



BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1 Tinjauan Pustaka

Sebagai acuan dalam pengembangan Sistem Informasi Penjadwalan Imunisasi Berbasis Website Dengan SMS *Gateway*, maka penulis mencari beberapa referensi dari penelitian terdahulu yang memiliki keterkaitan secara langsung maupun tidak langsung terhadap penelitian penulis.

Penelitian yang dilakukan oleh Wisda Liani pada tahun 2020 dengan judul “Sistem Informasi Peningkat Jadwal Imunisasi Anak Dan Ibu Hamil Pada Posyandu Desa Sungai Paku Berbasis Web Menggunakan SMS *Gateway*”. Permasalahan dari penelitian ini dikarenakan tidak adanya pengingat dalam penyampaian jadwal terkait dengan proses pelaksanaan imunisasi anak dan ibu hamil pada Posyandu Desa Sungai Paku sehingga dapat menghambat dalam kegiatan pemberian imunisasi. Berdasarkan permasalahan tersebut penulis membangun aplikasi untuk proses pelaksanaan imunisasi anak dan ibu hamil dengan pengingat waktu menggunakan SMS *Gateway* serta Website sebagai perangkat lunak dalam membangun aplikasi. Dengan dibangunnya aplikasi tersebut diharapkan dapat membantu dalam penyampaian informasi jadwal imunisasi sehingga proses pelaksanaan imunisasi dapat berjalan dengan efektif. Pengembangan sistem menggunakan metode *prototype* [5].

Penelitian selanjutnya dilakukan oleh Ali Akbar Mushavi, Dewi Primasari, Jejen Jaenudin tahun 2019 dengan judul “Sistem Informasi Penjadwalan Kegiatan Posyandu Berbasis Web dan *Whatsapp Gateway*” untuk proses pengumpulan datanya dilakukan dengan metode wawancara, studi data dan studi pustaka. Proses pemberitahuan jadwal kegiatan posyandu masih menggunakan cara manual yaitu dengan menyampaikan menggunakan pengeras suara pada masjid atau mushola yang memungkinkan terjadinya kesalahan menerima informasi yang menyebabkan orang tua dari anak tidak dapat menghadiri kegiatan imunisasi di posyandu. Dengan permasalahan tersebut maka penulis membuat sebuah sistem informasi yang dapat membantu bidan pada desa dalam proses pemberitahuan jadwal posyandu dan membantu orang tua untuk mengetahui jadwal terbaru kegiatan posyandu yang akan

dilaksanakan. Sistem ini menggunakan *website* dalam proses pengolahan data dan whatsapp *gateway* untuk mengirimkan informasi jadwal posyandu. Sistem ini dirancang menggunakan bahasa pemrograman HTML, CSS, dan PHP. Pengembangan sistem menggunakan metode waterfall [6].

Penelitian selanjutnya dilakukan oleh Pinarti Puspitarini, dan Fendi Hidayat, S.T., M.Kom pada tahun 2021 dengan judul “Sistem Informasi Pendataan Dan Penjadwalan Imunisasi Bayi Pada Klinik Harapan Kita Menggunakan PHP”. Permasalahan dari penelitian ini yaitu pada pendataan bayi yang baru dilahirkan pada klinik tersebut dan juga jadwal yang akan mengikuti imunisasi masih bersifat manual dan belum optimal dalam penyampaiannya dimana klinik tersebut tidak memiliki catatan jadwal imunisasi yang tetap dan hanya orangtua dari bayi yang diberikan catatan jadwal untuk imunisasi. Sistem ini dirancang dengan bahasa pemrograman PHP dan Database MySQL [7].

Penelitian selanjutnya dilakukan oleh Ahmad Chusyairi, Tri Kurnia Hadi Mukting Nur, dan David Haryanto pada tahun 2018 dengan judul “Sistem Informasi Jadwal Imunisasi Kesehatan Bayi Dan Balita Berbasis SMS *Gateway*”. Unit Pelayanan Terpadu Daerah (UPTD) Puskesmas pada Kabupaten Banyuwangi belum mempunyai sistem informasi yang digunakan untuk mengingatkan jadwal imunisasi pada kader posyandu dan juga orang tua bayi serta balita karena sistem yang digunakan masih bersifat manual, sehingga menyulitkan petugas UPTD. Dari permasalahan tersebut peneliti membangun sistem informasi dengan berbasis SMS *Gateway* yang dapat mempermudah dalam penyampaian informasi jadwal imunisasi kepada kader posyandu dan juga orang tua bayi serta balita. Sistem ini juga menghasilkan laporan jadwal sms yang terkirim dan laporan ringkasan posyandu. Pengembangan sistem yang dibangun menggunakan metode *prototyping* dan untuk proses pengumpulan datanya dilakukan dengan metode wawancara, observasi dan studi pustaka [8].

Penelitian lainnya dilakukan oleh Nur Amanah, Ahmad Khambali, dan Edy Subowo pada tahun 2020 dengan judul “Sistem Informasi Pengingat Jadwal Imunisasi Pada Puskesmas Karanganyar Berbasis Web Dan Android”. Penyampaian informasi pelayanan kesehatan pada Puskesmas Karanganyar masih menggunakan buku jadwal imunisasi yang diberikan pada saat pendaftaran program

imunisasi sehingga mengharuskan orang tua bayi untuk menghafalkan jadwal imunisasi sendiri. Selain itu, buku imunisasi juga rentan hilang yang mengakibatkan orang tua tidak hadir dalam kegiatan imunisasi anaknya sesuai jadwal yang sudah ditentukan. Sehingga membutuhkan sebuah sistem informasi pengingat jadwal imunisasi untuk membantu mengingatkan para orang tua dalam proses penyampaian jadwal imunisasi. Sistem ini menggunakan framework React Native untuk membangun aplikasi Android, sedangkan untuk membangun aplikasi websitenya peneliti menggunakan framework Laravel [9].

Setelah melakukan evaluasi dari penelitian sebelumnya, penulis akan membuat sistem yang berjudul “Sistem Informasi Penjadwalan Imunisasi Berbasis *Website* Dengan *SMS Gateway*”. Sistem yang akan penulis rancang hanya digunakan untuk kegiatan imunisasi balita. Sistem informasi yang akan dibuat digunakan untuk memberikan kemudahan dalam melakukan penjadwalan imunisasi, memberikan reminder kepada pasien, memudahkan petugas dalam melakukan rekap data imunisasi, serta untuk melihat informasi daftar pasien yang kurang sadar dan sadar akan imunisasi sehingga dapat menjadi evaluasi petugas imunisasi untuk melakukan sosialisasi imunisasi. Sistem ini memiliki 3 level user yaitu petugas, pasien dan kepala puskesmas. Pengembangan sistem informasi ini menggunakan framework CodeIgniter (CI).

Tabel 2. 1 Perbandingan Penelitian

Penulis, Tahun dan Judul	Tujuan	Metode	Hasil yang Dicapai	Proses
Wisda Liani pada tahun 2020 dengan judul “Sistem Informasi Pengingat Jadwal Imunisasi Anak Dan	Untuk penyampaian jadwal terkait dengan proses pelaksanaan imunisasi anak dan ibu hamil pada Posyandu Desa Sungai Paku	Metode <i>protoype</i>	Aplikasi untuk imunisasi anak dan ibu hamil dengan pengingat waktu menggunakan SMS <i>Gateway</i>	Sistem digunakan untuk melakukan proses input data imunisasi anak dan ibu hamil

<p>Ibu Hamil Pada Posyandu Desa Sungai Paku Berbasis Web Menggunakan SMS Gateway”</p>				
<p>Ali Akbar Mushavi, Dewi Primasari, Jejen Jaenudin tahun 2019 dengan judul “Sistem Informasi Penjadwalan Kegiatan Posyandu Berbasis Web Dan <i>Whatsapp Gateway</i>”</p>	<p>Untuk membantu bidan pada desa dalam pemberitahuan jadwal posyandu dan membantu orangtua untuk mengetahui jadwal terbaru kegiatan posyandu yang akan dilaksanakan</p>	<p>Metode <i>waterfall</i></p>	<p>Aplikasi untuk pemberitahuan jadwal posyandu menggunakan <i>Whatsapp Gateway</i></p>	<p>Sistem digunakan untuk melakukan penjadwalan kegiatan posyandu dan penyampaian informasi kepada orang tua.</p>

Pinarti Puspitarini dan Fendi Hidayat, S.T., M.Kom pada tahun 2021 dengan judul “Sistem Informasi Pendataan Dan Penjadwalan Imunisasi Bayi Pada Klinik Harapan Kita Menggunakan PHP”	Untuk pendataan bayi	Metode <i>waterfall</i>	Aplikasi pendataan bayi dan pemberitahuan jadwal untuk imunisasi	Sistem dilakukan untuk memproses pendataan dan penjadwalan imunisasi
--	----------------------	-------------------------	--	--

<p>Ahmad Chusyairi, Tri Kurnia Hadi Muktining Nur, dan David Haryanto pada tahun 2018 dengan judul “Sistem Informasi Jadwal Imunisasi Kesehatan Bayi Dan Balita Berbasis SMS Gateway”</p>	<p>Untuk mengingatkan jadwal imunisasi pada kader Posyandu dan juga orang tua bayi serta balita</p>	<p>Metode <i>prototype</i></p>	<p>Aplikasi untuk menyampaikan informasi jadwal imunisasi kepada kader Posyandu dan juga orang tua bayi serta balita. Sistem ini juga menghasilkan laporan jadwal sms terkirim dan laporan ringkasan posyandu</p>	<p>Sistem digunakan untuk melakukan proses input data imunisasi bayi dan balita</p>
<p>Nur Amanah, Ahmad Khambali, dan Edy Subowo pada tahun 2020 dengan judul “Sistem Informasi Pengingat</p>	<p>Untuk penyampaian informasi pelayanan kesehatan pada Puskesmas Karanganyar</p>	<p>Metode <i>waterfall</i></p>	<p>Sistem informasi pengingat jadwal imunisasi untuk membantu mengingatkan para orang tua dalam penyampaian jadwal</p>	<p>Sistem digunakan untuk melakukan proses penjadwalan imunisasi</p>

Jadwal Imunisasi Pada Puskesmas Karanganyar Berbasis Web Dan Android”			imunisasi	
Mega Nur Indah pada tahun 2022 dengan judul “Sistem Informasi Penjadwalan Imunisasi Berbasis Website Dengan SMS Gateway”	Untuk penyampaian informasi penjadwalan imunisasi dan pendataan balita pada Puskesmas Cilacap Tengah I	Metode <i>prototype</i>	Sistem informasi pengingat jadwal imunisasi untuk membantu mengingatkan para orang tua dalam penyampaian jadwal imunisasi	Sistem yang dibuat digunakan untuk melakukan proses input data imunisasi, penjadwalan imunisasi serta perekapan data imunisasi

2.2 Landasan Teori

2.2.1 Imunisasi

Imunisasi merupakan salah satu cara yang dilakukan sebagai upaya untuk membentuk kekebalan tubuh seseorang terhadap paparan penyakit serta meningkatkan kesehatan secara menyeluruh dan berkesinambungan [10]. Imunisasi diberikan melalui vaksin yang dimasukkan ke dalam tubuh pasien melalui suntikan ataupun berbentuk cairan yang diminum (oral). Vaksin yang diberikan kepada pasien terbuat dari virus atau bakteri yang sudah dilemahkan, dibunuh, atau yang sudah dilakukan modifikasi terhadap bagian-bagian dari bakteri atau virus^[11]. Jika tubuh kita kemasukan virus atau bakteri maka tubuh akan melakukan pembentukan antibodi. Imunisasi dasar yang dilakukan pada bayi atau balita adalah salah satu cara pencegahan dan penanganan dari penyakit menular. Akan tetapi, kurangnya pengetahuan ibu akan

pentingnya imunisasi menjadikan alasan yang melatarbelakangi masih banyaknya bayi/balita yang belum mendapatkan imunisasi dasar lengkap.

2.2.2 Sistem Informasi

Sistem informasi merupakan suatu sistem yang ada di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian untuk mendukung fungsi operasi dari organisasi yang bersifat manajerial dengan kegiatan strategi suatu organisasi agar dapat menyediakan laporan dan informasi yang diperlukan oleh pihak tertentu untuk pengambilan keputusan [9].

Sistem informasi mempunyai beberapa karakteristik yang dapat menunjukkan sifat dari informasi itu sendiri [12]. Karakteristik-karakteristik tersebut antara lain yaitu :

1. Benar atau salah, yaitu karakteristik yang memiliki hubungan atau keterkaitan dengan sesuatu yang realitas.
2. Baru, dapat berarti informasi sama sekali baru bagi penerimanya.
3. Tambahan, yaitu informasi dapat memperbarui atau memberikan nilai tambah pada informasi yang sudah ada.
4. Korektif, yaitu informasi digunakan menjadi bahan korektif bagi informasi sebelumnya.
5. Penegas, yaitu informasi dapat mempertegas informasi yang sudah ada agar dapat meningkatkan persepsi penerima atas kebenaran tersebut.

Kualitas informasi (*quality of information*) sangat mempengaruhi atau ditentukan oleh 3 (tiga) hal pokok yaitu *relevancy*, *accuracy* dan *timeliness*^[12].

1. Relevansi (*relevancy*), yaitu informasi dapat dikatakan berkualitas jika relevan bagi penggunaannya. Suatu informasi dikatakan bernilai apabila manfaat yang didapatkan lebih efektif dibandingkan dengan biaya mendapatkannya.
2. Akurasi (*accuracy*), yaitu informasi dapat dikatakan akurat jika informasi yang disajikan tidak bias atau tidak menyesatkan.
3. Tepat waktu (*timeliness*), yaitu informasi yang diperoleh dari proses pengolahan data datangnya tidak terlambat. Informasi yang datangnya terlambat tidak akan mendapatkan nilai yang baik,

sehingga jika digunakan sebagai dasar dalam proses pengambilan keputusan dapat menimbulkan resiko terhadap kesalahan dalam tindakan yang akan diambil.

2.2.3 Rekayasa Web

Rekayasa web merupakan aplikasi yang menggunakan pendekatan sistematis disiplin, terstruktur, serta operasi dan pemeliharaan aplikasi berbasis web (*Web Based applications*). Kebanyakan literature tentang rekayasa web mengacu terhadap situs web, aplikasi berbasis web, dan sistem berbasis web. Tujuan utama dari rekayasa web adalah kesuksesan di dalam mengatur kompleksitas dan keanekaragaman pengembangan aplikasi berbasis web [13].

Rekayasa web merupakan subdisiplin dari rekayasa perangkat lunak yang dapat membantu menyediakan metodologi untuk merancang, mengembangkan, memelihara serta melibatkan aplikasi berbasis web. Adanya subdisiplin rekayasa web ini dapat menunjukkan suatu kebutuhan yang dipusatkan pada kesuksesan dari pengembangan aplikasi dan sistem berbasis web.

A. Metode *Prototype*

Model *prototype* terdiri dari beberapa tahapan proses pengembangan, tahapan inilah yang akan digunakan untuk menentukan keberhasilan dari sebuah software yang dibuat. Tahapan-tahapan dalam model *prototype* adalah sebagai berikut^[4] :

1. Pengumpulan Kebutuhan
Tahap pertama yang dilakukan adalah pengumpulan kebutuhan, pada tahap ini pelanggan dan pengembang sistem bersama-sama untuk mendefinisikan format dan kebutuhan keseluruhan dari perangkat lunak.
2. Membangun *Prototyping*
Pada tahapan ini pelanggan dan juga pengembang sistem akan bersama-sama melakukan pembuatan format untuk input maupun output yang nantinya akan dihasilkan oleh sistem yang dibuat.
3. Evaluasi *Prototyping*
Tahapan selanjutnya yaitu evaluasi *prototyping*, tahap ini dilakukan untuk mengetahui apakah *prototyping* yang diberikan oleh pengembang sistem sudah sesuai dengan yang diharapkan

- oleh pelanggan atau belum.
4. Pengkodean Sistem
Pada tahapan ini *prototyping* yang telah disepakati oleh pelanggan kemudian akan diubah ke dalam bahasa pemrograman yang sesuai.
 5. Menguji Sistem
Dalam tahapan pengujian sistem, coding yang telah dibuat sebelumnya oleh pengembang sistem akan dilakukan uji coba apakah sudah dapat berjalan dengan baik dan sesuai dengan permintaan pelanggan atau belum.
 6. Evaluasi Sistem
Tahapan ini dilakukan untuk melakukan evaluasi sistem atau perangkat lunak yang sudah jadi apakah sudah sesuai dengan keinginan dari pelanggan atau belum. Jika belum, maka pengembang sistem akan melakukan revisi kembali.
 7. Menggunakan Sistem
Tahap ini merupakan tahapan akhir dari proses pembuatan sistem dengan metode *prototyping* model. Dalam tahapan ini perangkat lunak yang sudah jadi dan sudah lulus uji coba, siap untuk digunakan oleh pelanggan/*user*.

Adapun keunggulan dan kelemahan metode *prototype* yaitu :

- a. Keunggulan
 - 1) Komunikasi yang dilakukan antara pengembang sistem dan juga pelanggan/*user* akan terjalin baik.
 - 2) Pengembang sistem akan dapat bekerja lebih baik dan efisien dalam menentukan kebutuhan dari setiap pelanggannya.
 - 3) Pelanggan dapat berperan aktif di dalam proses pengembangan sistem yang sedang dibuat.
 - 4) Dapat lebih menghemat waktu proses pengembangan sistem.
 - 5) Dalam penerapannya dapat menjadi lebih mudah dan efektif karena pemakai/*user* telah mengetahui apa yang diharapkan.
- b. Kelemahan
 - 1) Pelanggan/*user* terkadang tidak melihat dan menyadari bahwa perangkat lunak yang ada belum mencantumkan kualitas perangkat lunak secara keseluruhan dan juga belum memikirkan kemampuan pemeliharaan untuk jangka waktu yang lama.

- 2) Pengembang sistem menggunakan algoritma dan bahasa pemrograman yang sederhana untuk membangun *prototyping* karena ingin cepat menyelesaikan proyek yang dibuat tanpa memikirkan lebih lanjut bahwa program tersebut hanya sebuah kerangka kerja (*blueprint*) dari sistem yang dibuat.
- 3) Hubungan antara pelanggan dan komputer yang disediakan mungkin tidak mencerminkan teknik perancangan yang baik dan benar.

B. Black-Box Testing (Pengujian Kotak Hitam)

Black-Box Testing merupakan teknik pengujian perangkat lunak yang berfokus pada segi spesifikasi fungsional yang ada pada perangkat lunak yang dikembangkan tanpa menguji desain dan juga kode program. Pengujian ini dimaksudkan untuk mengetahui apakah fungsi-fungsi, masukan, dan keluaran dari perangkat lunak yang dibuat sudah sesuai dengan spesifikasi yang dibutuhkan.

Pengujian kotak hitam (*black-box testing*) dilakukan dengan cara membuat kasus uji untuk mencoba semua fungsi perangkat lunak apakah sudah sesuai dengan spesifikasi yang dibutuhkan atau belum. Kasus uji yang dibuat untuk melakukan pengujian kotak hitam atau *black-box testing* harus dibuat berdasarkan kasus benar dan juga kasus salah [14].

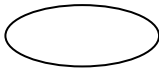




C. UML (Unified Model Language)

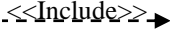
UML adalah bahasa yang digunakan untuk menspesifikasi, memvisualisasi, membangun dan mendokumentasikan *artifacts* atau bagian dari informasi yang digunakan atau dihasilkan dalam proses pembuatan perangkat lunak seperti pada permodelan bisnis dan sistem non perangkat lunak lainnya. Dalam perkembangan teknik pemrograman berorientasi objek, terdapat sebuah standarisasi bahasa pemodelan untuk pembangunan perangkat lunak yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek, yaitu *Unified Modeling Language* (UML). UML merupakan bahasa visual yang digunakan untuk pemodelan dan komunikasi sebuah sistem dengan menggunakan diagram dan teks pendukung lainnya [14].

1. Use Case Diagram

Use case diagram merupakan pemodelan yang digunakan untuk kelakuan (*behavior*) dari sistem informasi yang akan dibuat. *Use case* digunakan untuk mendeskripsikan sebuah interaksi yang terjadi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat [14].

Tabel 2. 2 Simbol dan Fungsi *Use Case*


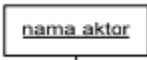

No	Simbol	Fungsi
1.	<i>Use Case</i> 	Fungsionalitas yang disediakan oleh sistem sebagai unit-unit yang digunakan untuk saling bertukar pesan antar unit atau actor, biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i>
2.	Aktor / <i>actor</i> 	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i>
3.	Asosiasi / <i>association</i> 	Komunikasi antara <i>actor</i> dan <i>use case</i> yang saling berhubungan atau <i>use case</i> yang memiliki interaksi dengan aktor
4.	Ekstensi / <i>extend</i> 	Menunjukkan bahwa suatu bagian dari elemen dapat disisipkan ke dalam elemen yang lainnya
5.	Generalisasi / <i>generalization</i> 	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>)

6.	<i>Include</i> 	Menunjukkan bahwa suatu bagian dari elemen dapat memicu eksekusi bagian dari elemen lain (yang ada pada tanda panah)
----	---	--

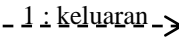
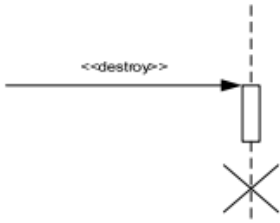
2. *Sequence Diagram*

Sequence diagram dapat menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan juga diterima oleh antar objek. Dalam menggambarkan diagram *sequence* maka harus diketahui objek-objek yang terlibat di dalam sebuah *use case* dan metode-metode yang dimiliki oleh kelas yang diinstansiasi menjadi objek tersebut [14].

Tabel 2. 3 Simbol dan Fungsi *Sequence Diagram*

No	Simbol	Fungsi
1.	Aktor  Atau  Tanpa waktu aktif	Aktor menyatakan orang, proses, atau sistem lain yang berkaitan dengan sistem informasi yang akan dibuat di luar sistem informasi tersebut
2.	Garis hidup / <i>lifeline</i> 	<i>Lifeline</i> menyatakan keberadaan dari suatu objek dalam basis waktu. Notasi yang digunakan untuk <i>lifeline</i> adalah garis lurus vertikal putus-putus yang ditarik dari sebuah objek


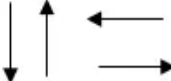
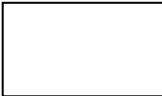
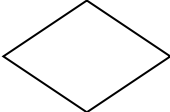


3.	<p>Objek</p> <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 10px auto;"> <u>nama objek : nama kelas</u> </div>	Menyatakan sebuah objek yang saling berinteraksi
4.	<p>Waktu aktif</p> <div style="border: 1px solid black; width: 20px; height: 40px; margin: 10px auto;"></div>	Menyatakan objek dalam keadaan aktif dan saling terhubung yang merupakan sebuah tahapan yang dilakukan di dalamnya
5.	<p>Pesan tipe <i>create</i></p> <div style="text-align: center; margin: 10px auto;"> <code>..<<create>>..</code> → </div>	Pesan tipe <i>create</i> , menyatakan suatu objek membuat objek yang lain dengan arah panah yang mengarah pada objek yang dibuat
6.	<p>Pesan tipe <i>call</i></p> <div style="text-align: center; margin: 10px auto;"> <u>1 : nama_metode()</u> → </div>	Pesan tipe <i>call</i> , menyatakan suatu objek yang memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri
7.	<p>Pesan tipe <i>send</i></p> <div style="text-align: center; margin: 10px auto;"> <u>1 : masukkan</u> → </div>	Pesan tipe <i>send</i> , menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya dengan arah panah yang mengarah pada objek yang dikirim

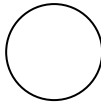
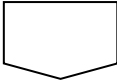
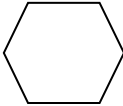




8.	Pesan tipe <i>return</i> 	Pesan tipe <i>return</i> , menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode akan menghasilkan suatu kembalian ke objek tertentu dengan arah panah yang mengarah pada objek yang menerima kembalian
9.	Pesan tipe <i>destroy</i> 	Pesan tipe <i>destroy</i> , menyatakan suatu objek yang mengakhiri hidup objek lain dengan arah panah yang mengarah pada objek yang diakhiri, <i>destroy</i> sebaiknya digunakan jika ada <i>create</i>


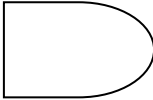
3. *Flowchart*

Flowchart atau bagan alir merupakan suatu penggambaran secara bentuk grafik atau diagram dari algoritma dimana simbol-simbol standart yang mewakili tampilan operasi perlu memperlihatkan urutan pelaksanaannya. Bagan alir ini berguna bagi seorang analis sistem untuk menggambarkan proses di dalam suatu prosedur [9].

Tabel 2. 4 Simbol dan Fungsi *Flowchart*

No	Simbol	Fungsi
1.	<i>Terminal point</i> 	Simbol yang digunakan untuk menyatakan awal atau akhir suatu program
2.	<i>Flow direction</i> 	Simbol yang digunakan untuk menghubungkan antar simbol, berfungsi juga untuk menunjukkan alur dari sebuah proses
3.	Proses 	Menunjukkan sebuah kegiatan yang dilakukan oleh komputer atau sistem
4.	<i>Decision</i> 	Simbol yang menunjukkan suatu kondisi tertentu yang menghasilkan dua kemungkinan jawaban yaitu ya atau tidak
5.	<i>Input atau output</i> 	Menunjukkan sebuah proses <i>input</i> atau <i>output</i> yang ada dan tidak bergantung pada jenis peralatan yang digunakan
6.	<i>Predefined process</i> 	Simbol yang digunakan untuk menyatakan kumpulan langkah proses ditulis sebagai prosedur

7.	<i>Connector (On-page)</i> 	Simbol yang digunakan untuk menghubungkan suatu simbol pada satu halaman yang letaknya saling berjauhan
8.	<i>Connector (Off-page)</i> 	Digunakan untuk menghubungkan simbol yang berada pada halaman yang berbeda
9.	<i>Preparation</i> 	Simbol yang menunjukkan persiapan penyimpanan ke dalam <i>storage</i>
10.	<i>Manual input</i> 	Simbol yang menunjukkan proses <i>input</i> yang dilakukan secara <i>manual</i> dengan menggunakan <i>online keyboard</i>
11.	<i>Manual operation</i> 	Simbol yang menunjukkan proses atau kegiatan yang dilakukan tanpa menggunakan komputer
12.	<i>Document</i> 	Menunjukkan proses <i>input</i> berupa dokumen dalam bentuk kertas ataupun <i>output</i> yang harus dicetak
13.	<i>Multiple document</i> 	Sama seperti simbol <i>document</i> hanya saja dokumen yang digunakan lebih dari satu

14.	<i>Display</i> 	Simbol yang menunjukkan adanya penggunaan peralatan <i>output</i>
15.	<i>Delay</i> 	Menunjukkan bahwa adanya proses <i>delay</i>

2.2.4 Pemrograman Berorientasi Objek

Metodologi berorientasi objek merupakan suatu strategi yang digunakan dalam pembangunan perangkat lunak sebagai kumpulan objek-objek yang berisi data dan juga operasi yang diberikan terhadapnya. Metodologi berorientasi objek merupakan suatu cara yang digunakan untuk mengetahui bagaimana sistem perangkat lunak yang dibangun melalui pendekatan objek secara sistematis. Metode berorientasi objek ini didasarkan pada penerapan prinsip-prinsip pengelolaan kompleksitas. Metode berorientasi objek antara lain yaitu rangkaian aktivitas analisis berorientasi objek, perancangan berorientasi objek, pemrograman berorientasi objek, dan pengujian berorientasi objek [14].

Keuntungan dalam penggunaan metodologi berorientasi objek adalah sebagai berikut :

1. Meningkatkan produktivitas
Kelas dan objek yang ditemukan di dalam suatu masalah masih dapat digunakan atau diulang kembali untuk menyelesaikan masalah lainnya yang melibatkan objek tersebut (*reusable*).
2. Kecepatan pengembangan
Sistem yang dibangun atau dirancang dengan baik dan benar pada saat analisis dan perancangan akan menyebabkan berkurangnya kesalahan pada saat pengkodean.

3. Kemudahan pemeliharaan
Model objek, pola-pola yang cenderung tetap dan stabil dapat dipisahkan dari pola-pola yang mungkin akan sering berubah-ubah.
4. Adanya konsistensi
Sifat pewarisan dan penggunaan notasi yang sama pada saat analisis, perancangan maupun pengkodean.
5. Meningkatkan kualitas perangkat lunak
Pendekatan pengembangan perangkat lunak lebih dekat dengan dunia nyata dan adanya konsistensi pada saat pengembangannya, maka perangkat lunak yang dihasilkan akan mampu memenuhi kebutuhan pemakai serta peluang untuk kesalahannya sangat kecil.

2.2.5 Basis Data

Basis data (*database*) merupakan suatu kumpulan informasi yang diatur agar mudah untuk dicari. Atau dalam penjelasan lain basis data yaitu sistem yang telah terkomputerisasi dengan tujuan utamanya adalah untuk memelihara data yang sudah diolah dan membuat informasi tersebut tersedia pada saat dibutuhkan. Pada intinya basis data adalah media yang digunakan untuk menyimpan data agar dapat diakses dengan mudah dan juga cepat [14].

A. DBMS (*Database Management System*)

DBMS (*Database Management System*) atau yang dalam bahasa Indonesia sering disebut sebagai Sistem Manajemen Basis Data merupakan sistem aplikasi yang digunakan untuk menyimpan, mengelola, dan juga menampilkan data [14]. Suatu sistem aplikasi dapat disebut DBMS apabila memenuhi persyaratan minimal sebagai berikut :

1. Dapat menyediakan fasilitas yang digunakan untuk mengelola akses data
2. Dapat menangani integritas data
3. Dapat menangani akses data
4. Dapat menangani backup data

Adapun bahasa yang digunakan dalam DBMS diantaranya adalah sebagai berikut :

1. *Data Definition Language (DDL)*

Hasil dari perintah DDL adalah suatu set dari tabel yang disimpan pada file khusus yang disebut data *dictionary/directory*.

2. *Data Manipulation Language (DML)*

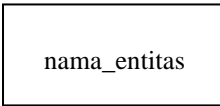
DML merupakan bahasa yang memperbolehkan pemakai atau pengguna untuk mengakses atau memanipulasi data seperti yang telah diorganisasikan sebelumnya dalam model data yang tepat.

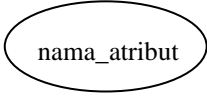
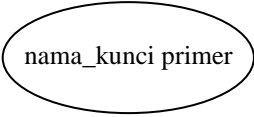
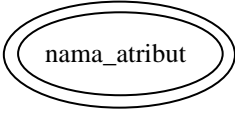
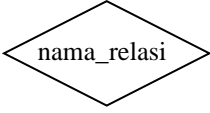
Karena pentingnya data terhadap suatu organisasi ataupun perusahaan, maka sebagian besar perusahaan memanfaatkan DBMS ini dalam proses pengelolaan data yang mereka miliki. Pengelolaan DBMS sendiri biasanya ditangani oleh tenaga ahli atau spesialis yang khusus untuk menangani masalah DBMS atau yang biasa disebut sebagai DBA (*Database Administrator*).


B. *Entity Relationship Diagram (ERD)*

ERD merupakan suatu cara yang digunakan untuk pemodelan basis data relasional. Sehingga apabila penyimpanan basis data menggunakan OODBMS (*Object Oriented Database Management System*) maka dalam perancangan basis data tidak perlu menggunakan ERD. ERD sendiri memiliki beberapa aliran notasi seperti notasi Chen (dikembangkan oleh Peter Chen), Barker (dikembangkan oleh Richard Barker, Ian Palmer, Harry Ellis), notasi Crow's Foot, dan beberapa notasi yang lainnya. Namun, notasi yang banyak digunakan adalah notasi dari Chen. Berikut ini adalah simbol-simbol yang digunakan pada ERD dengan notasi Chen [14].

Tabel 2. 5 Simbol dan Fungsi ERD

No	Simbol	Fungsi
1.	Entitas / <i>entity</i> 	Entitas merupakan objek yang mewakili sesuatu yang nyata dan dibedakan dari sesuatu yang lain. Objek dasarnya dapat berupa orang, benda atau suatu hal yang keterangannya perlu disimpan di dalam basis data

2.	<p>Atribut</p> 	<p>Atribut merupakan <i>field</i> atau kolom data yang butuh untuk disimpan pada suatu entitas</p>
3.	<p>Atribut kunci primer</p> 	<p><i>Field</i> dan kolom data yang harus disimpan di dalam entitas dan digunakan sebagai kunci akses record yang diinginkan, contohnya berupa id yang bersifat unik (berbeda tanpa ada yang sama)</p>
4.	<p>Atribut multivalai / <i>multivalve</i></p> 	<p><i>Field</i> atau kolom data yang harus disimpan pada satu entitas yang dapat memiliki nilai lebih dari satu</p>
5.	<p>Relasi</p> 	<p>Relasi digunakan untuk menghubungkan antar entitas</p>

6.	<p data-bbox="274 199 509 231">Asosiasi / <i>association</i></p>  <p>The diagram shows a horizontal line representing an association. Above the line, towards the right end, is the letter 'N'. At the right end of the line, it forks into two separate lines, indicating a multiplicity of two at that end.</p>	<p data-bbox="655 199 946 558">Asosiasi digunakan untuk menghubungkan antara relasi dan entitas yang dimana kedua ujungnya memiliki <i>multiplicity</i> kemungkinan dari jumlah pemakaian dan kemungkinan dari jumlah maksimum keterhubungan antara entitas satu dengan entitas yang lainnya yang disebut dengan kardinalitas</p>
----	--	---