

BAB II
TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Penelitian terdahulu ini menjadi salah satu penulis dalam melakukan penelitian sehingga dapat memperkaya teori yang digunakan dalam mengkaji penelitian yang dilakukan. Dari penelitian terdahulu, penulis tidak menemukan penelitian dengan judul yang sama seperti judul penelitian penulis. Namun penulis mengangkat beberapa penelitian sebagai referensi dalam memperkaya bahan kajian pada penelitian penulis. Berikut merupakan penelitian terdahulu berupa beberapa jurnal terkait dengan penelitian yang dilakukan penulis.

Tabel 2.1 Penelitian Tedahulu

No	Tahun	Judul	Peneliti	Hasil penelitian	Perbandingan
1	2015	Rancang Bangun Sistem Informasi Desa untuk Meningkatkan Pelayanan Terhadap Masyarakat	Maskur	Sistem Informasi Desa untuk Pelayanan terhadap masyarakat dibangun untuk meningkatkan kemudahan	Objek penelitian, Aplikasi berbasis web, laporan out put

				masyarakat bagi yang membutuhkan surat – surat pengantar dari desa	
2	2016	Rancang Bangun sistem informasi dan pengolahan data kelahiran, kematian, datang dan pindah dikantor kelurahan sekaran kecamatan gunungpati kota semarang	Rini Kartiko Sari, Riana Defi Mahadji Putri	sistem informasi dan pengolahan data kelahiran, kematian, datang dan pindah untuk mengidentifikasi tujuan dan syarat informasi yang dibutuhkan	Objek penelitian, Aplikasi berbasis web, laporan out put, struktur sistem
3	2016	Upaya Peningkatan Pelayanan Administrasi Kependudukan Menggunakan Teknologi Informasi: Rancang Bangun	Moh. Anshori Aris Widya, Yosi Agustia wan,	Sistem Informasi Pelayanan Administrasi Kependudukan untuk mempermudah pengolahan data	Objek penelitian, Aplikasi berbasis grafik, laporan out put

		Sistem Informasi diDesa Sumbermulyo Kecamatan Jogoroto Kabupaten Jombang	Ivan Dwi Fibrian, dan Zainal Muttaqin	kependudukan	
4	2017	Sistem Informasi Pelayanan Administrasi Kependudukan dan Catatan Sipil Kelurahan Dikecamatan Marangkayu Kutai Kartanegara	Masna Wati, Engla Despahar i	Sistem Informasi Pelayanan Administrasi Kependudukan dan Catatan Sipil mengelola data warga sehingga setiap pengelolaan selalu terhubung pada database	Objek penelitian, Aplikasi berbasis web, laporan out put

2.2 Landasan Teori

2.2.1 Sistem

Menurut Maniah dan Dini Hamidini (2017), mengatakan bahwa “Sistem adalah kumpulan dari elemen-elemen berupa data, jaringan kerja dari prosedur-prosedur yang

saling berhubungan, sumber daya manusia, teknologi baik hardware maupun software yang saling berinteraksi sebagai satu kesatuan untuk mencapai tujuan/sasaran tertentu yang sama”.

2.2.2 Informasi

Menurut Muhamad Muslihudin dan Oktafianto (2016), mengatakan bahwa “Informasi merupakan data yang diolah menjadi bentuk yang berguna untuk membuat keputusan”.

2.2.3 Sistem Informasi

Sistem informasi sendiri telah dikemukakan oleh beberapa penulis sebagai berikut: Sistem informasi adalah suatu kumpulan dari komponen-komponen yang saling berinteraksi untuk mengelola informasi pada suatu organisasi untuk mendukung kegiatan bisnis organisasi.

Menurut Yakub Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, terkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk tujuan tertentu. Selanjutnya menurut Sutabri menyatakan “Sistem adalah suatu kumpulan atau himpunan dari unsur, komponen, atau variable yang terorganisasi, saling berinteraksi, saling tergantung satu sama lain dan terpadu”.

2.2.4 Sistem Informasi Pelayanan kependudukan Administrasi Kependudukan

Kependudukan dan peristiwa penting memerlukan bukti yang sah untuk dilakukan pengadministrasian dan pencatatan sesuai dengan ketentuan undang-undang. Pada

pemenuhan hak penduduk, terutama di bidang Pencatatan Sipil, masih ditemukan penggolongan penduduk yang berdasarkan pada perlakuan diskriminatif yang membeda-bedakan suku, keturunan, dan agama sebagaimana diatur dalam berbagai peraturan produk colonial Belanda. Penggolongan penduduk dan pelayanan diskriminatif yang demikian itu tidak sesuai dengan Pancasila dan Undang – Undang Dasar Negara Republik Indonesia Tahun 1945. Kondisi tersebut mengakibatkan pengadministrasian kependudukan mengalami kendala yang mendasar sebab sumber data kependudukan belum terkoordinasi dan terintegrasi, serta terbatasnya cakupan pelaporan yang belum terwujud dalam suatu sistem administrasi.

kependudukan yang utuh dan optimal. Kondisi social dan administratif tersebut tidak memiliki system database kependudukan yang menunjang pelayanan administrasi kependudukan (Burhanudin 2008 : 1).

Pada buku pintar kependudukan (Sudjarwo,2004:75), pelayanan administrasi kependudukan diartikan sebagai pelayanan di bidang kependudukan yang diberikan oleh aparat pemerintah dan nonpemerintah dari tingkat pusat sampai tingkat desa atau kelurahan, RW dan RT. Pada pelayanan administrasi kependudukan, aparat pemerintah dan non pemerintah memberikan pelayanan misalnya pengurusan izin nikah, permohonan KTP, surat keterangan, dan kartu keluarga, dan surat-surat kependudukan yang lain.

2.2.5 Unified Modeling Language (UML)

Unified Modeling Language (UML) adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak. UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem.




Unified Modeling Language (UML) adalah sebuah bahasa yang berdasarkan grafik atau gambar untuk memvisualisasi, menspesifikasikan, membangun, dan pendokumentasian dari sebuah sistem pengembangan software berbasis OO (*Object-Oriented*). UML sendiri juga memberikan standar penulisan sebuah sistem blue print, yang meliputi konsep bisnis proses, penulisan kelas-kelas dalam bahasa program yang spesifik, skema database, dan komponen-komponen yang diperlukan dalam sistem software.

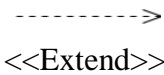

2.2.5.1 Use Case Diagram

Use case atau diagram use case merupakan pemodelan untuk kelakuan (behavior) sistem informasi yang akan dibuat. Use case mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, usecase digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu. Syarat penamaan pada *usecase* adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada use case yaitu pendefinisian apa yang disebut aktor dan *usecase*.

- a. Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.
- b. Use case merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.

Tabel 2.2 Simbol Use Case

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Use Case</i>	fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal diawal frase nama use case
2		<i>Actor</i>	orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama aktor
3		<i>Association</i>	komunikasi antara aktor dan use case yang berpartisipasi pada use case atau use

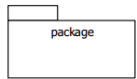
			casememiliki interaksi dengan actor
4		<i>Extend</i>	relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan dapat berdiri sendiri walau tanpa use case tambahan itu; mirip dengan prinsip inheritance pada pemrograman berorientasi objek; biasanya use case tambahan memiliki nama depan yang sama dengan use case yang ditambahkan
5		<i>Generalization</i>	Hubungan generalisasi dan spesialisasi (umum - khusus) antara dua buah use case dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya



2.2.5.2 Class Diagram

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

- a. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas
- b. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas

Tabel 2.3. Simbol Class Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Package</i>	<i>package</i> merupakan sebuah bungkusan dari satu atau lebih kelas
2		<i>Class</i>	kelas pada struktur system
3.		<i>Interface</i>	sama dengan konsep interface dalam pemrograman berorientasi objek
4		<i>Association</i>	relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan multiplicity
5		<i>directed association</i>	relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan


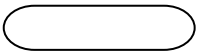
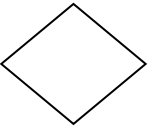


			multiplicity
6		<i>generalisasi</i>	relasi antar kelas dengan makna generalisasi-spesialisasi (umum-khusus)
7		<i>Dependency</i>	relasi antar kelas dengan makna kebergantungan antar kelas
8		<i>aggregation</i>	relasi antar kelas dengan makna semua-bagian (whole-part)

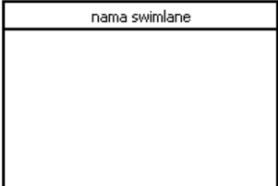
2.2.5.3 Activity Diagram

Activity diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal-hal berikut:

- a. Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan
- b. Urutan atau pengelompokan tampilan dari sistem/user interface dimana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan
- c. Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya

Tabel 2.4 Simbol Activity Diagram







No.	Simbol	Keterangan
1	<p><i>Start Stat/Status Awal</i></p> 	status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal
2	<p><i>Activity/Aktivitas</i></p> 	aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
3	<p><i>Decision/percabangan</i></p> 	asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
4	<p>penggabungan / <i>join</i></p> 	asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
5.	<p><i>End State/Status Akhir</i></p> 	status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir

6	<p style="text-align: center;"><i>Swimlane</i></p> 	<p style="text-align: center;">memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi</p>
---	--	--

2.2.5.4 Sequence Diagram

Diagram sekuen menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dengan message yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah use case beserta sekuen juga dibutuhkan untuk melihat skenario yang ada pada use case. Banyaknya diagram sekuen yang harus digambar adalah minimal sebanyak pendefinisian use case yang memiliki proses sendiri atau yang penting semua use case yang telah didefinisikan interaksinya pesan sudah dicakup dalam diagram sekuen sehingga semakin banyak use case yang didefinisikan maka diagram sekuen yang harus dibuat juga semakin banyak

Tabel 2.5 Simbol Sequence Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		<i>State</i>	Nilai atribut dan nilai link pada suatu waktu tertentu, yang dimiliki oleh suatu objek.
2		<i>InitialPseudo State</i>	Bagaimana objek dibentuk atau diawali
3		<i>FinalState</i>	Bagaimana objek dibentuk dan dihancurkan
4		<i>Transition</i>	Sebuah kejadian yang memicu sebuah state objek dengan cara memperbaharui satu atau lebih nilai atributnya
5		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
6		<i>Node</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

2.2.6 Web

Murya (2012), “Web adalah suatu layanan sajian informasi yang menggunakan konsep hyperlink, yang memudahkan surfer atau browser dalam melakukan penelusuran informasi melalui internet”.

Sibero (2013), “Web adalah suatu sistem yang berkaitan dengan dokumen digunakan sebagai media untuk menampilkan teks, gambar dan berbagai multimedia pada jaringan internet”.

Sidik dan Pohan (2010), “World Wide Web, lebih dikenal dengan web merupakan salah satu layanan yang didapat oleh pemakai komputer yang terhubung ke internet”

2.2.7 HTML (Hyper Text Markup Language)

HyperText Markup Language (HTML) adalah skrip yang mengatur struktur website.

Beberapa tugas utama HTML dalam membangun website diantaranya sebagai berikut:

- Menentukan layout website
- Memformat text dasar seperti pengaturan paragraf dan format font
- Membuat list
- Menyisipkan gambar
- Membuat link
- Membuat format

2.2.8 PHP (Hypertext Preprocessor)

PHP adalah sebuah bahasa pemrograman yang berjalan dalam sebuah web-server (serverside). PHP diciptakan oleh *programmer unix* dan *Perl* yang bernama Rasmus Lerdoft pada bulan agustus September 1994. *Script* PHP adalah bahasa program yang berjalan pada sebuah webserver, atau sering disebut *serverside*. Oleh karena itu, PHP dapat melakukan apa saja yang bisa dilakukan program CGI lain, yaitu mengolah data dengan tipe apapun, menciptakan halaman web yang dinamis, serta menerima dan menciptakan *cookies*, dan bahkan PHP bisa melakukan lebih dari itu.

2.2.9 MySQL

MySQL merupakan sebuah implementasi dari sistem manajemen basis data relasional (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (*General Public License*). Setiap pengguna dapat secara bebas menggunakan MySQL, namun dengan batasan perangkat lunak tersebut tidak boleh dijadikan produk turunan yang bersifat komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam basis data yang telah ada sebelumnya yaitu SQL (*Structured Query Language*). SQL adalah sebuah konsep pengoperasian basis data, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis. Mysql adalah sebuah server database

open source yang terkenal yang digunakan berbagai aplikasi terutama untuk server atau membuat bermacam aplikasi. Mysql berfungsi sebagai SQL (*Structured Query Language*) yang dimiliki sendiri dan sudah diperluas oleh Mysql umumnya digunakan bersamaan dengan aplikasi yang membutuhkan data untuk membuat aplikasi server yang dinamis dan *powerfull* (Harison dan Syarif, 2016).

2.2.10 XAMPP

Madcoms (2016), “Xampp adalah sebuah paket kumpulan software yang terdiri dari Apache, MySQL, PhpMyadmin, PHP, Perl, Filezilla, dan lainlain.

2.2.11 Database

Menurut Rosa dan Shalahuddin (2016) “basis data merupakan salah satu bagian dalam rekayasa perangkat lunak yang terkomputerisasi dan bertujuan utama memelihara data yang sudah diolah atau media penyimpanan informasi agar dapat diakses dengan mudah dan cepat”.

Dapat disimpulkan bahwa basis data bagian dari rekayasa perangkat lunak yang terkomputerisasi sebagai media penyimpanan informasi yang saling berhubungan atau punya relasi untuk penyimpanan data informasi agar dapat diakses dengan mudah dan cepat.

2.2.12 Metode Pengembangan Sistem Menggunakan Metode OOAD

Pada penelitian ini ada beberapa metode yang digunakan dalam pengembangan sistem. Salah satu yang terkenal adalah Menurut Mathiassen *Object-Oriented Analysis and Design* (OOAD) adalah metode untuk menganalisa dan merancang sistem dengan pendekatan berorientasi object. *Object* diartikan sebagai suatu entitas yang memiliki identitas, *state*, dan *behavior*.

Pada analisa, identitas sebuah *object* menjelaskan bagaimana seorang *user* membedakannya dari *object* lain, dan *behavior object* digambarkan melalui event yang dilakukannya. Sedangkan pada perancangan, identitas sebuah *object* digambarkan dengan cara bagaimana *object* lain mengenalinya sehingga dapat diakses, dan *behavior object* digambarkan dengan *operation* yang dapat dilakukan *object* tersebut yang dapat mempengaruhi *object* lain dalam sistem.

2.2.12.1 Karakteristik OOAD

Dalam pendekatan berorientasi objek ada 4 pilar utama yang harus dipahami dalam pendekatan berorientasi objek yaitu karakteristik. Karakteristik (ciri) suatu program termasuk OOAD/OOP, apabila terdapat abstraksi, pembungkusan (*encapsulation*), *polymorphisme*, dan turunan (*inheritance*).

a. Abstraction

Kemampuan untuk menjadikan dalam bentuk yang lebih sederhana. Hal ini juga dikenal dalam metodologi pendekatan struktur yaitu dekomposisi seperti menyerderhanakan suatu sistem dalam bentuk *Context Diagram*.

- b. *Encapsulation* Merupakan suatu karakteristik OOAD dimana program terbungkus (jadi satu) data dan perilaku, artinya lebih memperhatikan aspek internal daripada aspek eksternal. Contoh: dalam program terdapat tombol *button close* didalamnya ada *method system.exit(0)* untuk keluar dari sistem java. Berbeda dengan metodologi terdahulu, metodologi ini menggabungkan atribut dan fungsi / proses kedalam suatu objek yang disebut dengan *encapsulation*. Setiap objek dapat “menyembunyikan” kompleksitasnya dan berhubungan dengan objek lain dengan mengirim “pesan / *message*” yang dapat dikenal dan diproses oleh objek penerima.
- c. *Polymorphisme* Dengan kata lain suatu mekanisme yang memungkinkan suatu objek memiliki semua atau sebagian definisi dari objek induk. Menurut Bambang Hariyanto (2007:67) *Polymorphism* berasal dari kata *Poly* yang artinya banyak dan *morph* yang artinya bentuk. Jadi *polymorphism* adalah kemampuan suatu atribut atau *method* dapat berubah dalam berbagai bentuk dalam implementasi. Contoh Pada obyek mobil, walaupun minibus dan truk merupakan jenis obyek mobil yang sama, namun memiliki juga perbedaan. Misalnya suara truk lebih keras dari pada minibus, hal ini juga berlaku pada obyek anak (*child*) melakukan metoda yang sama dengan algoritma berbeda dari obyek induknya. Hal ini yang disebut *polymorphism*, teknik atau konsep dasar lainnya adalah ruang lingkup/pembatasan. Artinya setiap obyek mempunyai ruang lingkup kelas, atribut, dan metoda yang dibatasi.
- d. *Inheritance* Merupakan suatu karakteristik OOAD di mana suatu kelas (*parent/base class*) dapat diturunkan ke kelas lain (*child/derived class*), sehingga kelas anak

dapat memiliki data atau perilaku kelas orangtuanya. Contoh dengan beberapa buah mobil yang mempunyai kegunaan yang berbeda-beda. Ada mobil bak terbuka seperti truk, bak tertutup seperti sedan dan minibus. Walaupun demikian obyek-obyek ini memiliki kesamaan yaitu teridentifikasi sebagai obyek mobil, obyek ini dikatakan obyek induk (*parent*). Sedangkan minibus obyek anak (*child*), berarti semua operasi yang berlaku pada mobil berlaku pada minibus.