

BAB II

TINJAUAN PUSATAKA DAN DASAR TEORI

2.1. Tinjauan Pustaka

Novitalia (2010) melakukan penelitian dengan judul Sistem Informasi Imunisasi Balita Studi Kasus: Dinkes Kota Batam. Sistem Informasi Imunisasi Balita digunakan untuk mengelola data imunisasi secara komputerisasi, melakukan perhitungan hasil imunisasi secara otomatis dan pemantauan kegiatan imunisasi secara online bagi orangtua.

Martina Novelia Nurendah Retnani(2014) melakukan penelitian dengan judul Analisis Dan Perancangan Aplikasi Pengingat Waktu Imunisasi Pada Bayi Di Puskesmas Gondokusuman 1 Yogyakarta. Sistem informasi pengingat kegiatan digunakan untuk mengingatkan kegiatan apa saja yang akan dilakukan baik kegiatan dalam waktu jangka pendek maupun jangka panjang. Sistem pengingat kegiatan ini diharapkan dapat membantu para orang tua untuk mengingatkan jadwal imunisasi bagi anak , sehingga kegiatan imunisasi anak dapat berjalan dengan efektif.Aplikasi pengingat ini menggunakan SMS yang akan dikirimkan kepada pengguna yang terdaftar pada sistem. Metode penelitian yang digunakan yaitu waterfall. Bahasa pemrograman yang digunakan yaitu Visual Basic 6.0 dan MySQL Database Server.

Evasaria Magdalena Sipayunga , Herastia Maharani , Tamsir H. Siraitc , Erik Gunawan (2018) melakukan penelitian dengan judul Perancangan Sistem Informasi Imunisasi Dasar Lengkap (IDL) Menggunakan Teknologi SMS Gateway. Perancangan sistem informasi imunisasi menyediakan catatan historis imunisasi yakni informasi imunisasi dan jadwal imunisasi yang akan dilakukan berikutnya. Implementasi sistem ini menggunakan teknologi SMS Gateway dapat mengingatkan (reminder) dan memperingatkan (warning) orang tua mengenai jadwal dan jenis imunisasi yang akan diterima berikutnya melalui teknologi SMS Gateway.

Tabel 2.1. Tinjauan Pustaka

Penulis	Objek Penelitian	Metode/Teknologi	Interface
Novitalia (2010)	Sistem Informasi Imunisasi Balita Studi Kasus: Dinkes Kota Batam.	Waterfall /Website	GUI
Martina Novelia Nurendah Retnani(2014)	Analisis Dan Perancangan Aplikasi Peningat Waktu Imunisasi Pada Bayi Di Puskesmas Gondokusuman 1 Yogyakarta	Waterfall / Desktop	GUI
Evasaria Magdalena Sipayunga(2018)	Perancangan Sistem Informasi Imunisasi Dasar Lengkap (IDL) Menggunakan Teknologi SMS Gateway	Waterfall / Desktop/SMS Gateway	GUI
Usulan Penelitian (2020)	Aplikasi Mobile Ibupedia Sebagai Media Informasi Perkembangan Anak Dan Panduan MPASI	Waterfall/ Mobile	GUI

Berdasarkan penelitian pada tabel 2.1 dapat di ketahui bahwa terdapat perbedaan antara penelitian sebelumnya, baik berupa perbedaan metode dan bahasa pemrograman dalam merancang aplikasi. Dalam penelitian ini media informasi ini dibangun menggunakan Android Studio, PHP dan MySQL.

2.2. Dasar Teori

Dasar teori merupakan bagian yang sangat penting. Hal ini dikarenakan pada bagian dasar atau landasan teori ini membahas apa saja teori-teori yang berhubungan dan berfungsi sebagai acuan atau pedoman agar penelitian tidak menyimpang dari tujuan awal yang telah ditetapkan.

2.2.1. Imunisasi

Imunisasi adalah salah satu cara mencegah penularan penyakit dengan memberikan vaksin agar kekebalan tubuh seseorang meningkat secara aktif terhadap penyakit tertentu. Metode pemberian vaksin beragam; ada yang diberikan dengan cara disuntikkan, ditetes ke dalam mulut, dan disemprotkan ke dalam mulut atau hidung. Ada jenis vaksin yang hanya diberikan sekali seumur hidup dan ada juga yang perlu diberikan secara berkala. Peraturan Menteri Kesehatan no. 12 tahun 2017 menjelaskan bahwa imunisasi wajib diberikan sesuai waktu yang telah ditetapkan dalam pedoman penyelenggaraan.

Jadwal Imunisasi Anak Usia 0 – 18 Tahun
Rekomendasi Ikatan Dokter Anak Indonesia (IDAI) Tahun 2017

Imunisasi	Usia																				
	Bulan												Tahun								
	Lahir	1	2	3	4	5	6	9	12	15	18	24	3	5	6	7	8	9	10	12	18
Hepatitis B	1		2	3	4																
Polio	0		1	2	3					4											
BCG	1 kali																				
DTP			1	2	3					4				5						6 (Td/Tdap)	7 (Td)
Hib			1	2	3					4											
PCV			1		2				3		4										
Rotavirus			1		2			3*													
Influenza	Ulangan 1 kali setiap tahun																				
Campak								1			2				3						
MMR										1					2						
Tifoid	Ulangan setiap 3 tahun																				
Hepatitis A	2 kali, interval 6 – 12 bulan																				
Varisela	1 kali																				
HPV	2 atau 3 kali*																				
Japanese encephalitis									1					2							
Dengue	3 kali, interval 6 bulan																				

Gambar 2.1 Standar Jadwal Imunisasi menurut IDAI tahun 2017

1.2. Basis Data

Basis data atau *database* merupakan kumpulan data yang saling berhubungan satu dengan yang lainnya, tersimpan di perangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya. Basis data merupakan salah satu komponen yang penting dalam sistem informasi, karena merupakan basis data dalam menyediakan informasi bagi para pemakai. Jenjang data yang diperlukan dalam membentuk basis data adalah :

1. *Characters* merupakan bagian data terkecil, yang dapat berupa karakter *numeric*, huruf ataupun karakter-karakter khusus (*special characters*) yang membentuk suatu item data *field*.
2. *Field* mempresentasikan suatu atribut dari *record* yang menunjukkan suatu item dari data, seperti misalnya nama, alamat dan lain sebagainya. Kumpulan dari *field* yang membentuk suatu *record*.

3. *Record* menggambarkan suatu unit data individu yang tertentu. Kumpulan dari *record* membentuk suatu file.
4. File terdiri dari *record-record* yang menggambar suatu kesatuan data yang sejenis.
5. *Database* kumpulan dari file/tabel membentuk suatu *database*. (Fathansyah, 1999)

2.2.3. MySQL

MySQL adalah *database server relasional* yang gratis dibawah *lisensi General Public License* (GNU). Dengan sifatnya yang *open source*, memungkinkan juga user untuk melakukan modifikasi pada *source code* untuk memenuhi kebutuhan spesifik mereka sendiri. MySQL merupakan *database server multi-user* dan *multitrade* yang tangguh (*robust*). Dengan memiliki banyak fitur MySQL bisa bersaing dengan *database komersil* sekalipun. Tidak mengejutkan, MySQL bisa menjadi *database* pilihan untuk banyak pengguna PHP. MySQL dikembangkan oleh 14 MySQL AB, sebuah perusahaan *komersil* yang membangun layanan bisnisnya melalui *database* MySQL. (Ferrar Utdirartatmo, 2001)

2.2.4. Unified Modelling Language








Unified Modelling Language (UML) adalah salah satu alat bantu yang sangat handal di dunia pengembangan sistem yang berorientasi objek (Munawar, 2005). Diagram UML yang digunakan dalam penelitian ini berjumlah 4 buah karena telah cukup untuk menjelaskan tentang sistem yang akan dibangun. Diagram-diagram tersebut adalah :


2.2.4.1 Use Case Diagram

Menurut Dharwiyanti, (2003) *Use case diagram* menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem.

Use case merupakan sebuah pekerjaan tertentu. Notasi yang digunakan dalam *use case diagram* dapat dilihat pada tabel 2.2.

Tabel 2.2 Notasi Komponen Diagram *Use Case Diagram*







Notasi	Nama Notasi	Fungsi
	<i>Package</i>	Merupakan kontainer atau wadah konseptual yang digunakan untuk mengelompokkan elemen-elemen dari sistem yang sedang dibangun.
	<i>Actor</i>	Menambah aktor baru pada diagram. Aktor ini menggambarkan apa yang harus diperbuat oleh aplikasi perangkat lunak.
	<i>Use Case</i>	Menambah <i>use case</i> pada diagram. <i>Use case</i> merupakan notasi dalam UML untuk menjelaskan urutan kegiatan yang dilakukan aktor dan system untuk mencapai suatu tujuan tertentu. <i>Use case</i> hanya menjelaskan apa yang dilakukan oleh aktor dan sistem, bukan bagaimana aktor dan sistem melakukan kegiatan tersebut.
	<i>Dependency</i>	Merupakan relasi yang memberikan penjelasan bahwa perubahan yang terjadi pada salah satu elemen memberi pengaruh pada elemen lain.
	<i>Association</i>	Menggambarkan relasi antara aktor dan <i>use case</i> .
	<i>Generalization</i>	Menggambarkan relasi lanjut antar <i>use case</i> atau menggambarkan struktur pewarisan antar aktor.
	<i>Note/Comment</i>	<i>Note</i> digunakan untuk memberikan keterangan dan komentar tambahan dari suatu elemen sehingga bisa langsung terlampir dalam model. <i>Note</i> dapat ditempelkan ke semua notasi yang lain.



	<i>Anchor to note / Link Comment</i>	Melekatkan catatan pada <i>use case</i> atau aktor tertentu dalam diagram.
---	--------------------------------------	--

2.2.4.2 Activity Diagram

Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity* diagram juga dapat menggambarkan proses *parallel* yang mungkin terjadi pada beberapa eksekusi.

Tabel 2.3 Notasi Komponen Diagram *Activity Diagram*




Notasi	Nama Notasi	Fungsi
	<i>Note / Comment</i>	<i>Note</i> digunakan untuk memberikan keterangan dan komentar tambahan dari suatu elemen sehingga bisa langsung terlampir dalam model. <i>Note</i> dapat ditempelkan ke semua notasi yang lain.
	<i>Activity</i>	Aktivitas merupakan notasi UML yang berfungsi untuk menunjukkan kegiatan atau aktivitas yang terjadi dalam suatu rangkaian proses atau operasi.
	<i>Start State / Initial Node</i>	<i>Start State</i> menunjukkan keadaan awal suatu proses dalam diagram <i>activity</i> atau memulai sebuah <i>events</i> dalam sebuah transisi pada <i>state chart</i> .
	<i>End State / Activity Final Node</i>	<i>End State</i> menggambarkan dari akhir suatu proses dari <i>terminate static</i> pada <i>state chart</i> .
	<i>State Transition</i>	Menambahkan transisi dari suatu aktifitas ke aktifitas lain.
	<i>Decision</i>	Menambahkan titik keputusan pada aliran kerja





	<i>Horizontal Synchronization</i>	Menambahkan sinkronisasi horizontal pada diagram
	<i>Vertical Synchronization</i>	Menambahkan sinkronisasi vertikal pada diagram

2.2.4.3 Sequence Diagram

Diagram *sequence* menjelaskan aspek dinamis dari sistem yang akan dibangun. (Schmuller, 1999) Notasi-notasi yang digunakan di dalam UML untuk membangun *Sequence Diagram* dapat dilihat pada table 2.4 .

Tabel 2.4 Notasi Komponen Diagram *Sequence Diagram*



Notasi	Nama Notasi	Fungsi
	<i>Note / Comment</i>	<i>Note</i> digunakan untuk memberikan keterangan dan komentar tambahan dari suatu elemen sehingga bisa langsung terlampir dalam model. <i>Note</i> dapat ditempelkan ke semua notasi yang lain.
	<i>Package</i>	Merupakan kontainer atau wadah konseptual yang digunakan untuk mengelompokkan elemen-elemen dari sistem yang sedang dibangun. Dari hal ini bisa dibuat model yang lebih sederhana. Tujuan dari <i>package</i> ini untuk mempermudah penglihatan (<i>visibility</i>) dari model yang sedang dibangun.
	<i>Object / Lifeline</i>	Menambahkan objek baru pada diagram. Setiap objek memiliki garis hidup (<i>lifeline</i>) yang digambarkan sebagai garis vertikal di bawah nama objek.







	<i>Object Message</i>	Menggambarkan pesan (<i>message</i>) antar dua objek.
	<i>Message to Self</i>	Menggambarkan pesan (<i>message</i>) yang menuju objek itu sendiri.
	<i>Return Message</i>	Menggambarkan pengembalian dari pemanggilan prosedur.
	<i>Anchor to Note / Link Comment</i>	Melekatkan catatan pada objek tertentu dalam diagram.


2.2.4.4 Class Diagram

Class diagram mnggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi dan lain-lain. Notasi yang digunakan dalam class diagram dapat dilihat pada table 2.5.

Tabel 2.5 Notasi Komponen Diagram *Class Diagram*

Notasi	Nama Notasi	Fungsi
	<i>Note/Comment</i>	<i>Note</i> digunakan untuk memberikan keterangan dan komentar tambahan dari suatu elemen sehingga bisa langsung terlampir dalam model. <i>Note</i> dapat ditempelkan ke semua notasi yang lain
	<i>Package</i>	Merupakan kontainer atau wadah konseptual yang digunakan untuk mengelompokkan elemen-elemen dari sistem yang sedang dibangun. Dari hal ini bisa dibuat model yang lebih sederhana. Tujuan dari <i>package</i>

		ini untuk mempermudah penglihatan (<i>visibility</i>) dari model yang sedang dibangun.
 <p>A UML class diagram template showing a rectangular box divided into three horizontal sections. The top section is labeled 'Nama_Kelas' (Class Name). The middle section is labeled 'Attributes' (Atribut). The bottom section is labeled 'Operations' (Operasi).</p>	<i>Class</i>	Kelas berupa kumpulan obyek yang memiliki atribut dan operasi. Kelas digunakan untuk mengimplementasikan interface dan mengabstraksikan elemen-elemen dari system yang sedang dibangun. Kelas dapat pula digunakan untuk mempresentasikan perangkat lunak maupun perangkat keras, baik konsep maupun benda nyata.
 <p>A horizontal dashed line ending in an open arrowhead pointing to the right.</p>	<i>Dependency</i>	Merupakan relasi yang memberikan penjelasan bahwa perubahan yang terjadi pada salah satu elemen memberi pengaruh pada elemen lain.
 <p>A simple horizontal solid line.</p>	<i>Association</i>	<i>Association</i> menggambarkan hubungan antar kelas. Dalam suatu komunikasi <i>unidirectional association</i> , akhir dari arah panah mengindikasikan siapa atau apa yang menerima komunikasi
 <p>A horizontal solid line ending in a closed arrowhead pointing to the right.</p>	<i>Generalization</i>	<i>Generalization</i> menunjukkan hubungan antar elemen yang lebih umum ke elemen yang lebih khusus atau spesifik.
 <p>A horizontal dashed line.</p>	<i>Anchor to note / Link Comment</i>	Melekatkan catatan pada kelas tertentu dalam diagram.
 <p>A horizontal dashed line ending in an open arrowhead pointing to the right.</p>	<i>Realization</i>	Realization menunjukkan bahwa hubungan elemen yang ada dibagin tanpa panah akan merealisasikan apa yang dinyatakan oleh elemen yang ada di bagian dengan panah.

 <p>The diagram shows a rectangular box representing an interface. At the top, it contains the text '<<interface>>'. Below this, the name 'Nama_Interface' is written in bold. Underneath the name, there are two sections: 'Attributes' and 'Operations', each in a separate row.</p>	<p><i>Interface</i></p>	<p>Interface merupakan kumpulan operasi tanpa implementasi dari suatu kelas. Implementasi operasi dalam interface dijabarkan oleh operasi dalam kelas.</p>
---	-------------------------	--