembangkan untuk menampilkan status kesehatan balita yang dilaksanakan di posyandu setiap bulan.

1.4 Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah sebagai berikut :

1. Merancang aplikasi *monitoring* kesehatan balita melalui e-posyandu berbasis *Progressive Web Apps* (PWA).
2. Menerapkan *hybrid programming* pada aplikasi *monitoring* kesehatan balita melalui e-posyandu.
3. Merancang antarmuka aplikasi *monitoring* kesehatan balita melalui e-posyandu.

1.5 Manfaat Penelitian

Adapun manfaat dari penelitian ini adalah sebagai berikut :

1. Dapat menghasilkan aplikasi *monitoring* kesehatan balita melalui e-posyandu berbasis *Progressive Web Apps* (PWA).

2. Dapat menerapkan *hybrid programming* pada aplikasi *monitoring* kesehatan balita melalui e-posyandu
3. Menyediakan layanan informasi kesehatan balita melalaui e-posyandu yang dapat diakses melalui *website* maupun *smartphone*.
4. Mempermudah dalam pengolahan data pelayanan yang ada di posyandu.
5. Mempermudah orang tua dalam mengetahui status kesehatan balita.

BAB II

LANDASAN TEORI

2.1 Posyandu

Pos Pelayanan Terpadu (Posyandu) merupakan salah satu bentuk Upaya Kesehatan Bersumber Daya Masyarakat (UKBM) yang keberadaannya dikelola dan diselenggarakan berasal dari, oleh dan untuk masyarakat dalam penyelenggaraan guna memberdayakan masyarakat dan memberikan kemudahan kepada masyarakat dalam memperoleh pelayanan kesehatan dasar/sosial dasar untuk mempercepat penurunan Angka Kematian Ibu dan Bayi (Encang Saepudin, Edwin Rizal, dan Agus Rusman : 2017).

Posyandu merupakan salah satu upaya kesehatan bersumber daya masyarakat yang melibatkan partisipasi masyarakat dalam pelaksanaannya. Posyandu juga sebagai perpanjangan tangan puskesmas memberikan pelayanan dan pemantauan kesehatan yang dilaksanakan secara terpadu. Kegiatan dilaksanakan oleh kader kesehatan yang telah mendapatkan pendidikan dan pelatihan dari puskesmas mengenai pelayanan kesehatan dasar (Ruyatul Hasanah dan Suhat : 2014).

Jadi dapat disimpulkan bahwa, posyandu merupakan salah satu peran penting dalam masyarakat serta dalam pengembangan kesehatan masyarakat dengan membentuk kemampuan untuk hidup sehat bagi setiap penduduk dalam membentuk kesehatan yang ideal.

2.1.1 Tujuan Penyelenggaraan Posyandu

Adapun tujuan dari penyelenggaraan posyandu ialah :

1. Menurunkan angka kematian ibu dan anak.
2. Dsa Meningkatkan pelayanan kesehatan ibu untuk menurunkan angka kematian bayi.
3. Mempercepat penerimaan norma keluarga kecil bahagia sejahtera (NKKBS) atau norma keluarga kecil bahagia dan sejahtera.
4. Meningkatkan kemampuan masyarakat untuk mengembangkan kegiatan kesehatan dan menunjang peningkatan hidup sehat.
5. Pendekatan dan pemerataan pelayanan kesehatan kepada masyarakat sehingga tercapai peningkatan cakupan pelayanan kesehatan.
6. Meningkatkan dan membina peran serta masyarakat dalam rangka alih teknologi untuk usaha kesehatan masyarakat.

2.1.2 Sasaran Posyandu

Sasaran posyandu adalah seluruh masyarakat dan utamanya adalah sebagai berikut :

1. Bayi usia kurang dari 1 tahun.
2. Anak balita usia 1 sampai 5 tahun.
3. Ibu hamil, ibu melahirkan, ibu nifas dan ibu menyusui wanita usia subur (WUS).

2.1.3 Fungsi Posyandu

Adapun fungsi dari posyandu adalah sebagai berikut :

1. Sebagai wadah pemberdayaan masyarakat dalam alih informasi dan keterampilan dari petugas kepada masyarakat dan antar sesama masyarakat dalam rangka mempercepat penurunan angka kematian ibu dan angka kematian bayi.
2. Sebagai wadah untuk mendekatkan pelayanan kesehatan dasar terutama berkaitan dengan penurunan angka kematian ibu dan angka kematian bayi.

2.1.4 Kegiatan Posyandu

Posyandu dilaksanakan setiap sebulan sekali, untuk tanggal dan waktunya ditentukan oleh petugas kesehatan dari posyandu. Pelaksanaan posyandu dilakukan dengan sistem 5 meja yaitu (Sunarto Kadir : 2017) :

1. Meja 1 : Pendaftaran.
2. Meja 2 : Penimbangan.
3. Meja 3 : Pengisian Kartu Menuju Sehat (KMS).
4. Meja 4 : Penyuluhan perorangan.
5. Meja 5 : Pelayanan Keluarga Berencana (KB) & Kesehatan (imunisasi, pemberian vitamin A tiap bulan Februari dan Agustus, pembagian pil atau kondom, pengobatan ringan, konsultasi kesehatan).

Petugas pada meja 1 sampai dengan dengan 4 dilaksanakan oleh kader posyandu, sedangkan meja 5 dilaksanakan oleh petugas kesehatan posyandu.

2.1.5 Pemantauan Kesehatan Anak

Pemantauan kesehatan anak di posyandu lebih ditujukan untuk memantau pertumbuhan yaitu suatu kegiatan yang dilakukan secara terus menerus dan teratur untuk mengidentifikasi secara dini bila ada gangguan keseimbangan gizi pada anak. Pemantauan pertumbuhan ini menjadi salah satu kegiatan penting dalam rangka kewaspadaan gizi atau sering disebut dengan surveilans gizi.

2.1.6 Imunisasi

Imunisasi adalah suatu proses untuk membuat seseorang imun atau kebal terhadap suatu penyakit. Proses ini dilakukan dengan pemberian vaksin yang menambahkan sistem kekebalan tubuh agar kebal terhadap penyakit tersebut.

Pada saat sekarang ini, vaksin dapat digunakan dalam pencegahan penyakit telah banyak beredar dan hasil daya lindung yang dihasilkan juga telah terbukti bermanfaat. Sebagai salah satu contoh adalah keberhasilan dalam menghilangkan penyakit cacar.

Imunisasi bertujuan untuk membangun kekebalan tubuh seseorang terhadap suatu penyakit, dengan membentuk antibodi dalam kadar tertentu. Agar antibodi terbentuk, seseorang harus diberikan

vaksin sesuai jadwal waktu yang telah ditentukan. Jadwal imunisasi tergantung pada jenis penyakit yang akan dicegah. Sejumlah vaksin cukup diberikan satu kali saja, tetapi ada juga yang harus diberikan beberapa kali, dan diulang pada usia tertentu. Vaksin dapat diberikan dengan cara disuntik atau tetes mulut.

Program imunisasi merupakan salah satu upaya untuk melindungi penduduk terhadap penyakit tertentu. Program imunisasi diberikan kepada populasi yang dianggap rentan terjangkit penyakit menular, yaitu bayi, anak usia sekolah, wanita usia subur, dan ibu hamil. Setiap bayi wajib mendapatkan lima imunisasi dasar lengkap (LIL) yang terdiri dari 1 dosis BCG, 3 dosis DPT, 4 dosis polio, 3 dosis hepatitis B, dan 1 dosis campak. Dari kelima imunisasi dasar lengkap yang diwajibkan tersebut, campak merupakan imunisasi yang mendapat perhatian lebih yang dibuktikan dengan komitmen Indonesia pada lingkup ASEAN dan SEARO untuk mempertahankan cakupan imunisasi campak sebesar 90%. Hal ini terkait dengan realita bahwa campak adalah salah satu penyebab utama kematian pada balita. Pencegahan campak memiliki peran signifikan dalam penurunan angka kematian balita.

Imunisasi pada bayi mengharapkan agar setiap bayi mendapatkan kelima jenis imunisasi dasar lengkap. Keberhasilan seorang bayi dalam mendapatkan 5 jenis imunisasi dasar tersebut diukur melalui indikator imunisasi dasar lengkap (Vivi Triana : 2015).

2.1.7 Gizi

Gizi yang baik adalah salah satu unsur penting untuk mewujudkan manusia yang berkualitas. Pemenuhan gizi anak harus diperhatikan terutama pada balita karena pada masa ini pertumbuhan mengalami peningkatan yang sangat pesat yang dapat berpengaruh terhadap status gizi balita.

Balita merupakan anak yang rentan terhadap berbagai penyakit. Salah satu upaya untuk meningkatkan kesehatan anak dengan memberikan makanan yang sehat dan imunisasi. Pada usia balita, anak - anak membutuhkan dukungan nutrisi yang lengkap untuk pertumbuhan dan perkembangan tubuh dan otak mereka. Masa balita adalah masa kritis, maka kebutuhan nutrisi bagi balita harus seimbang, baik dalam jumlah maupun kandungan gizi.

Untuk mengetahui sejak dini anak yang kurang gizi (gizi kurang dan gizi buruk) dapat dilakukan dengan pemeriksaan berat badan terhadap umur atau berat badan terhadap tinggi badan untuk memantau status gizi anak. Selain itu pemantauan tumbuh kembang anak dapat juga dengan menggunakan Kartu Menuju Sehat (KMS) (Nurul Azmi dan Fithria : 2015).

Pelayanan gizi di posyandu dilakukan oleh petugas posyandu. Sasarannya adalah bayi, balita, ibu hamil, wanita usia subur. Jenis pelayanan penimbangan berat badan (BB), memeriksa gangguan

pertumbuhan. Dapat dilakukan dengan beberapa cara seperti penyuluhan pada gizi, pemberian makanan tambahan dan vitamin A.

2.2 Aplikasi Berbasis *Hybrid*

Aplikasi berbasis *hybrid* adalah aplikasi yang menggabungkan teknologi pada aplikasi *web* dan aplikasi *native* tetapi tampilan dan cara kerja aplikasi *hybrid* lebih seperti aplikasi *web* daripada aplikasi *mobile*. Melalui pendekatan *hybrid*, pengembang menulis kode dengan bantuan beberapa *framework* dalam pengembangan.

Implementasi aplikasi *hybrid* lebih menjanjikan solusi pengembangan aplikasi *mobile*. Aplikasi *hybrid* menggabungkan kelebihan *web* dan aplikasi *native*. Aplikasi *hybrid* dikembangkan menggunakan HTML5, CSS, PHP, dan *Javascript* tanpa perlu adanya pengetahuan terperinci pada target *platform*. Keuntungan utama dari aplikasi *hybrid* adalah kemampuannya untuk menjalankan *source code* dalam berbagai *platform* dan pengembangannya *source code* yang menggunakan teknologi pengembangan *web* yang banyak digunakan (Indra Waspada dan Dimas Iqbal Pradana : 2019).

Adapun keuntungan membangun aplikasi *hybrid* diantaranya pemeliharaan aplikasi menjadi semakin mudah jika dibandingkan dengan aplikasi *native*, menjadi efisien dalam pengembangan aplikasi terutama untuk keperluan *cross platform*, dapat berjalan dalam kondisi *online* ataupun *offline*, Pengembang dapat melakukan pembaruan konten dan aplikasi itu sendiri tanpa harus meminta pengguna untuk memperbarui aplikasinya, Pengembang dapat melakukan

pembaruan konten maupun aplikasi itu sendiri tanpa harus meminta pengguna untuk memperbarui aplikasinya melalui *native app store*. Ini adalah keuntungan dari aplikasi *mobile* berbasis konten.

Terdapat beberapa kelebihan dari aplikasi *hybrid* dibandingkan aplikasi *native*, yaitu :

1. Waktu yang lebih cepat untuk memasarkan
2. Siklus pengembangan *cross-platform* yang tidak mahal
3. Sumber daya manusia yang melimpah
4. Biaya perawatan
5. Proses persetujuan

Aplikasi *hybrid* juga memiliki beberapa kelemahan dibandingkan dengan aplikasi *native* yang perlu di pertimbangkan dalam mengembangkan sebuah aplikasi nantinya, seperti (Djunaidy Santoso dan Richard Vinc Nicolay Santoso : 2015) :

1. Performa
2. Perbedaan dalam *cross-platform*
3. Ketidatersediaan fitur lanjutan
4. Antarmuka yang tidak konsisten
5. Aplikasi *hybrid* juga memiliki kelemahan dibandingkan dengan aplikasi *web*, misalnya saja tentang aplikasi *hybrid* pada dasarnya tidak dapat diakses via *website*, kecuali jika pengembang menyediakan fitur ini dari awal.

2.3 *Progressive Web Apps (PWA)*

Progressive Web Apps (PWA) adalah konsep pengalaman pengguna yang menggabungkan bagian terbaik *web* dan bagian terbaik *native apps*. *Progressive Web Apps* berguna bagi pengguna sejak pertama membuka halaman sebuah web dengan konsep *Progressive Web Apps*, dan seiring dengan pengguna menggunakan aplikasi web lebih banyak lagi, aplikasi akan menjadi semakin *powerful*. Aplikasi dapat dimuat dengan cepat, bahkan dalam kondisi internet yang kurang baik, bisa mengirim *push notifications*, punya ikon aplikasi di *home screen*, dan bisa berjalan dalam mode layar penuh.

Cache interface menyediakan mekanisme penyimpanan untuk pasangan objek *request* dan *response* mau disimpan ke dalam *cache*, contohnya sebagai dari daur hidup *service worker*. Perlu diketahui bahwa *cache interface* terbuka terhadap halaman *web* dan juga *workers*. *Cache* tidak harus selalu digunakan bersamaan dengan *service worker* walaupun *cache* tercantum di dalam spesifikasinya. *Cache* digambarkan sebagai sebuah *array* berisi objek *request* yang bertindak sebagai pasangan untuk responsnya yang disimpan di dalam *browser*.

Salah satu konsep yang juga diterapkan oleh *Progressive Web Apps* adalah *Service Worker*. *Service worker* adalah kode yang nantinya berjalan di belakang layar *browser* pengguna. *Service worker* pada dasarnya adalah berkas *javascript* yang berjalan pada *thread javascript* yang berjalan pada *thread* berbeda dengan *main thread browser*, menangani *network request*, *caching*, mengembalikan *resource* dari *cache*. *Service worker* tidak membutuhkan sebuah halaman ataupun interaksi dari pengguna untuk menjalankan tugasnya, dengan begitu *service worker* nantinya akan berjalan terus walaupun *web* tidak terbuka. *Service worker* sekarang juga sudah dilengkapi berbagai fitur seperti *push*

notifications. Setelah memasang *service worker* kita dapat mengembalikan salah satu *response* yang telah disimpan dalam *cache*.

Aset web dapat disimpan sebagai *cache* local, sehingga dengan keadaan jaringan internet yang kurang stabil pun, pengguna tetap mendapatkan akses dan pengalaman yang terbaik. Aplikasi nantinya tetap dapat menajlan halaman *web* yang sudah di *cache* atau memberikan status koneksi tanpa *browser* menampilkan tulisan eror karena tidak ada koneksi internet pada pengguna.

Untuk dapat memasang *service worker* ke halaman *web*, kita butuh mendaftarkan dengan menggunakan *javascript* yang ada di halaman *web* kita. Setelah kita daftarkan *browser* akan memulai tahap *install service worker* di latar. Setelah aktif maka *service worker* akan menangani semua halaman *web* di bawah *scope* dimana *service worker* di *install*, selanjutnya akan ada kemungkinan *state : terminated* untuk menghemat memori, atau menangani *fetch* ketika ada *network request* dari halaman *web*.

Service worker bisa menangani berbagai jenis *request*, tetapi yang bisa disimpan kedalam *cache* hanya semua *request* jenis *GET*. Pada *service worker* ini bisa di aplikasikan strategi *caching* sesuai keinginan kita, namun tidak ada satu strategi terbaik untuk *caching* konten dinamis, dan ada banyak situasi yang bisa mempengaruhi strategi *caching* (Wijayanti Nurul Khotimah, Laurensius Adi dan Rizky Januar Akbar : 2017).

2.4 HTML5 (*Hypertext Markup Language 5*)

HTML5 ini merupakan versi terbaru dari HTML. HTML5 menjadi Bahasa markup yang unggul saat ini dan telah memiliki peningkatan banyak fitur dari HTML biasa. HTML ini juga merupakan sebuah bahasa markah yang bahasa

markah yang digunakan untuk membuat sebuah halaman *web*. Perbedaan utama antara HTML dan HTML5 di audio ataupun video, namun di keseluruhan HTML dimana keduanya merupakan bagian dari HTML5. Pada HTML5 mendukung *canvas*, *scalable vector graphics* (SVG), dan grafis *vector* lainnya, sedangkan pada HTML hanya penggunaan grafis *vector* mungkin dilakukan dengan bantuan teknologi seperti *Flash*, *Vector Markup Language* (VML), dan sebagainya.

HTML5 mendukung dalam penggunaan *form control* yang baru seperti tanggal dan waktu, email, nomor, jumlah, pencarian, url, dan sebagainya. HTML5 juga dilengkapi dengan banyak elemen baru dan beberapa yang paling penting seperti *summary*, *aside*, *time*, *audio*, *data*, *command*, *footer*, *header*, *canvas*, *keygen*, *mark*, *nav*, *output*, *progress*, *section*, *track*, *audio*, *source*, dan *video*.

HTML5 telah dirancang sebagai bahasa yang *cross-platform* penggunaannya tidak didasarkan pada *platform* dimana HTML5 itu sendiri dapat berjalan. Namun secara umum HTML5 mulai banyak diaplikasikan pada banyak *website* di internet. Hal itu diikuti oleh penelusuran *web* ternama seperti Chrome, Opera, Mozilla Firefox, Safari, dan Internet Explorer yang memberikan dukungan pada HTML5. Hal ini menunjukkan bahwa HTML5 akan segera menjadi standar sebagai bahasa *markup* untuk *web* (Oesman Hendra Kelana, dan Prima Astiadi : 2014).

2.5 CSS (*Cascading Style Sheet*)

CSS (*Cascading Style Sheets*) adalah bahasa yang biasanya digunakan untuk mengatur tampilan pada elemen yang tertulis dalam bahasa markup seperti HTML.

Cascading style sheets berfungsi untuk memisahkan konten dari tampilan visualnya pada situs web. HTML dan CSS memiliki hubungan yang erat. Karena HTML adalah bahasa *markup* nya sebagai fondasi *web* dan CSS sebagai kode-kode yang akan dipakai untuk mendesain sebuah halaman HTML nantinya. Jika diibaratkan HTML sebagai kerangkanya maka CSS adalah sebagai membuat penampilan kerangkanya menjadi semakin menarik.

Dengan adanya CSS seorang *web designer* lebih mudah dalam mengendalikan *website* yang akan dibangun. Dalam memprogram ulang tampilan laman *website* tidak harus memperbaiki HTML nya. Pada CSS tingkat lanjut seorang *web designer* dapat mengatur *website* melalui media lain seperti suara, video, dan animasi. CSS juga akan melatih seorang *web designer* untuk membuat *website* yang beragam variasi tampilannya. Pembuatan CSS biasanya terpisah dengan halaman HTML. Meskipun CSS dapat disisipkan di dalam halaman HTML. Hal ini ditujukan untuk memudahkan pengaturan halaman HTML yang memiliki rancangan yang sama (Miftahul Khoir, Omar Pahlevi dan Astriana Mulyani : 2018).

2.6 JavaScript

JavaScript adalah sebuah bahasa komputer atau bahasa pemrograman yang akan digunakan pada *website* agar *website* akan menjadi lebih manis dan interaktif. *Javascript* adalah jenis bahasa pemrograman *client side* dan bersifat opsional, artinya tidak harus selalu ada. *Client Side Programming Language* adalah tipe bahasa pemrograman yang pemrosesannya dilakukan oleh *client*. Namun

website-website maupun blog modern hampir semuanya menggunakan kode *javascript* walaupun sedikit.

Dengan adanya *javascript* pada *website* maka *website* nantinya akan lebih hidup dan menarik. *Javascript* berbeda dengan bahasa pemrograman lainnya seperti pada HTML yang mengatur konten pada *website*, CSS yang digunakan untuk mendesain tampilan *website*, dan juga bahasa pemrograman PHP, bahasa pemrograman ini dijalankan di perangkat pengunjung situs *website* dan bukannya di *server*.

Javascript sebagai salah satu bahasa pemrograman juga memiliki banyak kelebihan antara lain, lebih mudah dipelajari, eror atau kesalahan lebih mudah dicari dan diatasi, dapat digunakan di berbagai *browser* maupun *platform*, lebih cepat dan ringan jika dibandingkan dengan bahasa pemrograman lainnya.

Javascript dapat merespon perintah *user* dengan cepat dan menjadikan halaman *web* menjadi responsif. *Javascript* memiliki struktur sederhana, kodenya dapat disisipkan pada dokumen HTML atau berdiri sebagai satu kesatuan aplikasi (Indra Yatini B : 2015).

2.7 Bootstrap

Bootstrap merupakan salah satu dari jenis *framework* pada CSS (*Cascading Style Sheet*) yang digunakan untuk merancang situs *web* dan aplikasi *web*. *Framework* ini berisi *template* yang sudah di desain berbasis HTML dan CSS untuk tipografi, formulir, tombol, navigasi dan komponen antarmuka lainnya. *Framework* ini bersifat *open source* dan hanya fokus pada pengembangan tampilan desain *website* saja.

Bootstrap juga telah menyediakan rangkaian aturan dan komponen antar muka dasar sebagai asset dalam pembuatan web yang telah dirancang sangat baik untuk memberikan tampilan yang sangat menarik. Ringan, dan memudahkan bagi penggunanya. *Bootstrap* juga telah dilengkapi dengan berbagai komponen *javascript* dan menyediakan tambahan elemen antarmuka untuk pengguna seperti kotak dialog, *tooltips*, dan *carousels*.

2.8 Hypertext Processor (PHP)

Hypertext Processor (PHP) merupakan bahasa pemrograman *server-side* scripting yang dirancang untuk pengembangan *web* yang juga menyatu dengan HTML. PHP disebut juga sebagai bahasa pemrograman *server side* karena diproses pada *server* kemudian hasilnya akan dikirim ke *browser* dengan format HTML. *Server-side* adalah jenis bahasa pemrograman yang nantinya *script* atau program tersebut akan dijalankan/diproses oleh server. PHP banyak di pakai pada situs *web* yang bersifat dinamis. Kode yang ditulis dalam PHP nantinya tidak akan terlihat oleh *user* sehingga keamanan *web* akan terjamin. PHP telah dirancang untuk membentuk suatu tampilan berdasarkan permintaan terkini, seperti menampilkan suatu data ke halaman *web*.

Adapun beberapa kelebihan PHP dari bahasa pemrograman web lainnya, sebagai berikut (Wiyli Yustanti, dan Dio Lavarino : 2016) :

1. Bahasa pemrograman pada PHP ialah sebuah bahasa *script* yang tidak melakukan sebuah kompilasi dalam penggunaannya.
2. PHP memiliki tingkat akses yang lebih cepat

3. PHP memiliki tingkat *lifecycle* yang cepat sehingga selalu mengikuti perkembangan teknologi.
4. PHP juga dapat mendukung akses ke beberapa database yang sudah ada baik bersifat *open source* ataupun komersial. *Database* yang mendukung PHP seperti *MySQL*, *PostgreSQL*, *infomix*, dan *MicrosoftSQL Server*.

Dalam sisi pengembangan lebih mudah, karena itu banyak yang memilih bahasa pemrograman PHP dalam pembuatan maupun pengembangan aplikasi.

2.9 MySQL

MySQL bersifat *open source* sehingga kita bisa menggunakannya secara gratis. MySQL merupakan basis data yang paling diminati dikalangan programmer *web*, dengan alasan bahwa program ini merupakan basis data yang sangat kuat dan cukup stabil untuk digunakan sebagai media penyimpanan data. Sebagai sebuah basis data *server* yang mampu untuk manajemen basis data dengan baik, mysql terhitung merupakan basis data yang paling digemari dan paling banyak digunakan dibanding basis data lainnya. Selain mysql masih terdapat beberapa jenis basis data server yang juga memiliki kemampuan yang juga tidak bisa dianggap mudah, basis data itu adalah Oracle dan PostgreSQL.

Sebagaimana *database* sistem yang lain, dalam SQL juga dikenal hirarki *server* dengan *database - database*. Tiap - tiap *database* memiliki tabel - tabel, tiap - tiap tabel memiliki *field - field*. Umumnya informasi tersimpan dalam tabel - tabel yang secara logic merupakan struktur dua dimensi terdiri atas baris dan kolom. *Field - field* tersebut dapat berupa data seperti *int* , *realm char*, *date*, *time* dan lainnya. SQL tidak memiliki fasilitas

pemrograman yang lengkap, tidak ada *looping* ataupun percabangan, misalnya. Sehingga untuk menutupi kelemahan ini perlu digabung dengan bahasa pemrograman semisal C.

MySQL adalah salah satu jenis database yang banyak digunakan untuk membuat aplikasi berbasis *web* yang dinamis. *MySQL* termasuk jenis RDBMS (*Relational Database Management Sistem*). *MySQL* ini mendukung Bahasa pemrograman PHP. *MySQL* juga mempunyai *query* atau bahasa SQL (*Structured Query Language*) yang sederhana dan menggunakan *escape character* yang sama dengan PHP (Miftahul Khoir, Omar Pahlevi, dan Astriana Mulyani : 2018).

2.10 Xampp

Xampp merupakan aplikasi yang membantu menyediakan paket perangkat lunak ke dalam sebuah paket. Dengan menginstal *Xampp* maka tidak perlu lagi melakukan instalasi dan konfigurasi *Web Server Apache*, PHP, dan *MySQL* secara manual. Setelah *Xampp* diinstall maka akan melakukan konfigurasi secara otomatis. *Xampp* ini juga merupakan satu paket dengan PHP dan berbasis *open source* yang telah dikembangkan oleh sebuah komunitas *open source*.

Xampp juga mendukung banyak sistem operasi dan juga merupakan kompilasi dari beberapa program yang berfungsi sebagai *server* yang berdiri sendiri atau disebut dengan *localhost* yang terdiri dari program *Apache*, *HTTP Server*, *MySQL Database*, dan penerjemah bahasa yang telah ditulis dengan bahasa pemrograman PHP dan *Perl*. Dengan melakukan instalasi pada *Xampp* semua kebutuhan sudah tersedia jadi tidak perlu menginstal program lain karena sudah

tersedia seperti *Apache*, *MySQL*, *PHP*, *FilZilla FTP Server*, dan *PHPmyAdmin* (Radna Nurmalina dan Santoso : 2017).

2.11 Visual Studio Code

Visual Studio Code merupakan aplikasi *code editor* yang sangat ringan yang berjalan pada *desktop*. *Visual studio code* ini juga mendukung untuk *Javascript*, *Node.js*, dan memiliki beragam ekstensi yang tersedia untuk bahasa lain, termasuk *C++*, *C#*, *Python*, dan *PHP*. *Visual Studio Code* ini memiliki fitur lengkap yang pengembangan terpadu pada *Integrated Development Environment (IDE)* yang telah dirancang untuk pengembang yang bekerja dengan teknologi *cloud* yang terbuka *Microsoft*. *Visual Studio Code* juga merupakan *open source* dan memiliki antarmuka yang mudah untuk digunakan. Dalam hal ini *Visual Studio Code* telah dikembangkan dengan baik, dan menyenangkan mata dan juga memiliki fungsi yang baik.

Visual Studio Code juga dapat melakukan pencarian apapun yang akan diinginkan untuk cepat dan mudah didapat dengan perintah *Git*, mengubah tema tampilan *code editor*, menginstal *extension* yang akan diinginkan via *Command Palette*. *Visual Studio Code* ini juga sudah terintegrasi dengan *Git*.

2.12 Adobe XD CC

Adobe XD CC merupakan sebuah perangkat lunak desain yang hanya berfokus untuk desain *user interface* dan *user experience*. Pengguna dapat membuat desain berupa *interface* aplikasi yang sederhana kemudian nantinya dapat dibuat menjadi *prototype*-nya dan dapat dilihat oleh anggota tim lain, seperti *project manager* dan *developer*. *Adobe XD*

CC didesain dan dibuat oleh *Adobe System*, sehingga mendukung format file yang dibuat oleh aplikasi *Adobe* lainnya, seperti *Adobe Photoshop* dan *Adobe Illustrator*.

Adobe XD CC merupakan perangkat lunak yang baru rilis. Sebelum aplikasi ini dikenal dengan nama *Project Comet* pada April 2016. Kemudian, *Project Comet* ini berubah menjadi *Adobe Experience Design CC*. Perangkat lunak ini akhirnya resmi meluncur dengan nama *Adobe XD CC* pada akhir Oktober 2017. Perangkat lunak ini juga menjadi alternatif lainnya seperti *Sketch* dan *Adobe Photoshop* (Ta'zirah Marwan : 2018).

2.13 Unified Modeling Language (UML)

Unified Modeling Language (UML) adalah sebuah bahasa yang telah menjadi bahasa standar yang digunakan untuk mendokumentasikan, menspesifikasikan dan merancang bangun perangkat lunak. Dengan adanya UML kita dapat membuat model untuk semua jenis aplikasi dalam mendukung pengembangan sistem.

Adapun alat bantu yang akan digunakan dalam perancangan berorientasi objek berbasis UML adalah sebagai berikut :

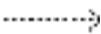
1. Use Case Diagram

Use Case Diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use Case Diagram* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut.

Use Case Diagram sangat dapat membantu kita dalam menyusun rancangan sebuah sistem, dan merancang test case pada semua fitur yang ada

pada sistem. *Use case* ini dapat digunakan atau diaplikasikan ke *use case* lainnya, sehingga duplikasi fungsionalitas dapat dihindari, caranya adalah dengan menarik keluar fungsional yang *common*. Adapun simbol-simbol yang digunakan pada *Use Case Diagram* yaitu sebagai berikut :

Tabel 2.1 Simbol *Use Case Diagram*

No.	Simbol	Nama	Keterangan
1.		<i>Actor</i>	Aktor adalah orang atau sistem yang lain yang melakukan untuk mengaktifkan fungsi dari target sistem. Orang atau sistem nantinya akan muncul dalam beberapa peran saja.
2.		<i>Dependency</i>	Interaksi antara actor dan <i>use case</i> yang menunjukkan bila aktor sedang berinteraksi dengan dengan sitem secara pasif
3.		<i>Include</i>	Merupakan pemanggilan antara <i>use case</i> dan <i>use case</i> lainnya.
4.		<i>Extend</i>	Merupakan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
5.		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.

6.		<i>Use Case</i>	Merupakan urutan dari aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
----	---	-----------------	---

Sumber : Ade Hendini : 2016

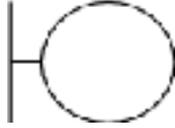
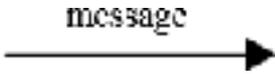
2. *Sequence Diagram*

Sequence diagram merupakan gambaran interaksi antar objek di dalam dan di setikar sistem dan juga dapat menggambarkan kelakuan *user case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek.

Sequence diagram biasanya digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sehingga respons dari sebuah *event* untuk menghasilkan *event* tertentu. Adapun simbol-simbol yang digunakan pada *Sequence Diagram* yaitu sebagai berikut :

Tabel 2.2 Simbol *Sequence Diagram*

No.	Simbol	Nama	Keterangan
1.		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .

2.		<i>Lifeline</i>	Merupakan suatu keberadaan sebuah object ke dalam basis waktu.
3.		<i>Activation</i>	Merupakan sebuah objek yang akan melakukan sebuah aksi.
4.		<i>Boundary</i>	Merupakan kumpulan kelas yang akan menjadi tampilan atau hubungan antara satu atau lebih aktor dengan suatu sistem, seperti <i>form input</i> dan <i>form cetak</i> .
5.		<i>Control</i>	Merupakan suatu objek yang tidak memiliki logika tanggung jawab pada entitas
6.		<i>Entity</i>	Merupakan bagian dari suatu sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.
7.		<i>Message</i>	Merupakan suatu gambaran komunikasi antara object atau komunikasi antar <i>class</i> .
8.		<i>Self-Message/ Recursive</i>	Merupakan suatu komunikasi kembali kedalam sebuah objek itu sendiri.

3. *Activity Diagram*

Activity Diagram menggambarkan berbagai aliran aktivitas dalam sebuah sistem yang akan dirancang, *Activity Diagram* nantinya menampilkan proses-proses apa saja yang akan terjadi pada sistem yang terjadi pada sistem. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. Adapun simbol-simbol yang digunakan pada *Activity Diagram* yaitu sebagai berikut :

Tabel 2.3 Simbol *Activity Diagram*

No.	Simbol	Nama	Keterangan
1.		<i>Initial/Start Point</i>	Merupakan sebuah titik awal untuk memulai suatu aktivitas.
2.		<i>Dependency /End Point</i>	Merupakan suatu titik akhir untuk mengakhiri suatu aktivitas.
3.		<i>Activity</i>	Menandakan adanya sebuah proses aktivitas.
4.		<i>Decision</i>	Merupakan suatu proses pilihan dalam mengambil keputusan, seperti <i>true</i> dan <i>false</i>
5.		<i>Fork / Join</i>	Merupakan suatu kegiatan yang ditujukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.

6.		<i>Rake</i>	Menunjukkan adanya dekomposisi.
----	---	-------------	---------------------------------

Sumber : Sumber : Ade Hendini : 2016

4. *Class Diagram*

Class Diagram merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga menampilkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem.

Class Diagram secara khas meliputi Kelas (*Class*), Relasi *Assosiations*, *Generalitation* dan *Aggregation*, Atribut (*Attributes*), operasi (*operation/method*) dan *visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan *Multiplicity* atau *Cardinality* (Ade Hendini : 2016).

Tabel 2.4 Simbol *Class Diagram*

No.	Multiplicity	Penjelasan
1.	1	Satu dan hanya satu
2.	0..*	Boleh tidak ada atau 1 atau lebih

3.	1..*	1 atau lebih
4.	0..1	Boleh tidak ada, maksimal 1
5.	n..n	Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimal 4

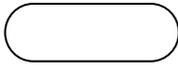
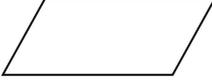
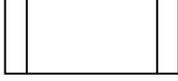
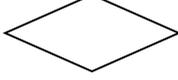
Sumber : Ade Hendini : 2016

2.14 Flowchart

Flowchart merupakan gambaran secara grafik dari suatu langkah-langkah atau prosedur menyelesaikan suatu masalah maupun program, dengan adanya *flowchart* nantinya dapat memudahkan seseorang dalam melakukan pengecekan bagian-bagian yang terlupakan dalam analisis masalah ataupun program. Selain itu *flowchart* dapat juga berguna untuk berkomunikasi antara *programmer* yang berkerja dalam suatu tim pada saat membangun proyek, program, dan masalah.

Flowchart dapat juga membantu memahami dalam urutan-urutan logika yang rumit dan panjang. *Flowchart* dapat menolong seorang analisis maupun *programmer* dalam memecahkan masalah kedalam bagian-bagian yang lebih kecil dan memabntu menganalisis hal-hal yang nantinya akan menjadikan alternatif kedalam pengoperasian. Adapun Simbol-simbol *flowchart* sebagai berikut (Radna Nurmalina dan Santoso : 2017) :

Tabel 2.5 Simbol *Flowchart*

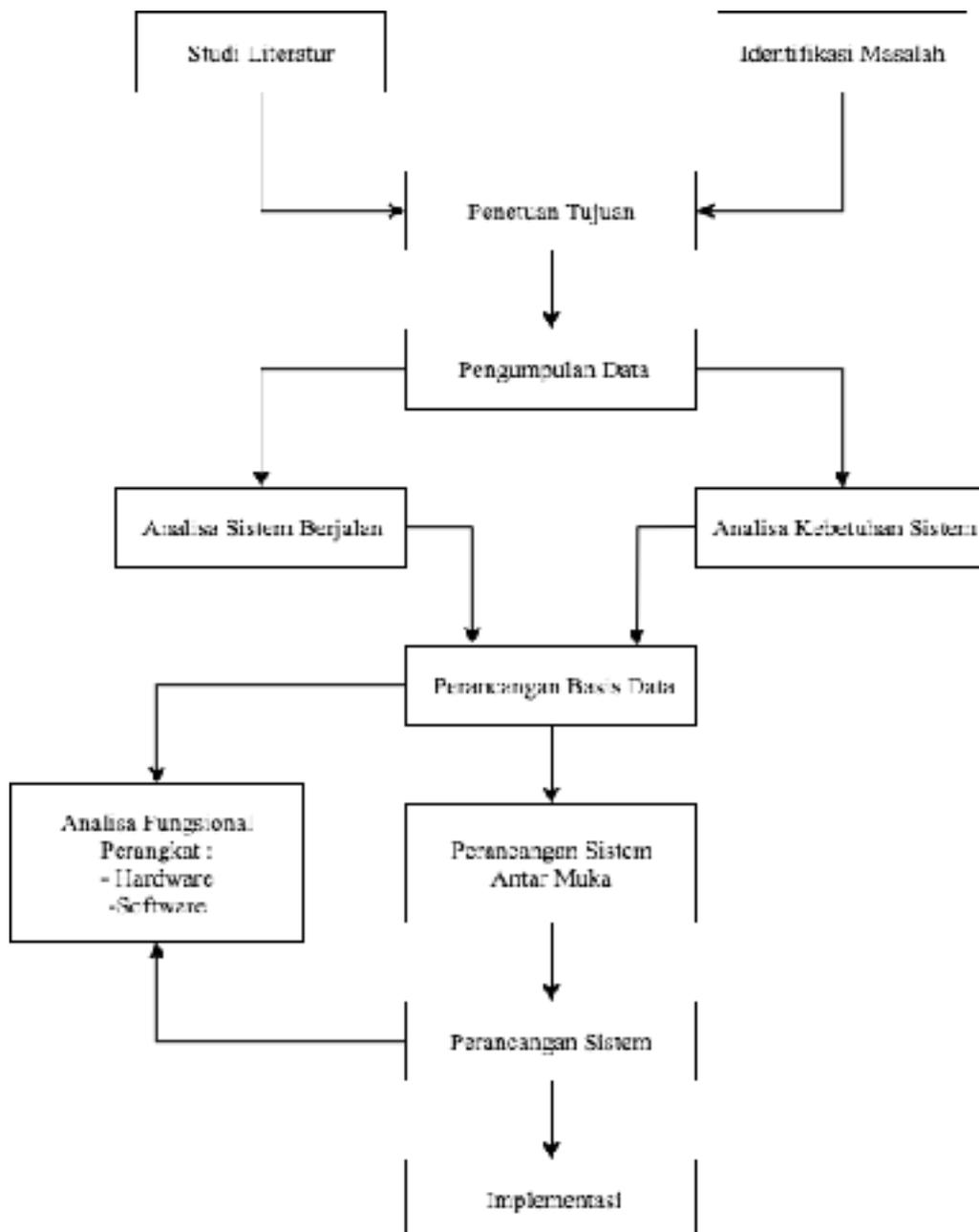
No.	Simbol	Nama	Keterangan
1.		<i>Terminal</i>	Awal atau akhir pada program
2.		<i>Arus/Flow</i>	Arah aliran pada program
3.		<i>Preparation</i>	Inisialisasi atau pemberian nilai awal pada program
4.		<i>Proses</i>	Proses/pengolahan data
5.		<i>Input / Output</i>	Proses <i>input/output</i> pada data
6.		<i>Predefined Process</i>	Permulaan pada sub program
7.		<i>Keputusan</i>	Perbandingan atau kondisi data yang menentukan langkah selanjutnya
8.		<i>Penghubung</i>	Penghubung bagian-bagian <i>flowchart</i> pada halaman yang sama
9.		<i>Off Page Connector</i>	Penghubung bagian-bagian <i>flowchart</i> pada halaman yang berbeda
10.		<i>Penjelasan</i>	Tempat komentar tentang suatu proses untuk memperjelas

11.		Dokumen	Simbol yang menyatakan <i>input</i> berasal dari dokumen dalam bentuk kertas atau <i>output</i> di cetak dikertas
-----	---	---------	---

Sumber : Radna Nuralina dan Santoso : 2017

BAB III
METODE PENELITIAN

3.1 Tahapan Penelitian



Gambar 3.1 Tahapan Penelitian

Berikut ini sedikit penjelasan tentang alur pada tahapan penelitian *workflow* diatas merupakan proses pembuatan program:

a. Studi Literatur

Studi literatur merupakan suatu kegiatan mencari beberapa teori digunakan sebagai refrensi yang dapat dipahami dan diterapkan dalam penyelesaian masalah. Penulis juga mengumpulkan refrensi buku-buku mengenai perancangan *website* dan *Progressive Web Apps* maupun jurnal-jurnal yang memiliki kemiripan dalam pembuatan sistem ini.

b. Identifikasi Masalah

Identifikasi Masalah merupakan penegasan pada batasan permasalahan, sehingga jangkauan peneltian yang dilakukan tidak keluar dari tujuan dan mencari solusi dalam masalah tersebut, sehingga masalah dapat terselesaikan dengan baik.

c. Penentuan Tujuan

Setelah menemukan masalah yang sedang terjadi pada penelitian maka selanjutnya menentukan tujuan untuk menemukan solusi terutama dalam bidang teknologi informasi dalam penyelesaian masalah dengan cara menentukan sistem apa yang cocok untuk permasalahan yang ada, sehingga masalah dapat terselesaikan dengan baik.

d. Pengumpulan Data

Pada saat pengumpulan data dapat dilakukan dengan cara paling mendasar seperti melakukan observasi, wawancara, kuesioner dan kilas balik dokumen untuk mencapai proses suatu proses yang diinginkan.

e. Analisa Sistem Berjalan

Analisa sistem berjalan merupakan proses merupakan gambaran tentang sistem yang saat ini sedang berjalan di posyandu pada bagian memantau hasil kesehatan anak, sistem yang digunakan masih sederhana dan manual yaitu dengan menggunakan media kertas sebagai tempat menyimpan hasil kesehatan anak di posyandu. Analisa sistem ini bertujuan untuk membuat sistem yang baru sehingga dapat lebih efektif dan efisien.

f. Analisa Fungsional Perangkat

Bedasarkan analisis permasalahan yang telah disampaikan sebelumnya, maka diperlukan suatu aplikasi *monitoring* kesehatan balita melalui e-posyandu berbasis *hybrid programming* yang memudahkan posyandu dalam melakukan *monitoring* hasil pemeriksaan kesehatan balita sehingga dapat diakses dimana saja hal ini sebagai alternatif dari pada menggunakan media kertas sebagai rekam medis yang dapat hilang sewaktu-waktu. Dengan adanya analisa ini kita dapat mengetahui apa saja yang dibutuhkan untuk menjalankan sistem yang akan digunakan, seperti perangkat lunak, *database*, kebutuhan *design* pada tampilan yang akan digunakan nantinya dan juga pengalaman *user experience* dalam penggunaan sistem.

g. Perancangan Basis Data

Perancangan ini merupakan untuk menentukan penyimpanan data, gambar, *username*, *password*, maka diperlukan tabel-tabel data pengguna, kapasitas penyimpanan yang dibutuhkan oleh sistem.

h. Analisa Kebutuhan Perangkat

Analisa kebutuhan perangkat ini merupakan suatu proses dari pengembangan sistem, maka kebutuhan perangkat dalam pengembangan sistem harus diketahui agar dapat digunakan, seperti perangkat keras dan perangkat lunaknya.

i. Perancangan Sistem Antarmuka

Dalam perancangan ini sistem akan di desain semudah dan seinteraktif mungkin dalam penggunaannya, dengan menggunakan pengalaman terbaik dalam mendesain maka harus mengutamakan pengalaman pengguna yang baik dan tata letak seperti tombol, tabel, data, dan gambar disajikan dengan baik sehingga membuat pengguna dapat mudah menjalankan aplikasinya.

j. Perancangan sistem

Setelah sistem didesain dan memiliki konsep alur yang akan diterapkan dari beberapa elemen yang terpisah selanjutnya semua akan disatukan untuk mencapai hasil sistem yang diinginkan.

k. Implementasi

Proses implementasi ini merupakan proses menerapkan dari hasil program sistem yang telah kita buat ke sistem yang sudah ada maupun sistem baru.

3. 2 Metode Pengumpulan Data

Adapun proses yang akan dilalui dalam proses pengumpulan data dalam membantu penulisan ini, yaitu sebagai berikut:

a. Penelitian Kepustakaan

Penulis melakukan penelitian kepustakaan dalam mencari jurnal-jurnal resmi dari internet untuk mencari informasi yang berkaitan dengan teori judul skripsi. Dengan mengangkat suatu teori tidak mengalami kendala pada penelitian maka diperlukan observasi terhadap teori-teori yang digunakan oleh penulis dan penulis juga mengambil teori dari berbagai sumber jurnal dalam membantu perancangan sistem yang digunakan.

b. Wawancara

Setelah melakukan observasi selanjutnya penulis akan mewawancarai kepala badan kesehatan ibu dan anak dan masyarakat sekitar maupun ibu pasien apa saja yang seharusnya akan dimuat nantinya di aplikasi yang akan dirancang, termasuk tampilannya, apa saja yang akan disimpan ke *database*, menunya sebaiknya bagaimana, dengan melibatkan masyarakat, petugas posyandu, mahasiswa hal ini dapat membantu dalam meningkatkan pengalaman dalam penggunaan aplikasi yang akan dirancang.

3. 3 Analisis Sistem Berjalan

Proses sistem saat ini akan diuraikan sebagai prosedur analisa sistem yang berjalan. Adapun uraian yang merupakan dari prosedur-prosedur saat ini sebagai berikut:

1. Pendaftaran Pelayanan

Pada saat pasien datang ke posyandu pasien harus mendaftar terlebih dahulu agar mendapatkan pelayanan dari posyandu. Selanjutnya pasien memberikan kartu menuju sehat (KMS) kepada petugas posyandu. Kemudian pasien balita melakukan penimbangan berat badan, tinggi badan, vaksin, dan imunisasi hasilnya akan dicatat pada kartu menuju sehat (KMS) dan menghasilkan data pasien (DP) yang nantinya akan diserahkan kepada dokter yang sedang bertugas dan akan disalin ke buku data pasien untuk di arsipkan.

2. Praktek Medis

Pada proses ini dokter akan melakukan tugasnya yaitu dalam praktek medis. Lalu hasil dari praktek medis ini akan masuk dalam data praktek dan diserahkan kepada petugas posyandu. Selanjutnya petugas akan menyalin hasil data praktek nya kedalam arsip lalu menyerahkan kembali hasil data praktek dan kartu menuju sehat (KMS) kepada pasien.

3. Laporan

Hasil dari arsip data praktek dan arsip data pasien, selanjutnya petugas posyandu membuat laporan kegiatan yang telah dilaksanakan pada posyandu pada setiap bulannya untuk diserahkan kepada kepala posyandu dan arsipkan.

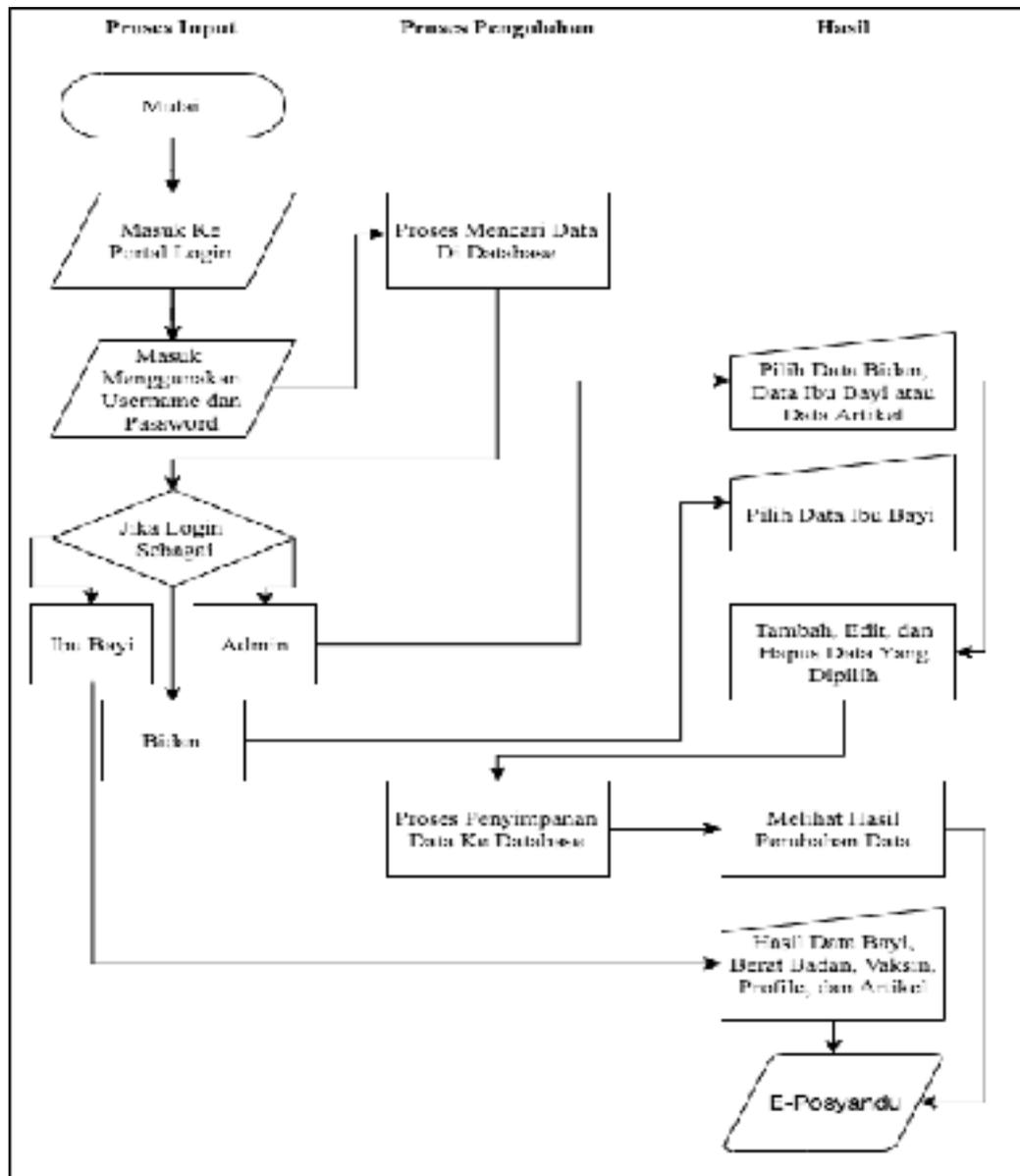
3. 4 Rancangan Penelitian

Pada rancangan penelitian ini diperlukan agar terciptanya sistem yang di perlukan dan dapat berjalan dengan baik dan juga menjadi lebih efisien serta dapat menghasilkan pengalaman terbaik terhadap pengguna,

3.4.1 Analisis Sistem Usulan

Sistem yang akan dirancang saat ini adalah aplikasi *monitoring* kesehatan balita melalui e-posyandu berbasis hybrid programming, dimana semua informasi yang berkaitan dari hasil rekap medis ataupun laporan kegiatan posyandu dapat diakses di aplikasi ini dengan cepat dan mudah tanpa mencari dokumen atau buku hasil rekap medis di posyandu.

Adapun *Flow of Document* Sistem yang akan berjalan pada aplikasi ini sebagai berikut:



Gambar 3.2 *Flow of Document* Sistem Usulan

Berikut ini penjelasan analisis *Flow of Document* sistem berjalan dari gambar 3.2 sebagai berikut:

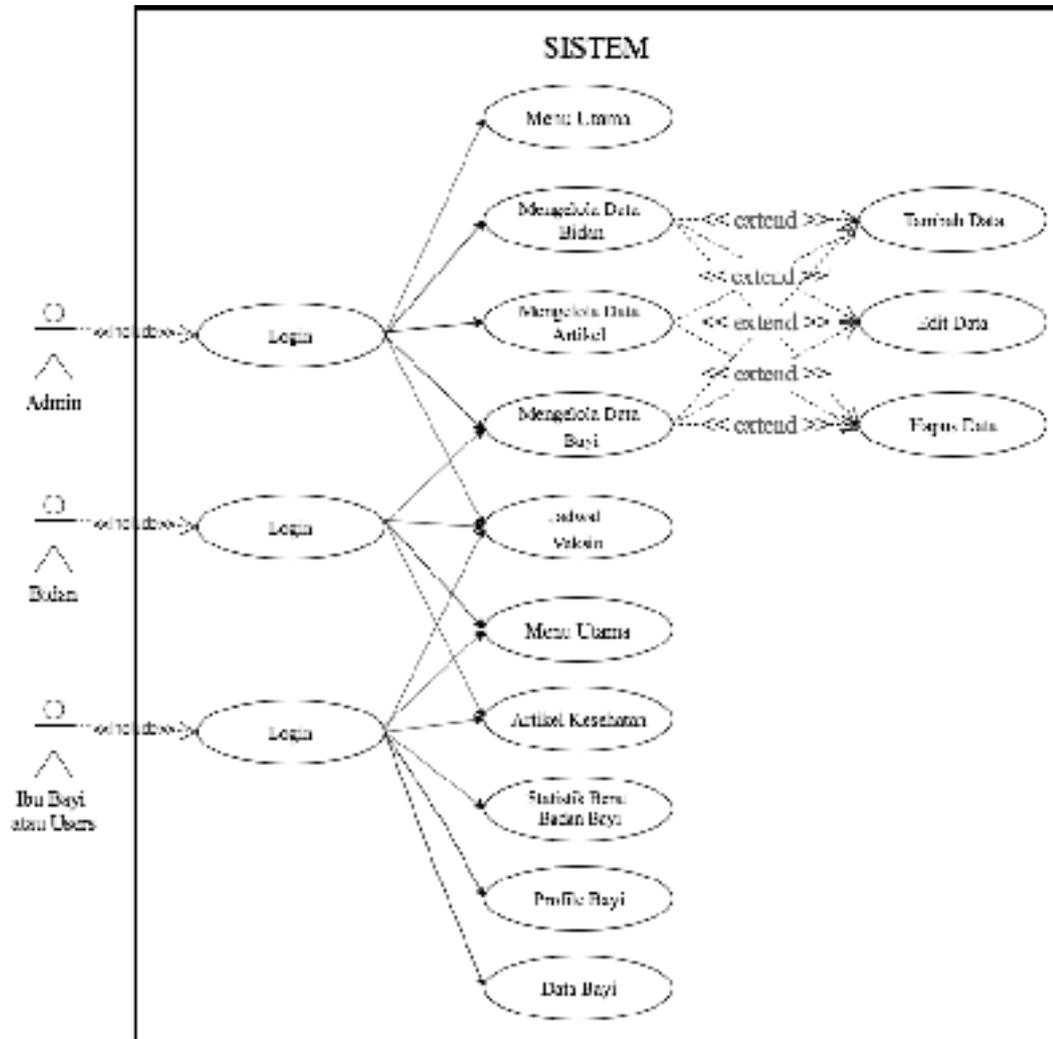
1. Membuka Aplikasi.
2. Masuk ke portal *login*.
3. Masuk menggunakan *username* dan *password*.

4. Tunggu proses mencari data di *database*.
5. Jika *login* sebagai *admin* maka akan masuk dihalaman *dashboard*, yang berisi data bidan, data ibu bayi, data artikel.
6. Jika *login* sebagai bidan maka akan masuk dihalaman *dashboard*, yang berisi data ibu bayi.
7. Pilih data yang akan ditambah, diedit, dan dihapus.
8. Lalu data yang tadi akan disimpan ke *database*.
9. Lihat hasil perubahan data.
10. Jika *login* sebagai ibu bayi maka akan masuk dihalaman *dashboard*, yang berisi data bayi, statistik berat badan bayi, jadwal vaksin bayi, dan artikel tentang kesehatan bayi.
11. E-posyandu berhasil digunakan.

3. 5 Rancangan Secara Global

a. Use Case Diagram

Use Case Diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat dan mendeskripsikan sebuah interaksi antara satu dan lebih actor dengan sistem yang akan dibuat. *Use Case Diagram* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Pada aplikasi *monitoring* kesehatan balita melalui e-posyandu berbasis *hybrid programming* terdapat *use case diagram* yaitu sebagai berikut :



Gambar 3.3 Use Case Diagram Pada Aplikasi E-Posyandu

Pada gambar 3.3 menggambarkan aktor-aktor yang terlibat di dalam sistem. Aktor pertama adalah *admin*, didalam sistem admin memiliki hak akses untuk masuk ke menu utama, mengelola data bidan, mengelola data artikel, mengelola data bayi, dan melihat jadwal vaksin.

Aktor kedua adalah bidan, didalam sistem bidan memiliki hak akses untuk masuk ke menu utama, mengelola data bayi, melihat jadwal vaksin dan melihat artikel kesehatan. Sifat dari pengguna ini adalah *multiuser*

karena terdapat lebih dari satu bidan dalam satu posyandu yang dapat mengelola data posyandu.

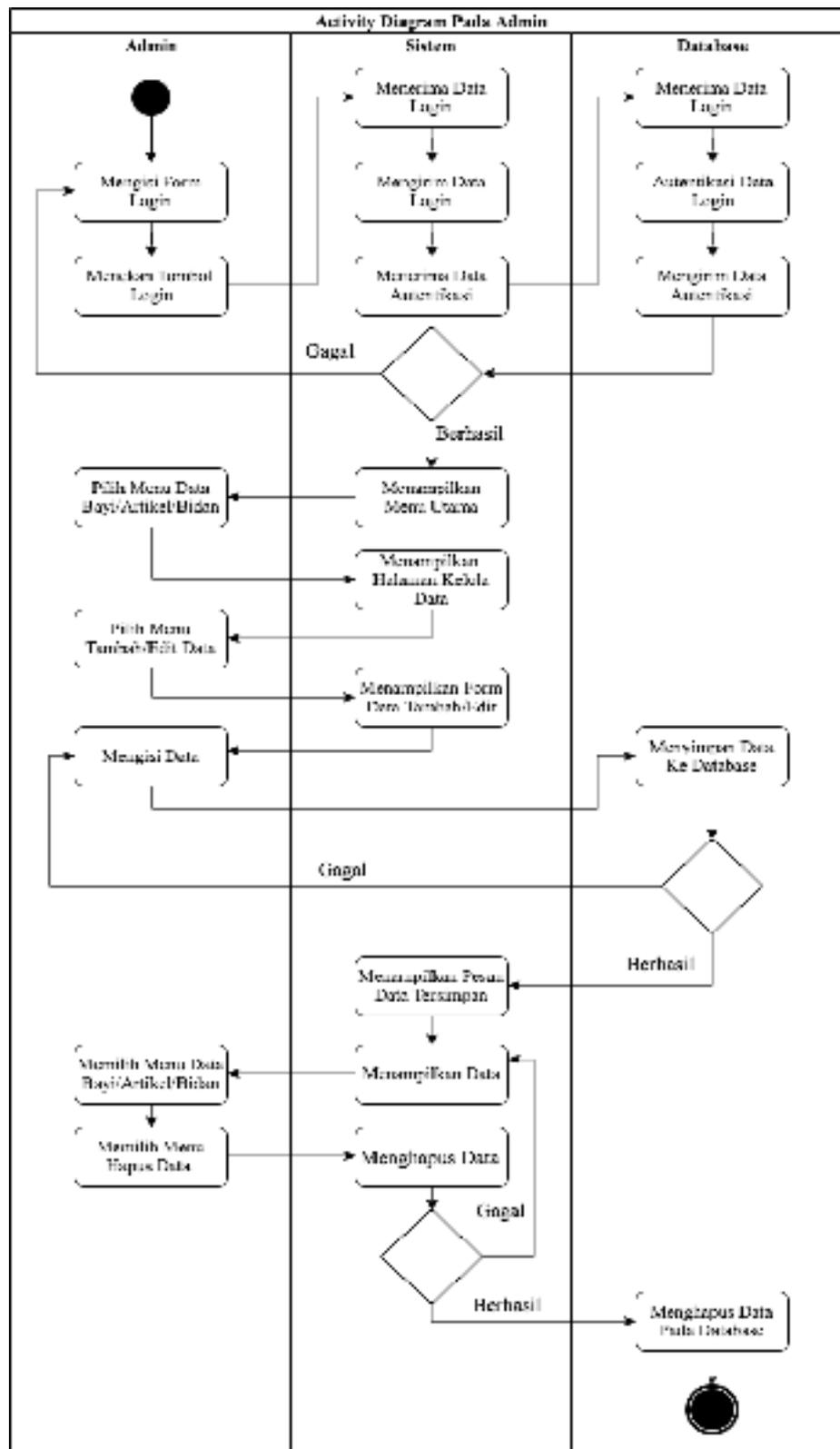
Aktor kedua adalah ibu bayi, didalam sistem bidan memiliki hak akses untuk masuk ke menu utama, melihat data bayi, melihat statistik berat badan bayi, melihat *profile* bayi, melihat jadwal vaksin dan melihat artikel kesehatan. Sifat dari pengguna ini adalah *multiuser* karena terdapat lebih dari satu pengguna atau ibu bayi dalam satu Posyandu yang dapat mengakses sistem.

b. Activity Diagram

Activity diagram merupakan bentuk dari visual alur kerja berisi aktivitas dan tindakan, yang juga dapat berisi pilihan atau pengulangan. *Activity diagram* nantinya menampilkan proses-proses apa saja yang akan terjadi pada sistem yang terjadi pada sistem. Pada aplikasi *monitoring* kesehatan balita melalui e-posyandu berbasis *hybrid programming* terdapat *activity diagram* yaitu sebagai berikut :

1. *Activity Diagram* pada *Admin*

Activity diagram dibawah ini menggambarkan sistem dari sudut pandang sebagai *admin*. Pada sistem ini *admin* sebagai pengelola data bayi, data bidan, dan data artikel. Berikut merupakan *activity diagram* pada *admin* terlihat pada gambar dibawah ini :



Gambar 3.4 Activity Diagram Pada Admin Aplikasi E-Posyandu

Pada gambar 3.4 diatas menjelaskan bahwa aktifitas diagram pada proses masuk ke dalam aplikasi, dimana ketika aplikasi dimulai maka secara otomatis akan menuju ke halaman *login*, lalu ke halaman *form login* ketika mengisi *form login* menekan tombol login lalu sistem menerima data *login* dan mengirim data *login* ke *database* lalu *database* nantinya akan mengirim autentikasi jika berhasil *login* menggunakan data pengguna selanjutnya akan menuju ke halaman utama aplikasi jika gagal maka akan kembali ke *form login* dan pada halaman menu utama terdapat data bayi, data bidan, dan data artikel yang dapat di pilih data nya yang akan ditambah, diedit atau dihapus. Jika data berhasil ditambah, edit dan hapus maka akan muncul pesan data yang tersimpan, Jika gagal maka akan kembali ke *form* mengisi data dan selanjutnya data tersebut akan disimpan ke *database*.

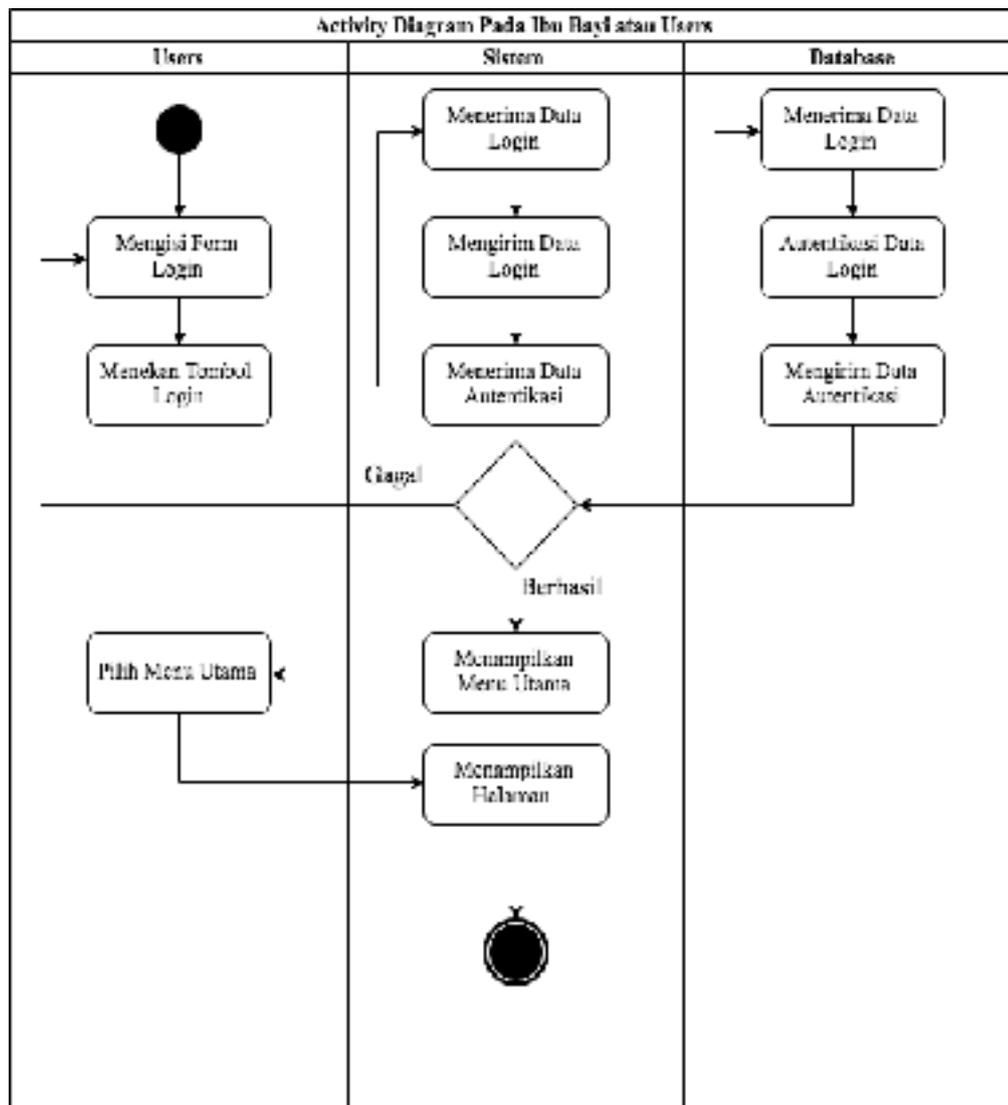
2. *Activity Diagram* pada Bidan

Activity diagram dibawah ini menggambarkan sistem dari sudut pandang sebagai bidan. Pada sistem ini bidan sebagai pengelola data bayi. Berikut merupakan *activity diagram* pada bidan terlihat pada gambar dibawah ini :

Pada gambar 3.5 diatas menjelaskan bahwa aktifitas diagram pada proses masuk ke dalam aplikasi, dimana ketika aplikasi dimulai maka secara otomatis akan menuju ke halaman *login*, lalu ke halaman *form login* ketika mengisi *form login* menekan tombol login lalu sistem menerima data *login* dan mengirim data *login* ke *database* lalu *database* nantinya akan mengirim autentikasi jika berhasil *login* menggunakan data pengguna selanjutnya akan menuju ke halaman utama aplikasi jika gagal maka akan kembali ke *form login* dan pada halaman menu utama terdapat data bayi yang dapat di pilih data nya yang akan ditambah, diedit atau dihapus. Jika data berhasil ditambah, edit dan hapus maka akan muncul pesan data yang tersimpan, Jika gagal maka akan kembali ke *form* mengisi data dan selanjutnya data tersebut akan disimpan ke *database*.

3. *Activity Diagram* pada Ibu Bayi

Activity diagram dibawah ini menggambarkan sistem dari sudut pandang sebagai ibu bayi atau *user*. Pada sistem ini ibu bayi atau *user* sebagai pengguna sistem. Berikut merupakan *use case diagram* pada ibu bayi terlihat pada gambar dibawah ini :



Gambar 3.6 Activity Diagram Pada Ibu Bayi Aplikasi E-Posyandu

Pada gambar 3.6 diatas menjelaskan bahwa aktifitas diagram pada proses masuk ke dalam aplikasi, dimana ketika aplikasi dimulai maka secara otomatis akan menuju ke halaman *login*, lalu ke halaman *form login* ketika mengisi *form login* menekan tombol login lalu sistem menerima data *login* dan mengirim data *login* ke *database* lalu *database* nantinya akan mengirim autentikasi jika

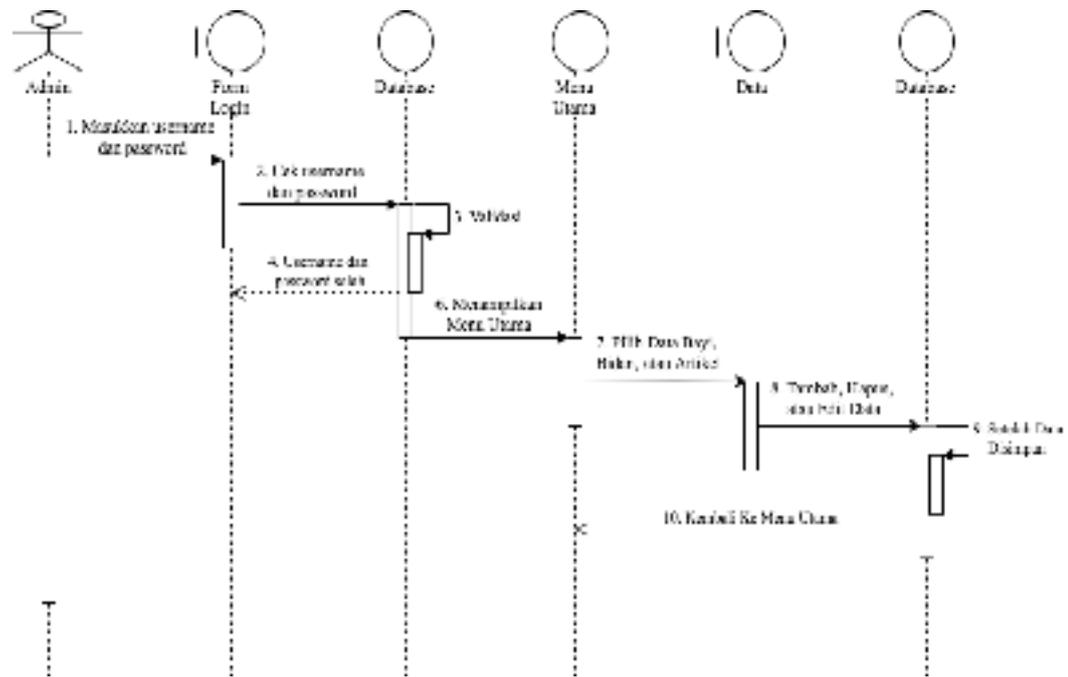
berhasil *login* menggunakan data pengguna selanjutnya akan menuju ke halaman utama aplikasi jika gagal maka akan kembali ke *form login* dan pada halaman menu utama terdapat beberapa menu seperti data bayi, statistik berat badan bayi, jadwal vaksin, artikel kesehatan dan *profile* bayi.

c. *Sequence Diagram*

Sequence diagram merupakan gambaran interaksi antar objek di dalam dan di sekitar sistem dan juga dapat menggambarkan kelakuan *user case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Pada aplikasi *monitoring* kesehatan balita melalui e-posyandu berbasis *hybrid programming* terdapat tiga *sequence diagram* yaitu sebagai berikut :

1. *Sequence Diagram* pada *Admin*

Sequence diagram dibawah ini menggambarkan sistem dari sudut pandang sebagai *admin*. Pada sistem ini *admin* sebagai pengelola data bayi, data bidan, dan data artikel. Berikut merupakan *sequence diagram* pada *admin* terlihat pada gambar dibawah ini :

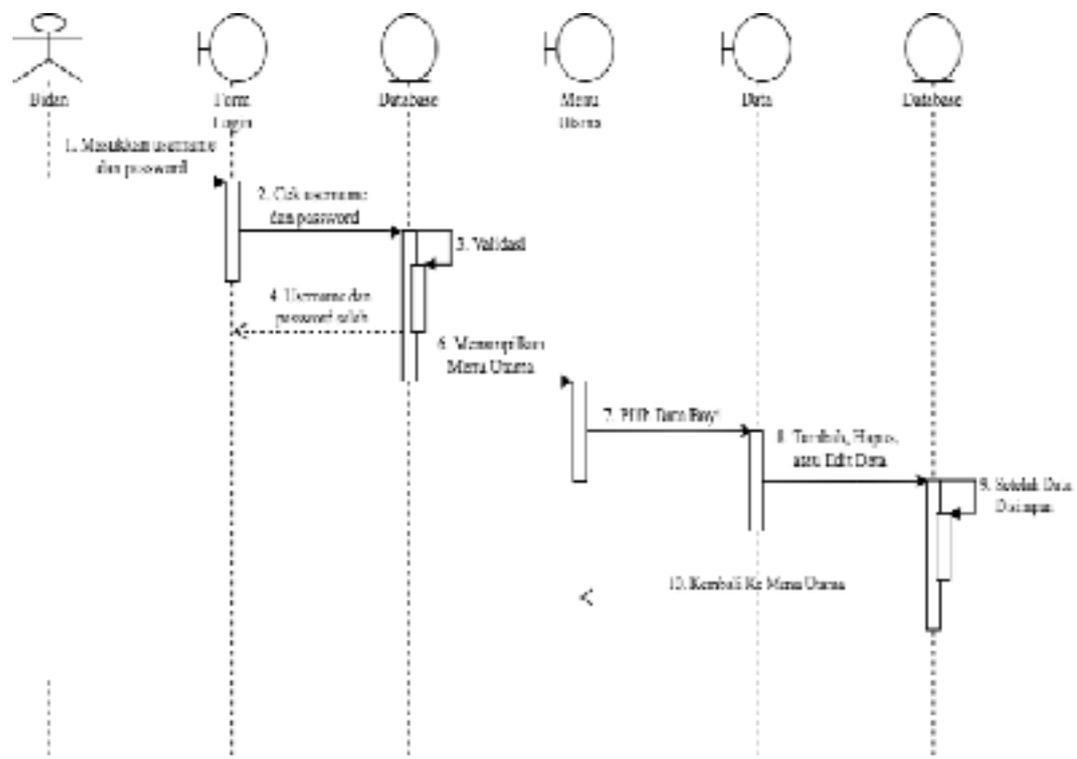


Gambar 3.7 Sequence Diagram Pada Admin Aplikasi E-Posyandu

Pada gambar 3.7 diatas, bisa dilihat bahwa yang menjadi aktor adalah *admin*. Pertama yang dilakukan oleh aktor atau *admin* yaitu memasukkan *username* dan *password* pada *form login*, selanjutnya akan mengecek *username* dan *password* setelah itu tunggu sampai pencarian data *admin* pada *database* selesai dan akan melakukan validasi jika *username* dan *password* yang dimasukkan salah maka akan kembali ke *form login* jika benar maka akan melanjutkan ke halaman utama lalu pilih data bayi, data bidan, atau data artikel setelah dipilih data dapat ditambah, edit, dan hapus selanjutnya data tersebut disimpan ke *database* dan kembali ke menu utama.

2. *Sequence Diagram* pada Bidan

Sequence diagram dibawah ini menggambarkan sistem dari sudut pandang sebagai bidan. Pada sistem ini bidan sebagai pengelola data bayi. Berikut merupakan *sequence diagram* pada bidan terlihat pada gambar dibawah ini :



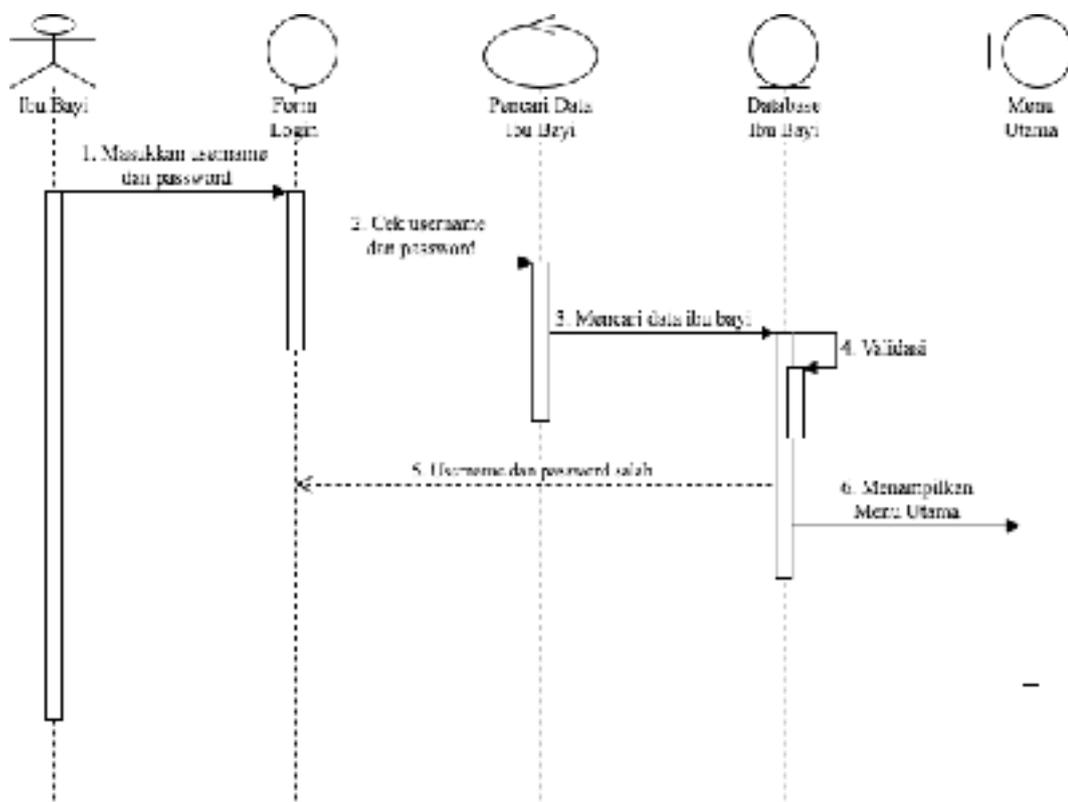
Gambar 3.8 *Sequence Diagram* Pada Bidan Aplikasi E-Posyandu

Pada gambar 3.8 diatas, bisa dilihat bahwa yang menjadi aktor adalah bidan. Pertama yang dilakukan oleh aktor atau bidan yaitu memasukkan *username* dan *password* pada *form login*, selanjutnya akan mengecek *username* dan *password* setelah itu tunggu sampai pencarian data bidan pada *database* selesai dan

akan melakukan validasi jika *username* dan *password* yang dimasukkan salah maka akan kembali ke *form login* jika benar maka akan melanjutkan ke halaman utama lalu pilih data bayi setelah dipilih data dapat ditambah, edit, dan hapus selanjutnya data tersebut disimpan ke *database* dan kembali ke menu utama.

3. *Sequence Diagram* pada Ibu Bayi

Sequence diagram dibawah ini menggambarkan sistem dari sudut pandang sebagai ibu bayi atau *user*. Pada sistem ini bidan sebagai pengelola data bayi. Berikut merupakan *sequence diagram* pada bidan terlihat pada gambar dibawah ini :

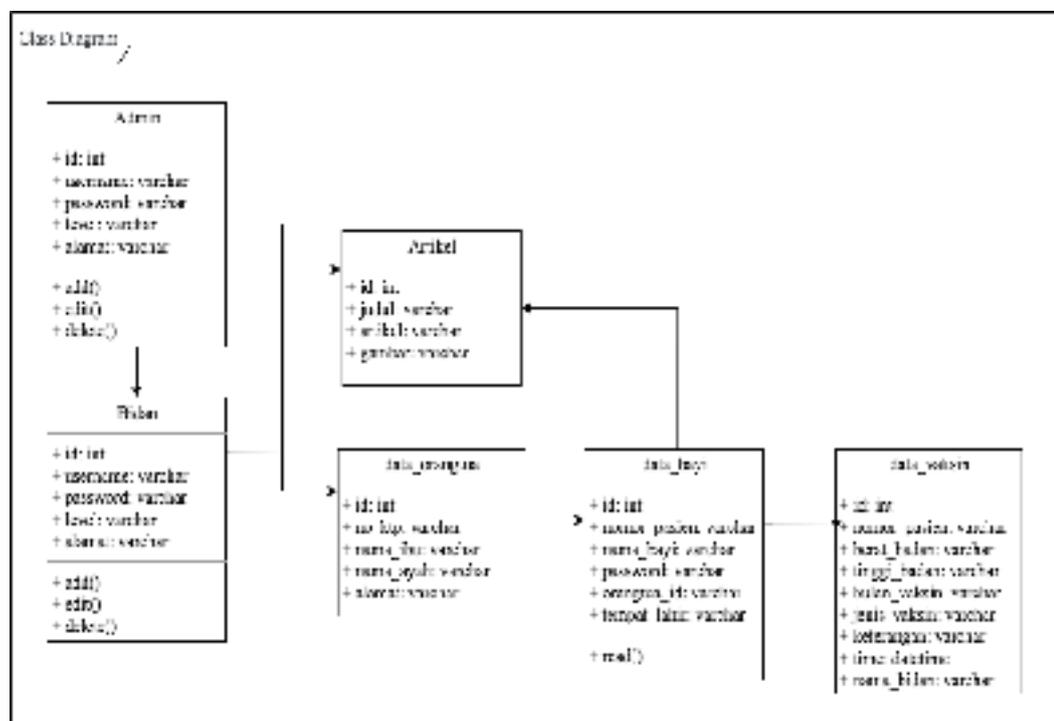


Gambar 3.9 *Sequence Diagram* Pada Ibu Bayi Aplikasi E-Posyandu

Pada gambar 3.9 diatas, bisa dilihat bahwa yang menjadi aktor adalah ibu bayi. Pertama yang dilakukan oleh aktor atau ibu bayi yaitu memasukkan *username* dan *password* pada *form login*, selanjutnya akan mengecek *username* dan *password* setelah itu tunggu sampai pencarian data ibu bayi pada *database* selesai dan akan melakukan validasi jika *username* dan *password* yang dimasukkan salah maka akan kembali ke *form login* jika benar maka akan melanjutkan ke halaman utama.

d. Class Diagram

Class diagram yaitu menggambarkan struktur kelas dan objek-objek yang akan dibangun dalam sebuah aplikasi. Adapun gambar penggunaan *class diagram* pada aplikasi yang akan dibangun adalah sebagai berikut :



Gambar 3.10 *Class Diagram* Aplikasi E-Posyandu

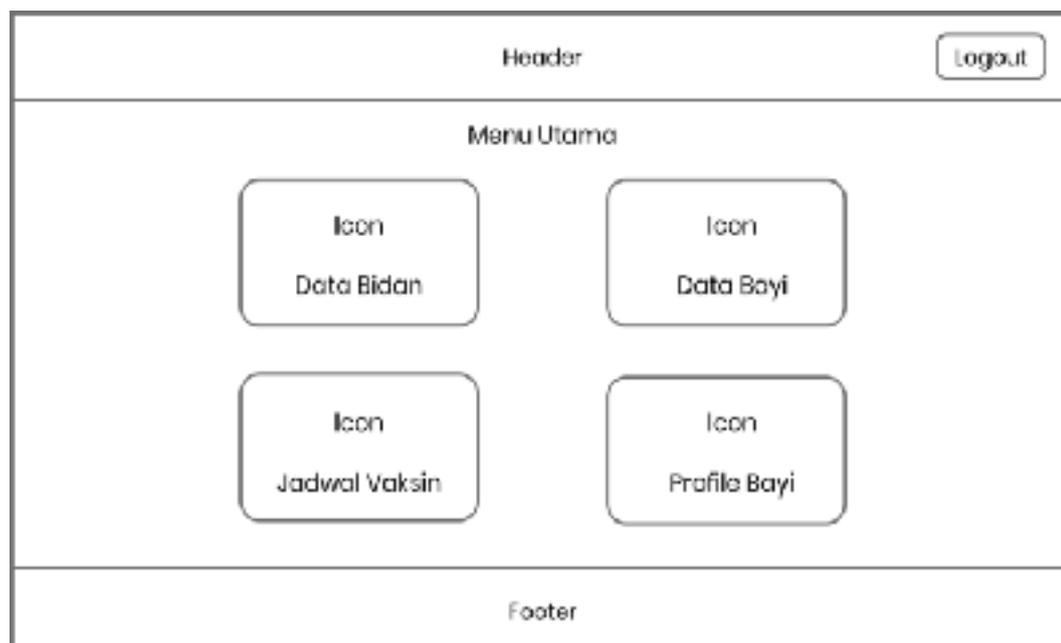
3. 6 Rancangan Tampilan Antarmuka

Rancangan tampilan antarmuka merupakan tampilan awal dari program yang merupakan mekanisme komunikasi antara pengguna atau *user* dengan sistem. Tampilan antarmuka dapat memberikan informasi dari sistem dan memberikan informasi kepada pengguna atau *user* untuk membantu mengarahkan alur penelusuran masalah sampai ditemukan solusi. Perancangan antarmuka juga merupakan gambaran tampilan awal ketika pengguna menjalankan aplikasi dan berikut merupakan rancangan antarmuka yang akan dibuat ketika *admin*, bidan, dan ibu bayi atau pengguna berhasil.

a. Rancangan Antarmuka Pada *Admin*

Adapun beberapa rancangan antarmuka halaman *admin* setelah berhasil *login* pada aplikasi yang akan dibangun adalah sebagai berikut :

1. Halaman Menu Utama



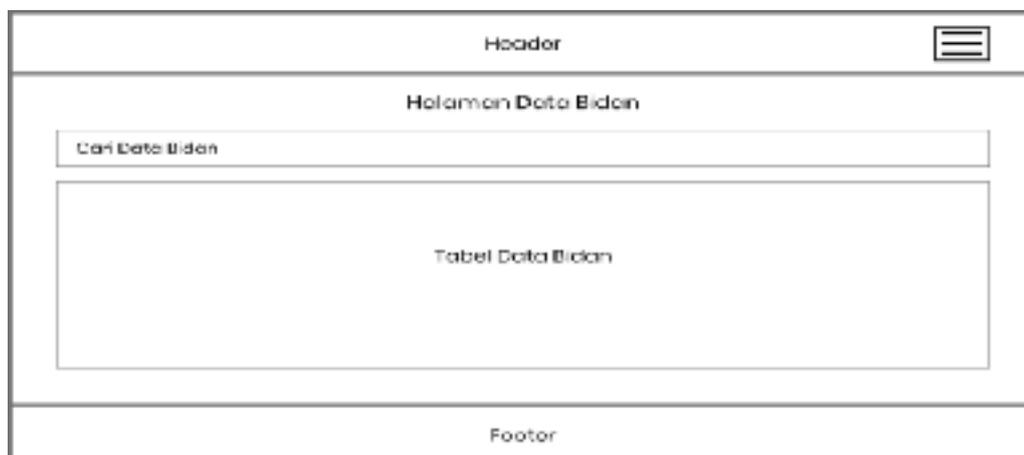
Gambar 3.11 Halaman Menu Utama Pada *Admin Di Website*



Gambar 3.12 Halaman Menu Utama Pada *Admin* Pada *Mobile*

Pada rancangan halaman *admin* setelah *login* terdapat beberapa menu-menu seperti halaman data bayi, data bidan, data artikel, dan jadwal vaksin. Pada bagian atas aplikasi atau *header* yang nantinya akan berisi logo aplikasi dan terdapat tombol *logout* yaitu keluar dari akun yang sudah kita *login* pada aplikasi. Pada bagian bawah aplikasi atau *footer* berisi informasi aplikasi seperti nama aplikasi dan keterangan tahun pembuatan.

2. Halaman Data Bidan



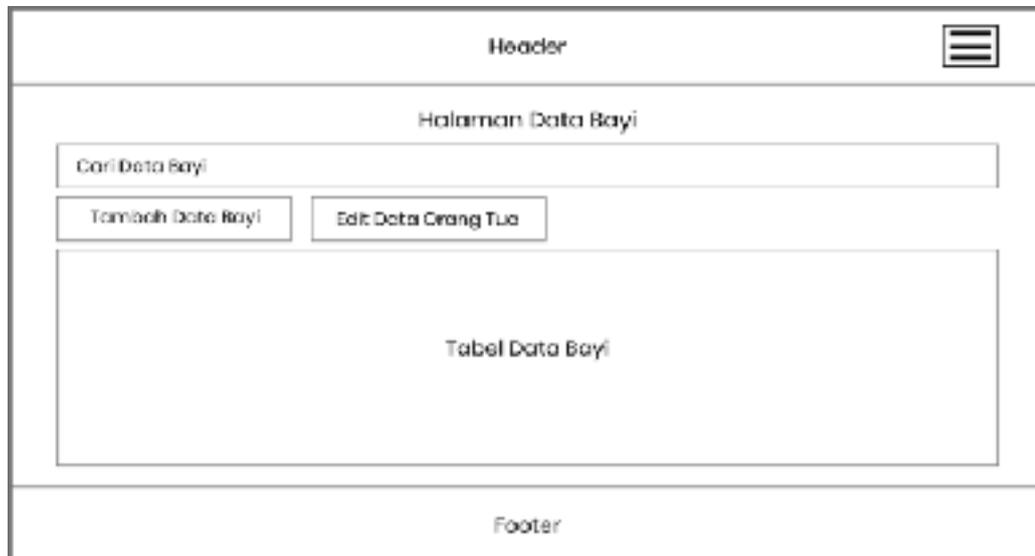
Gambar 3.13 Halaman Data Bidan Pada *Admin* Pada *Website*



Gambar 3.14 Halaman Data Bidan Pada *Admin Di Mobile*

Pada bagian atas aplikasi atau *header* yang nantinya akan berisi logo aplikasi dan terdapat tombol *icon* menu yang nantinya berisi dua pilihan yaitu kembali ke menu utama atau keluar dari akun yang sudah kita *login* pada aplikasi. Pada halaman data bidan terdapat kolom pencarian yang berguna untuk mencari data bidan dengan memasukkan data bidan seperti nomor bidan, nama bidan, dan alamat bidan dibawah kolom pencarian juga ada beberapa informasi pada tabel seperti nomor bidan, nama bidan dan alamat bidan dan juga terdapat tombol pada tabel tersebut yang berfungsi untuk menambah, edit dan hapus data bidan. Pada bagian bawah aplikasi atau *footer* berisi informasi aplikasi seperti nama aplikasi dan keterangan tahun pembuatan.

3. Halaman Data Bayi



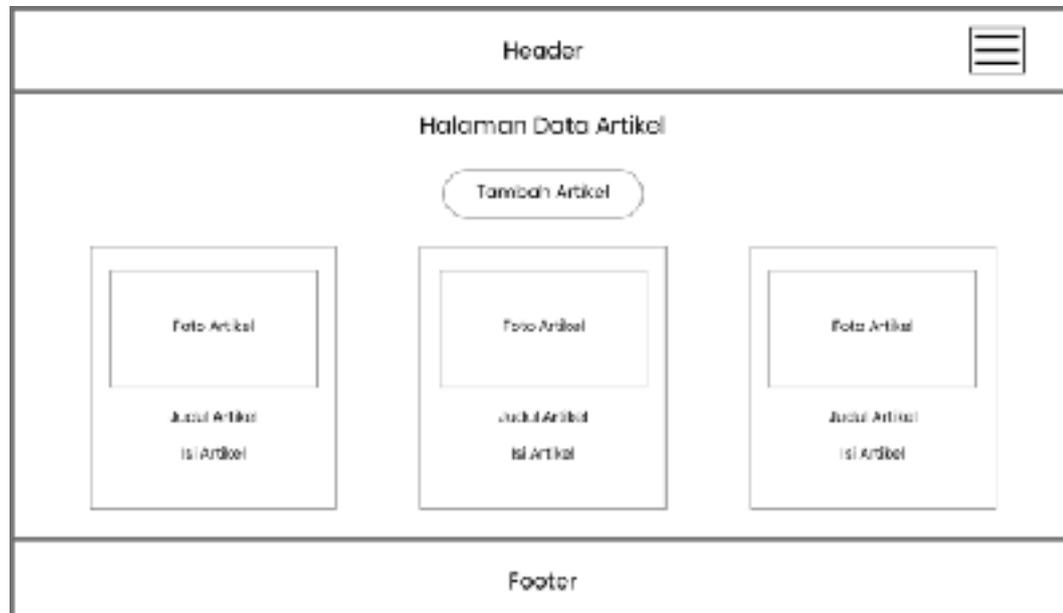
Gambar 3.15 Halaman Data Bayi Pada *Admin Di Website*



Gambar 3.16 Halaman Data Bayi Pada *Mobile*

Pada bagian atas aplikasi atau *header* yang nantinya akan berisi logo aplikasi dan terdapat tombol *icon* menu yang nantinya berisi dua pilihan yaitu kembali ke menu utama atau keluar dari akun yang sudah kita *login* pada aplikasi. Pada halaman data bayi terdapat kolom pencarian yang berguna untuk mencari data bayi dengan memasukkan data bayi seperti nama bayi, ibu bayi, dan ayah bayi. Di bawah kolom pencarian ini juga terdapat dua *button* yaitu tambah data bayi dan edit data orang tua dan beberapa informasi pada tabel seperti nama bayi, nama ibu bayi, nama ayah bayi dan juga terdapat tombol pada tabel tersebut yang berfungsi untuk edit dan hapus data bayi. Pada bagian bawah aplikasi atau *footer* berisi informasi aplikasi seperti nama aplikasi dan keterangan tahun pembuatan.

4. Halaman Data Artikel Kesehatan



Gambar 3.17 Halaman Data Artikel Pada *Admin Di Website*



Gambar 3.18 Halaman Data Artikel Pada *Admin Di Mobile*

Pada bagian atas aplikasi atau *header* yang nantinya akan berisi logo aplikasi dan terdapat tombol *icon* menu yang nantinya berisi dua pilihan yaitu kembali ke menu utama atau keluar dari akun yang sudah kita *login* pada aplikasi. Pada bagian bawah aplikasi atau *footer* berisi informasi aplikasi seperti nama aplikasi dan keterangan tahun pembuatan. Pada halaman data artikel terdapat beberapa informasi seperti foto artikel, judul artikel dan isi artikel juga terdapat tombol pada halaman tersebut yang berfungsi untuk menambah, edit dan hapus data artikel.

5. Halaman Jadwal Vaksin



Gambar 3.19 Halaman Jadwal Vaksin Pada *Admin Di Website*



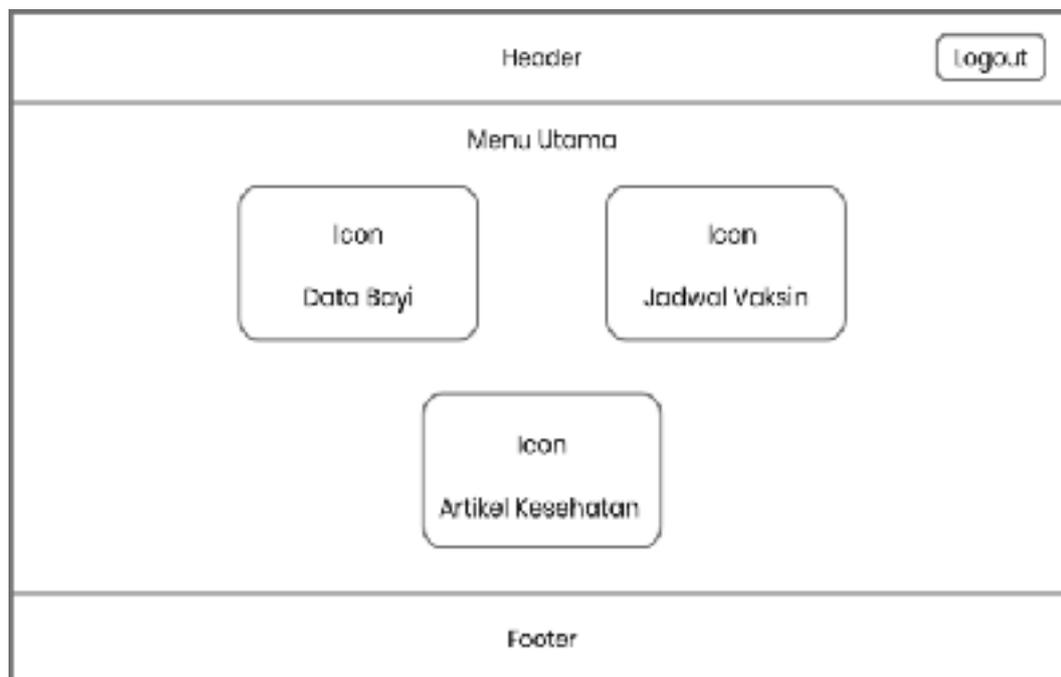
Gambar 3.20 Halaman Jadwal Vaksin Pada *Mobile*

Pada bagian atas aplikasi atau *header* yang nantinya akan berisi logo aplikasi dan terdapat tombol *icon* menu yang nantinya berisi dua pilihan yaitu kembali ke menu utama atau keluar dari akun yang sudah kita *login* pada aplikasi. Pada halaman jadwal vaksin terdapat beberapa informasi pada tabel seperti nama vaksin dan bulan vaksin. Pada bagian bawah aplikasi atau *footer* berisi informasi aplikasi seperti nama aplikasi dan keterangan tahun pembuatan.

b. Rancangan Antarmuka Pada Bidan

Adapun beberapa rancangan antarmuka halaman bidan setelah berhasil *login* pada aplikasi yang akan dibangun adalah sebagai berikut :

1. Halaman Utama



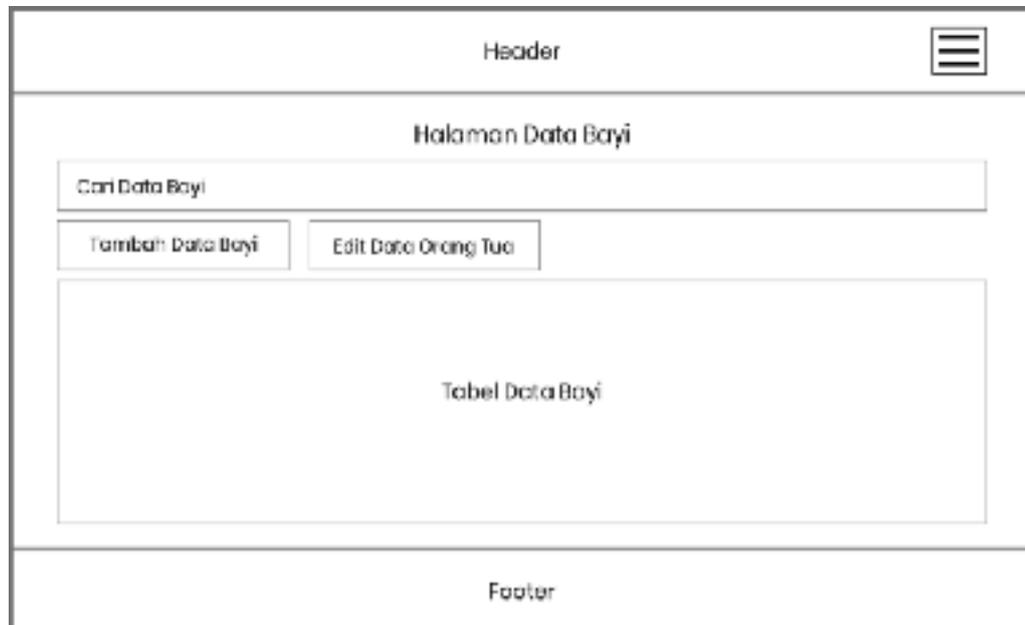
Gambar 3.21 Halaman Menu Utama Pada Bidan Di *Website*



Gambar 3.22 Halaman Menu Utama Pada Bidan Di *Mobile*

Pada rancangan halaman bidan setelah *login* terdapat beberapa menu-menu seperti halaman data bayi, jadwal vaksin bidan, dan data artikel. Pada bagian atas aplikasi atau *header* yang nantinya akan berisi logo aplikasi dan terdapat tombol *logout* yaitu keluar dari akun yang sudah kita *login* pada aplikasi. Pada bagian bawah aplikasi atau *footer* berisi informasi aplikasi seperti nama aplikasi dan keterangan tahun pembuatan.

2. Halaman Data Bayi



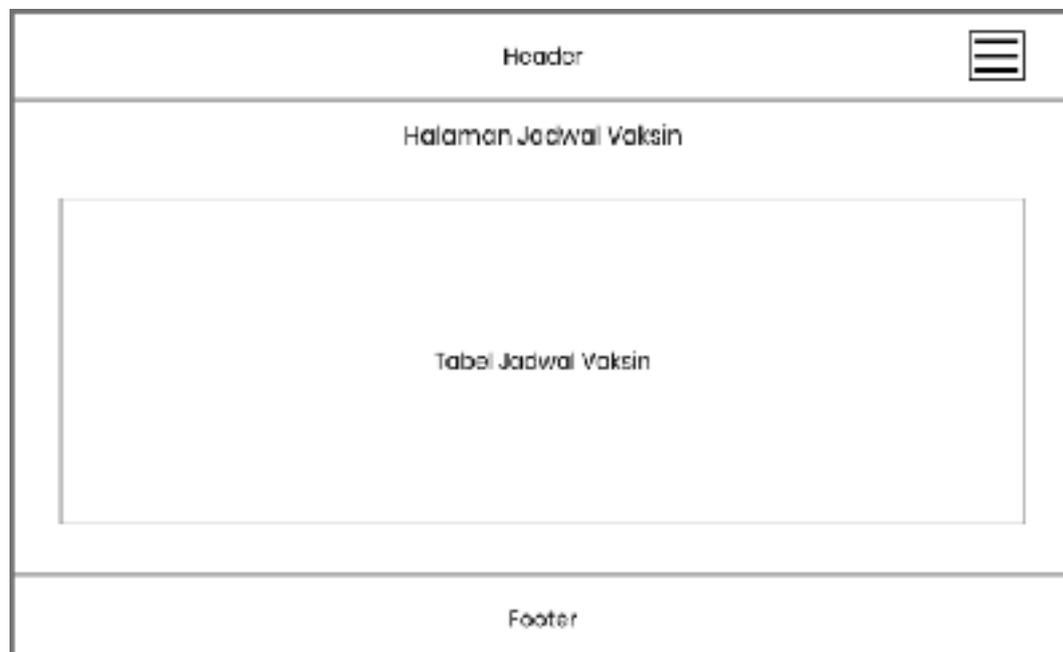
Gambar 3.23 Halaman Data Bayi Pada Bidan Di *Website*



Gambar 3.24 Halaman Data Bayi Pada Bidan Di *Mobile*

Pada bagian atas aplikasi atau *header* yang nantinya akan berisi logo aplikasi dan terdapat tombol *icon* menu yang nantinya berisi dua pilihan yaitu kembali ke menu utama atau keluar dari akun yang sudah kita *login* pada aplikasi. Pada halaman data bayi terdapat kolom pencarian yang berguna untuk mencari data bayi dengan memasukkan data bayi seperti nama bayi, ibu bayi, dan ayah bayi dibawah kolom pencarian ini juga terdapat dua *button* yaitu tambah data bayi dan edit data orang tua dan beberapa informasi pada tabel seperti nama bayi, nama ibu bayi, nama ayah bayi dan juga terdapat tombol pada tabel tersebut yang berfungsi untuk edit dan hapus data bayi. Pada bagian bawah aplikasi atau *footer* berisi informasi aplikasi seperti nama aplikasi dan keterangan tahun pembuatan.

3. Halaman Jadwal Vaksin



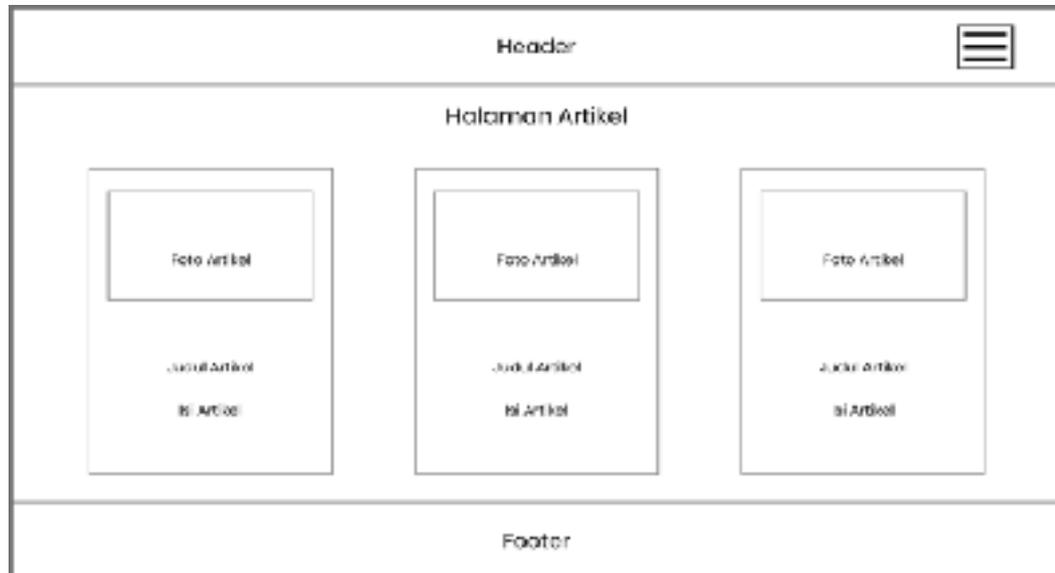
Gambar 3.25 Halaman Jadwal Vaksin Pada Bidan Di *Website*



Gambar 3.26 Halaman Jadwal Vaksin Pada Bidan Di *Mobile*

Pada bagian atas aplikasi atau *header* yang nantinya akan berisi logo aplikasi dan terdapat tombol *icon* menu yang nanti nya berisi dua pilihan yaitu kembali ke menu utama atau keluar dari akun yang sudah kita *login* pada aplikasi. Terdapat halaman jadwal vaksin terdapat beberapa informasi pada tabel seperti nama vaksin dan bulan vaksin. Pada bagian bawah aplikasi atau *footer* berisi informasi aplikasi seperti nama aplikasi dan keterangan tahun pembuatan.

4. Halaman Artikel Kesehatan



Gambar 3.27 Halaman Artikel Kesehatan Pada Bidan Di *Website*



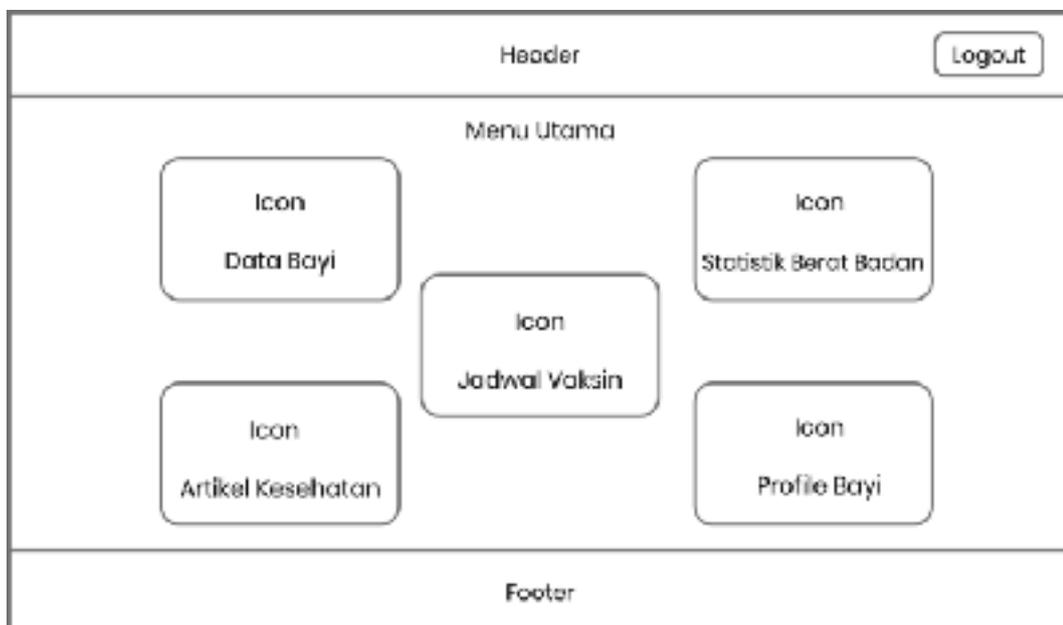
Gambar 3.28 Halaman Artikel Kesehatan Pada Bidan Di *Mobile*

Pada bagian atas aplikasi atau *header* yang nantinya akan berisi logo aplikasi dan terdapat tombol *icon* menu yang nantinya berisi dua pilihan yaitu kembali ke menu utama atau keluar dari akun yang sudah kita *login* pada aplikasi. Pada halaman artikel kesehatan terdapat beberapa informasi pada seperti foto artikel, judul artikel dan isi artikel kesehatan yang dapat dibaca. Pada bagian bawah aplikasi atau *footer* berisi informasi aplikasi seperti nama aplikasi dan keterangan tahun pembuatan.

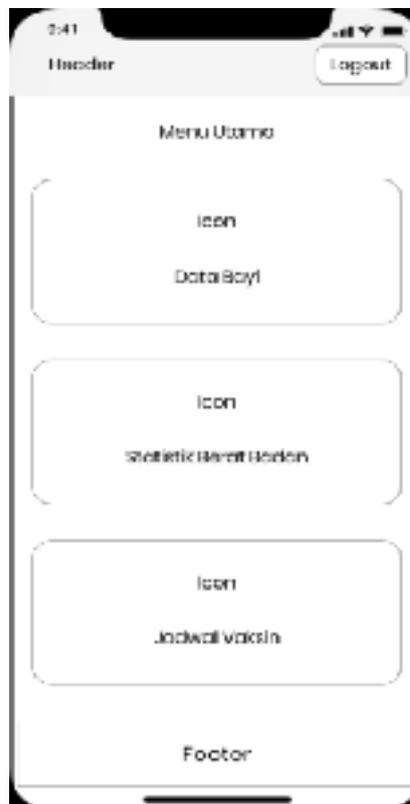
c. Rancangan Antarmuka Pada Ibu Bayi atau *Users*

Adapun beberapa rancangan antarmuka halaman ibu bayi atau *users* setelah berhasil *login* pada aplikasi yang akan dibangun adalah sebagai berikut :

1. Halaman Menu Utama



Gambar 3.29 Halaman Menu Utama Pada Ibu Bayi Di *Website*



Gambar 3.30 Halaman Menu Utama Pada Ibu Bayi Di *Mobile*

Pada rancangan halaman ibu bayi atau *users* setelah *login* terdapat beberapa menu-menu seperti halaman data bayi, jadwal vaksin, statistik berat badan, artikel kesehatan, dan *profile* bayi. Pada bagian atas aplikasi atau *header* yang nantinya akan berisi logo aplikasi dan terdapat tombol *logout* yaitu keluar dari akun yang sudah kita *login* pada aplikasi. Pada bagian bawah aplikasi atau *footer* berisi informasi aplikasi seperti nama aplikasi dan keterangan tahun pembuatan.

2. Halaman Data Bayi



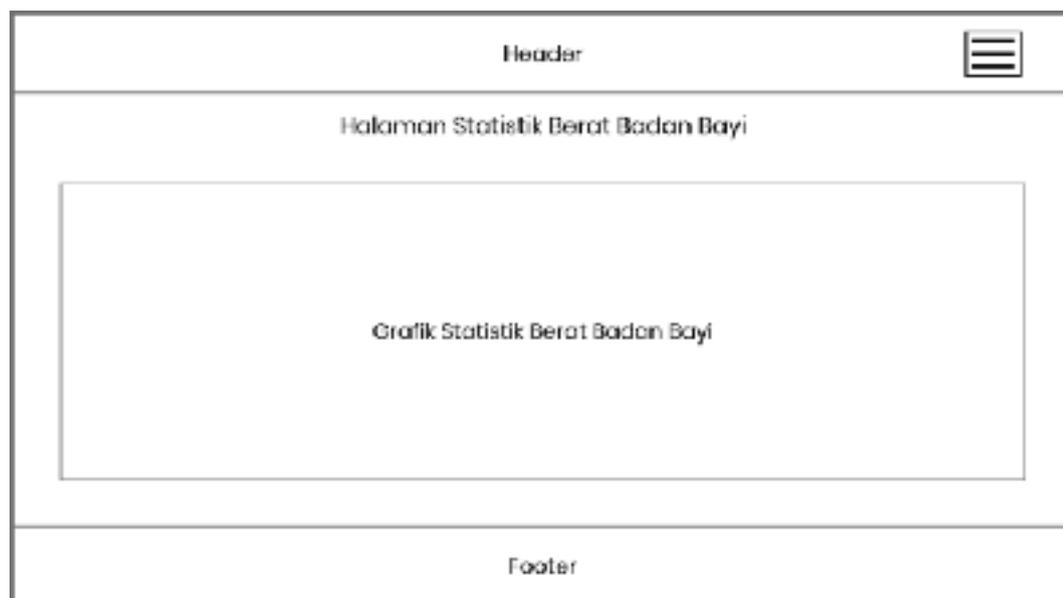
Gambar 3.31 Halaman Data Bayi Pada Ibu Bayi Di *Website*



Gambar 3.32 Halaman Data Bayi Pada Ibu Bayi Di *Mobile*

Pada bagian atas aplikasi atau *header* yang nantinya akan berisi logo aplikasi dan terdapat tombol *icon* menu yang nantinya berisi dua pilihan yaitu kembali ke menu utama atau keluar dari akun yang sudah kita *login* pada aplikasi. Pada halaman data bayi terdapat beberapa informasi pada tabel seperti nama bayi, nama ibu bayi, nama ayah bayi dan semua data ini didapat dari hasil posyandu. Pada bagian bawah aplikasi atau *footer* berisi informasi aplikasi seperti nama aplikasi dan keterangan tahun pembuatan.

3. Halaman Statistik Berat Badan Bayi



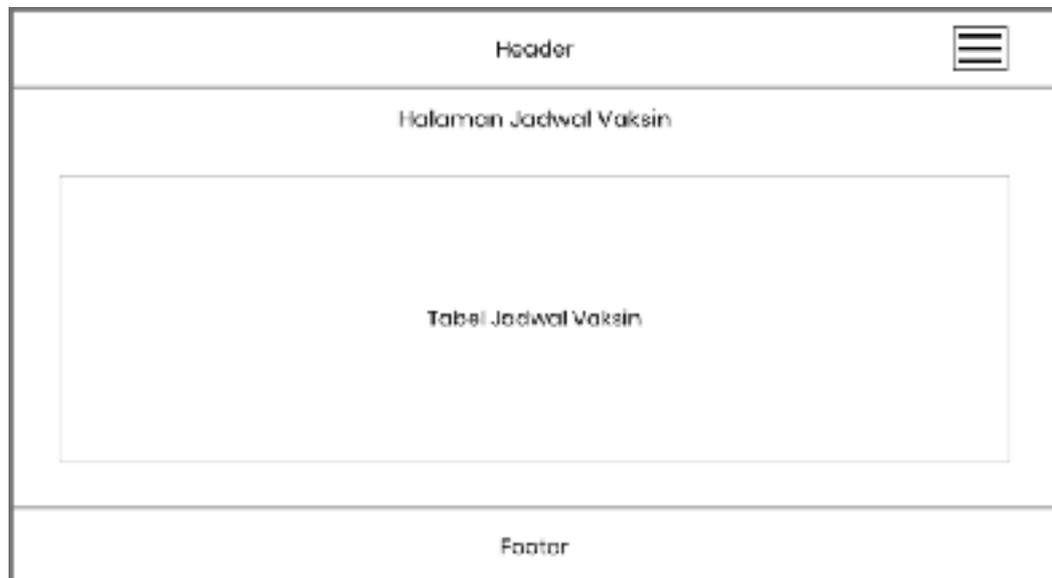
Gambar 3.33 Halaman Statistik Berat Badan Bayi Pada Ibu Bayi Di *Website*



Gambar 3.34 Halaman Statistik Berat Badan Bayi Pada Ibu Bayi Di *Mobile*

Pada bagian atas aplikasi atau *header* yang nantinya akan berisi logo aplikasi dan terdapat tombol *icon* menu yang nantinya berisi dua pilihan yaitu kembali ke menu utama atau keluar dari akun yang sudah kita *login* pada aplikasi. Pada halaman statistik berat badan terdapat beberapa informasi yang akan ditampilkan kedalam grafik data yang akan ditampilkan yaitu berat badan dan tinggi badan pada tiap bulannya semua data ini didapat dari hasil posyandu. Pada bagian bawah aplikasi atau *footer* berisi informasi aplikasi seperti nama aplikasi dan keterangan tahun pembuatan.

4. Halaman Jadwal Vaksin



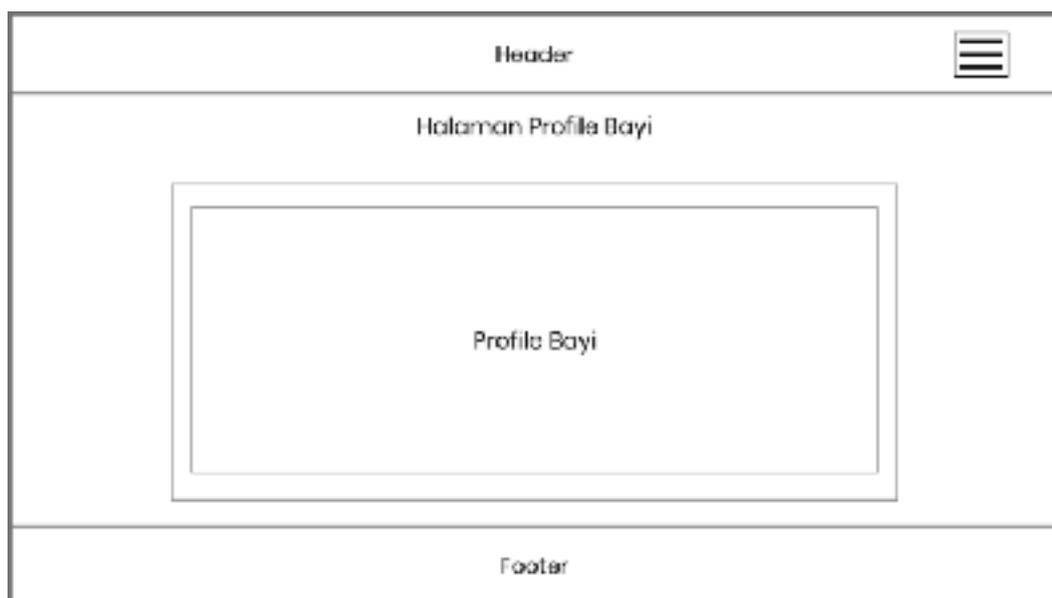
Gambar 3.35 Halaman Jadwal Vaksin Pada Ibu Bayi Di *Website*



Gambar 3.36 Halaman Jadwal Vaksin Pada Ibu Bayi Di *Mobile*

Pada bagian atas aplikasi atau *header* yang nantinya akan berisi logo aplikasi dan terdapat tombol *icon* menu yang nanti nya berisi dua pilihan yaitu kembali ke menu utama atau keluar dari akun yang sudah kita *login* pada aplikasi. Pada halaman jadwal vaksin terdapat beberapa informasi pada tabel seperti nama vaksin dan bulan vaksin informasi ini nantinya akan membantu ibu bayi untuk melihat anaknya bulan berikutnya akan diberikan vaksin apa. Pada bagian bawah aplikasi atau *footer* berisi informasi aplikasi seperti nama aplikasi dan keterangan tahun pembuatan.

5. Halaman Artikel Kesehatan



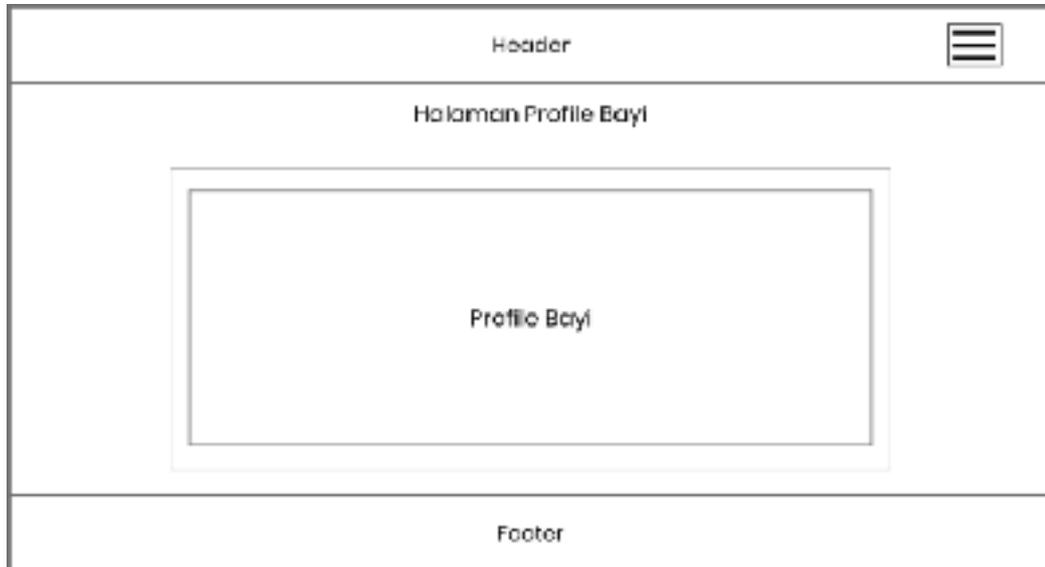
Gambar 3.37 Halaman Artikel Kesehatan Pada Ibu Bayi Di *Website*



Gambar 3.38 Halaman Artikel Kesehatan Pada Ibu Bayi Di *Mobile*

Pada bagian atas aplikasi atau *header* yang nantinya akan berisi logo aplikasi dan terdapat tombol *icon* menu yang nanti nya berisi dua pilihan yaitu kembali ke menu utama atau keluar dari akun yang sudah kita *login* pada aplikasi. Pada halaman artikel kesehatan terdapat beberapa informasi pada seperti foto artikel, judul artikel dan isi artikel kesehatan yang dapat dibaca. Pada bagian bawah aplikasi atau *footer* berisi informasi aplikasi seperti nama aplikasi dan keterangan tahun pembuatan.

6. Halaman *Profile* Bayi



Gambar 3.39 Halaman *Profile* Bayi Pada Ibu Bayi Di *Website*



Gambar 3.40 Halaman *Profile* Bayi Pada Ibu Bayi Di *Mobile*

Pada bagian atas aplikasi atau *header* yang nantinya akan berisi logo aplikasi dan terdapat tombol *icon* menu yang nanti nya berisi dua pilihan yaitu kembali ke menu utama atau keluar dari akun yang sudah kita *login* pada aplikasi. Pada halaman *profile* bayi terdapat beberapa informasi tentang *profile* bayi seperti nama bayi, ibu bayi, ayah bayi, dan alamat bayi. Pada bagian bawah aplikasi atau *footer* berisi informasi aplikasi seperti nama aplikasi dan keterangan tahun pembuatan.

BAB IV

HASIL DAN PEMBAHASAN

4.1 Kebutuhan Spesifikasi Minimum *Hardware* dan *Software*

Dalam perancangan dan pengujian aplikasi ini maka diperlukan syarat minimal perangkat keras dan perangkat lunak agar bisa berjalan optimal. Adapun spesifikasi *hardware* dan *software* yang dibutuhkan sebagai berikut :

a. *Hardware*

Adapun kebutuhan *hardware* yang digunakan untuk mendukung pengembangan sistem adalah sebagai berikut:

1. *Processor* : Intel Core i3 M390, 2,67 Ghz
2. *Memory* : 4 GB RAM
3. *Harddisk* : 500 GB

b. *Software*

Adapun kebutuhan *software* yang digunakan untuk mendukung pengembangan sistem adalah sebagai berikut:

1. Sistem Operasi : *macOS High Sierra*
2. *Software Database* : *XAMPP* Versi 7.1.33
3. *Software Design* : *Adobe XD*
4. Bahasa Pemrograman : *PHP* dan *JavaScript*

Gambar diatas merupakan tampilan halaman ketika mengakses atau masuk ke menu *profile* bayi dan terdapat beberapa informasi data seperti nama bayi, ayah bayi, ibu bayi dan alamat bayi.

4.3 Pengujian Aplikasi dan Pembahasan

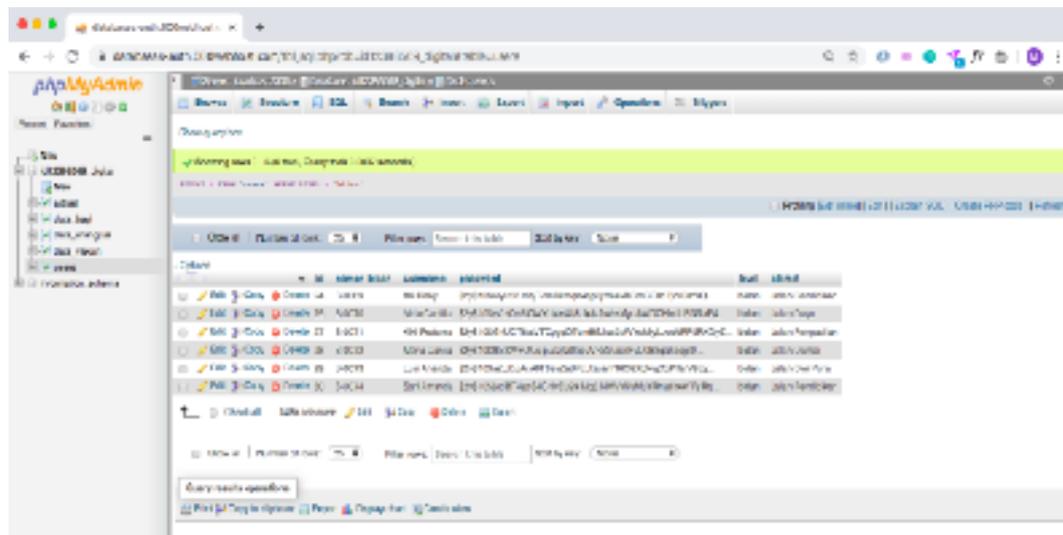
Adapun beberapa pengujian pada aplikasi yang akan dijalankan yaitu sebagai berikut :

a. Proses Pemasukan Data

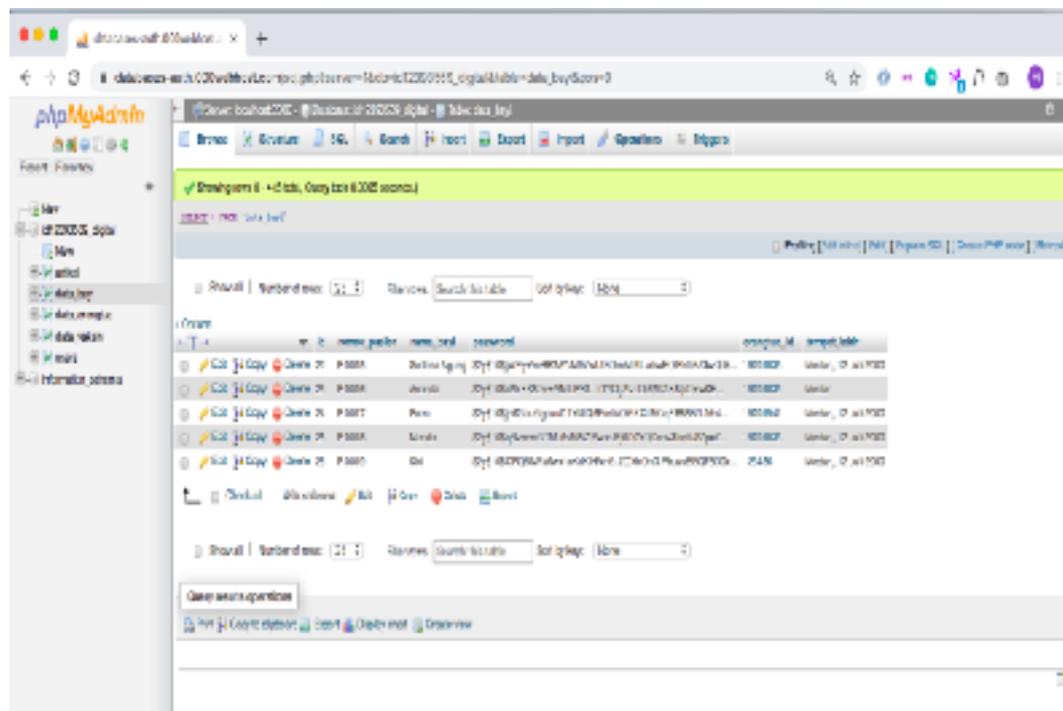
Berikut merupakan proses pemasukan data yang dilakukan oleh *admin* dan bidan :

1. Pemasukan Data Oleh *Admin*

Pada *database* data bidan dan ibu bayi atau *users* yang akan melakukan proses memasukkan data yaitu *admin*, sehingga nantinya data akan tampil pada aplikasi. Data bidan berupa nama bidan, *password* bidan, dan alamat bidan seperti gambar dibawah ini :



Gambar 4.54 Hasil *Input* Data Bidan Ke *Database* Oleh *Admin*



Gambar 4.56 Hasil *Input* Data Ibu Bayi Ke *Database* Oleh Bidan

b. Proses Transaksi Data

Ketika ibu bayi atau *users* melakukan *login* ke aplikasi maka terjadilah proses transaksi data dari *form login* ke *database*, aplikasi akan memeriksa apakah data yang diinputkan pengguna ada di *database*, jika ada maka selanjutnya akan menuju ke halaman menu utama. Adapun *form login* yang terdapat pada *users* sebagai berikut :

The image shows a desktop browser window with the URL `localhost:8080/web/indapa.com/loginbay.php`. The page has a blue header with the text "e-Pasyenda" and a "Logout" button. The main content area is titled "Halaman Login Data Bayi". Below the title, there are two input fields: "Nomor Pasien" with the placeholder text "Masukkan Nomor Pasien" and "Password" with the placeholder text "Masukkan Password". A blue "Login" button is positioned below the password field. At the bottom of the browser window, a black footer contains the text "e-Pasyenda - 2020".

Gambar 4.57 Tampilan *Form Login Users* Pada Website

The image shows a mobile phone screen displaying the same login page as in the desktop view. The phone's status bar at the top shows the time "10:22" and signal strength. The page layout is responsive, with the "e-Pasyenda" header and "Logout" button at the top. The title "Halaman Login Data Bayi" is centered. The input fields for "Nomor Pasien" and "Password" are also centered, with the "Login" button below them. The footer at the bottom of the screen shows "e-Pasyenda - 2020" and a small logo for "800webhost".

Gambar 4.58 Tampilan *Form Login Users* Pada Mobile

c. Proses Pelaporan Data

Setiap data yang telah di *input* oleh *admin* dan *bidan* jika berhasil data di *input* maka data akan tersimpan di database dan akan ditampilkan kembali jika pengguna membuka halaman data bayi. Adapun halaman data bayi yang terdapat pada *users* sebagai berikut :



No	Nama Janin	Nama Ibu	Nama Bapak/Ibu	Alamat	Tempat Tanggal Lahir	Berat Badan	Tinggi Badan	Bulan Tahun	Jenis Utsukin	Tanggal Utsukin	Keterangan
1	Sasena	DNI	Haris	Medan	Medan	31	111	11/1	Jadwal Utsukin	2020-11-02 15:30:00	Sehat
2	Sasena	DNI	Haris	Medan	Medan	41	441	11/1	Jadwal Utsukin	2020-11-04 23:35:42	Sehat
3	Sasena	DNI	Haris	Medan	Medan	45	451	11/1	Jadwal Utsukin	2020-11-04 23:35:42	Sehat

Gambar 4.59 Tampilan Data Bayi Pada *Website*



No	Nama Bayi	Nama Ibu	Nama Bapak/Ibu	Alamat	Keterangan
1	Sudora	DNI	Haris	Jalan...	
2	Sudora	DNI	Haris	Medan	
3	Sudora	DNI	Haris	Medan	

Tempat Tanggal Lahir	Berat Badan	Tinggi Badan	Jenis Utsukin	Keterangan
Medan	31	44	Jadwal Utsukin	Sehat
Medan	41	44	Jadwal Utsukin	Sehat
Medan	45	45	Jadwal Utsukin	Sehat

Gambar 4.60 Tampilan Data Bayi Pada *Mobile*

d. Proses Keamanan Data

Proses keamanan data pengguna sangat penting dalam pembuatan aplikasi ini, *database* yang digunakan ialah *database mysql* dengan bahasa pemrograman *php*. Pada setiap form input untuk login aplikasi ini sangat aman dengan menggunakan beberapa fitur oleh *php* seperti sintaks *mysql_real_escape_string* sintak ini berfungsi untuk mencegah seseorang yang akan melakukan *sql injection* pada aplikasi ini dan fitur lain seperti menambahkan sintak *htmlspecialchars* sintak ini berguna untuk menonaktifkan seluruh perintah-perintah yang terdapat pada *html*. Pada data akun pengguna pun dibuat sangat aman dengan menerapkan fitur *password hashing* menggunakan algoritma *bcrypt*, dalam algoritma *bcrypt* menggunakan teknik *hashing one way*.

nomor_pasien	nama_bayi	password	orangtua_id	tempat_lahir
P-0015	Sadana Agung	\$2y\$10\$ WF.yYreBR7AY/M9VwUIBOmM.KL.ahwEORNbK/GwQdk...	19701833	Medan, 12 Juli 2000
P-0016	Ananda	\$2y\$10\$d5frXK8DvwVMlbPR1.m1YOq2IzaDi67I43X8/pDYwKB...	19701833	Medan
P-0017	Putra	\$2y\$10\$jn6GauGgauul91YUtQ5FreIMOGKXLMcQPR58BI/Lbh4...	19701941	Medan, 12 Juli 2000
P-0018	Manda	\$2y\$10\$qNvkvmGTMJ6sM8iZiVwxs.Ej6D0YQDcnv3bvduS9pxZ...	19701833	Medan, 12 Juli 2000
P-0019	Kiki	\$2y\$10\$CPQBMFz6wx1oKdKIH/xhG.ZOXhOhG.PhuzsBEQF8D0x...	123456	Medan, 12 Juli 2000



Gambar 4.61 Tampilan Data *Password* Bayi

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan pembahasan dan penerapan aplikasi *monitoring* kesehatan balita melalui e-posyandu berbasis *hybrid programming*, maka dapat diambil kesimpulan sebagai berikut:

- a. Untuk membangun aplikasi *monitoring* kesehatan balita melalui e-posyandu dibangun dengan berbasis *Progressive Web Apps* (PWA).
- b. Untuk menerapkan *hybrid programming* pada aplikasi *monitoring* kesehatan balita melalui e-posyandu.

5.2 Saran

Berikut merupakan saran yang penulis berikan berdasarkan pembahasan dalam suatu aplikasi *monitoring* kesehatan balita melalui e-posyandu berbasis *hybrid programming*, yang akan diterapkan.

- a. Diharapkan aplikasi ini memiliki lebih banyak fitur didalamnya, seperti penambahan fitur seperti konsultasi langsung ke dokter atau bidan oleh karena itu diperlukan pengembangan lebih lanjut terhadap aplikasi ini.
- b. Dalam pengembangan selanjutnya, tampilan antarmuka perlu dibuat lebih menarik agar dapat meningkatkan minat *users* dalam menggunakan aplikasi ini.

- c. Dalam pengembangan selanjutnya, agar sistem dilengkapi dengan fitur notifikasi untuk mengingatkan ibu bayi setiap bulan untuk melakukan pergi posyandu.

DAFTAR PUSTAKA

- Pahlevi, O., Mulyani, A., & Khoir, M. (2018). Sistem Informasi Inventori Barang Menggunakan Metode Object Oriented Di Pt. Livaza Teknologi Indonesia Jakarta. *Jurnal PROSISKO*, 5(1). Retrieved from <https://livaza.com/>.
- Astiadi, P., & Kelana, O. H. (2014). Rancang Bangun Aplikasi Pembuatan Presentasi Berbasis Web Menggunakan HTML5. *Jurnal Buana Informatika*, 5(2), 115–126. <https://doi.org/10.24002/jbi.v5i2.359>
- Pradana, D. I., & Waspada, I. (2019). Aplikasi Hybrid Pada Sistem Informasi Penyewaan Buku. *Simetris: Jurnal Teknik Mesin, Elektro Dan Ilmu Komputer*, 10(1), 1–14. <https://doi.org/10.24176/simet.v10i1.2600>
- Navigasi, U., & Pertemuan, D. A. N. (2015). *ANALISIS DAN PERANCANGAN HYBRID APPLICATION*.
- Yatini B, I. (2014). Aplikasi pengolahan citra berbasis web menggunakan javascript dan jquery. *Jurnal Teknik*, 3(3), 1–8. Retrieved from <http://jurnalteknik.janabadra.ac.id/wp-content/uploads/2015/01/3-IndraYatini-April-2014.pdf>
- Saepuddin, E., Rizal, E., & Rusmana, A. (2018). Posyandu Roles as Mothers and Child Health Information Center. *Record and Library Journal*, 3(2), 201. <https://doi.org/10.20473/rlj.v3-i2.2017.201-208>
- Di, S., Palasari, P., Subang, K., Sessions, H. C., In, S., Health, P., ... Subang, O. F. (2014). Faktor-Faktor Yang Berhubungan Dengan Keaktifan Kader Dalam Kegiatan Posyandu (Studi Di Puskesmas Palasari Kabupaten Subang). *KESMAS - Jurnal Kesehatan Masyarakat*, 10(1), 73–79. <https://doi.org/10.15294/kemas.v10i1.3072>
- Adi, L., Akbar, R. J., & Khotimah, W. N. (2018). Platform e-Learning untuk Pembelajaran Pemrograman Web Menggunakan Konsep Progressive Web Apps. *Jurnal Teknik ITS*, 6(2), 2–6. <https://doi.org/10.12962/j23373539.v6i2.24291>
- Santoso, S., & Nurmalina, R. (2017). Perencanaan dan Pengembangan Aplikasi Absensi Mahasiswa Menggunakan Smart Card Guna Pengembangan Kampus Cerdas (Studi Kasus Politeknik Negeri Tanah Laut). *Jurnal Integrasi*, 9(1), 84–91.
- Badawi, A. (2018). Evaluasi Pengaruh Modifikasi Three Pass Protocol Terhadap Transmisi Kunci Enkripsi.
- Batubara, Supina. "Analisis perbandingan metode fuzzy mamdani dan fuzzy sugeno untuk penentuan kualitas cor beton instan." *IT Journal Research and Development* 2.1 (2017): 1-11.

- Bahri, S. (2018). *Metodologi Penelitian Bisnis Lengkap Dengan Teknik Pengolahan Data SPSS*. Penerbit Andi (Anggota Ikapi). Percetakan Andi Offset. Yogyakarta.
- Erika, Winda, Heni Rachmawati, and Ibnu Surya. "Enkripsi Teks Surat Elektronik (E-Mail) Berbasis Algoritma Rivest Shamir Adleman (RSA)." *Jurnal Aksara Komputer Terapan* 1.2 (2012).
- Fitriani, W., Rahim, R., Oktaviana, B., & Siahaan, A. P. U. (2017). Vernam Encrypted Text in End of File Hiding Steganography Technique. *Int. J. Recent Trends Eng. Res*, 3(7), 214-219.
- Hardinata, R. S. (2019). Audit Tata Kelola Teknologi Informasi menggunakan Cobit 5 (Studi Kasus: Universitas Pembangunan Panca Budi Medan). *Jurnal Teknik dan Informatika*, 6(1), 42-45.
- Hariyanto, E., Lubis, S. A., & Sitorus, Z. (2017). Perancangan prototipe helm pengukur kualitas udara. *KOMIK (Konferensi Nasional Teknologi Informasi dan Komputer)*, 1(1).
- Hariyanto, E., & Rahim, R. (2016). Arnold's cat map algorithm in digital image encryption. *International Journal of Science and Research (IJSR)*, 5(10), 1363-1365.
- Harumy, T. H. F., & Sulistianingsih, I. (2016). Sistem penunjang keputusan penentuan jabatan manager menggunakan metode mfep pada cv. Sapo durin. In *Seminar Nasional Teknologi Informasi dan Multimedia* (pp. 6-7).
- Iqbal, M., Siahaan, A. P. U., Purba, N. E., & Purwanto, D. (2017). Prim's Algorithm for Optimizing Fiber Optic Trajectory Planning. *Int. J. Sci. Res. Sci. Technol*, 3(6), 504-509.
- Marlina, L., Muslim, M., Siahaan, A. U., & Utama, P. (2016). Data Mining Classification Comparison (Naïve Bayes and C4. 5 Algorithms). *Int. J. Eng. Trends Technol*, 38(7), 380-383.
- Muttaqin, Muhammad. "ANALISA PEMANFAATAN SISTEM INFORMASI E-OFFICE PADA UNIVERSITAS PEMBANGUNAN PANCA BUDI MEDAN DENGAN MENGGUNAKAN METODE UTAUT." *Jurnal Teknik dan Informatika* 5.1 (2018): 40-43.
- Ramadhan, Z., Zarlis, M., Efendi, S., & Siahaan, A. P. U. (2018). Perbandingan Algoritma Prim dengan Algoritma Floyd-Warshall dalam Menentukan Rute Terpendek (Shortest Path Problem). *JURIKOM (Jurnal Riset Komputer)*, 5(2), 135-139.
- Rahim, R., Aryza, S., Wibowo, P., Harahap, A. K. Z., Suleman, A. R., Sihombing, E. E., ... & Agustina, I. (2018). Prototype file transfer protocol application for LAN and Wi-Fi communication. *Int. J. Eng. Technol.*, 7(2.13), 345-347.
- Wahyuni, Sri. "Implementasi Rapidminer Dalam Menganalisa Data Mahasiswa Drop Out." *Jurnal Abdi Ilmu* 10.2 (2018): 1899-1902.

Vardaro, M. J., Systems, H. I. T., AG, H. T., Jari, A., Pentti, M., Information, B. G., ... Measurements, C. (2016). No Title40–35), 1(2002, .بييب.
<https://doi.org/10.1109/ciced.2018.8592188>

Listing Program

Index.php

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Halaman Portal Login</title>
  <meta name="theme-color" content="#317EFB" />
  <link href="https://maxcdn.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css" rel="stylesheet">

  <!-- progressive web apps-->
  <!-- <script src="main.js"> </script>
  <script src="sw.js" > </script>
  <script src="pwa.js.js"> </script>
  <link rel="manifest" href="manifest.json"> -->

  <!-- icon support apple -->
  <link rel="apple-touch-icon" href="/images/icon-96x96.png" >
  <meta name="apple-mobile-web-app-status-bar" content="#FFE1C4">

</style>

@import
url('https://fonts.googleapis.com/css?family=Poppins&display=swap');

body {
  background: #f2f2f2;
  font-family: 'Poppins', sans-serif;
}

h3,
h1 {
  font-family: 'Poppins', sans-serif;
  /* color: #3282b8 */
}

.social-box .box {
  background: #FFF;
  border-radius: 10px;
```

```
padding: 40px 10px;
margin: 20px 0px;
cursor: pointer;
transition: all 0.5s ease-out;
}

.social-box .box:hover {
  box-shadow: 0 0 6px #4183D7;
}

.social-box .box {
  margin: 20px 0px;
  font-size: 15px;
  line-height: 30px;
}

.box {
  text-decoration: none;
  color: #4183D7;
  font-size: 15px;
}

.btn a {
  text-decoration: none;
}

.fa {
  color: #4183D7;
}

.bg {
  background-color: #3282b8;
}

footer {
  align-items: center;
  align-content: center;
  padding-top: 10px;
  font-family: 'Poppins', sans-serif;
}

.bg-4 {
  background-color: #2f2f2f;
  color: #ffffff;
  padding: 10px;
}
```

```

        font-family: 'Poppins', sans-serif;
    }

    .add-button {
        position: absolute;
        top: 1px;
        left: 1px;
    }
</style>
</head>

<body>

<nav class="navbar navbar-light bg">
    <a class="navbar-brand" >e-Posyandu</a>
    <form class="form-inline">

<h1 class="display-4 text-center">Portal Login </h1>

<div class="social-box">
    <div class="container">
        <div class="row">
            <div class="col-sm-2 col-xs-12 text-center">
            </div>
            <div class="col-sm-4 col-xs-12 text-center btn">
                <a href="loginadmin.php">
                    <div class="box">
                        <i class="fa fa-user" style="font-size:48px;"></i>
                        <div class="box-title">
                            <h3>Admin</h3>
                        </div>
                    </div>
                </a>
            </div>

            <div class="col-sm-4 col-xs-12 text-center btn">
                <a href="loginbayi.php">
                    <div class="box">
                        <i class="fa fa-users" style="font-size:48px" aria-
hidden="true"></i>
                        <div class="box-title">
                            <h3>Ibu Bayi</h3>
                        </div>
                    </div>
                </a>
            </div>
        </div>
    </div>
</div>

```

```
</div>

<div class="col-sm-2 col-xs-12 text-center">
</div>

</div>
</div>
</div>
<br>
<footer class="container-fluid bg-4 text-center">

  <p>e-Posyandu - 2020</p>

</footer>
</div>

<script src="https://code.jquery.com/jquery-3.4.1.slim.min.js"
integrity="sha384-
J6qa4849bIE2+poT4WnyKhv5vZF5SrPo0iEjwBvKU7imGFAV0wwj1yYfoRSJoZ+n"
crossorigin="anonymous">
  </script>
  <script
src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js"
integrity="sha384-
Q6E9RHvblyZFGft+2mJbHaEWldlvI9IOYy5n3zV9zzTtm13UksdQRvvoxMfooAo"
crossorigin="anonymous">
  </script>
  <script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.min.js"
integrity="sha384-
wfSDF2E50Y2D1uUdJ003uMBInjuUD4lh7YwaYd1iqfktj0Ued8GCExl3Og8ifwB6"
crossorigin="anonymous">
  </script>

</body>
</html>
```