

BAB II TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Pada penulisan skripsi ini, peneliti menggali informasi dari penelitian-penelitian sebelumnya sebagai bahan perbandingan, baik mengenai metode yang digunakan dan kesimpulan yang ada. Selain itu, peneliti juga menggali dari buku-buku maupun skripsi dalam rangka mendapatkan suatu informasi yang ada sebelumnya tentang teori judul yang digunakan untuk memperoleh landasan teori ilmiah.

Tabel 2.1 Penelitian Terkait

No	Penulis	Judul	Keterangan
1.	Rahayu, Alfeno, & Wahyono, 2019. STMIK Raharja.	Rancang Bangun Aplikasi Sistem Pengelolaan Data Pembuatan Akta Tanah Pada Kantor Desa Cangkudu Tangerang.	1. Berbasis <i>website</i> . 2. Tujuan merancang aplikasi ini yaitu dapat membantu sekretaris desa dalam melakukan pengolahan data sehingga proses pendataan menjadi lebih cepat. Proses pencarian data pemohon pembuatan akta tanah yang telah lama dibuat menjadi lebih cepat. Serta penyusunan laporan menjadi lebih efisien.

No	Penulis	Judul	Keterangan
2.	Salma,Pudjiantoro, & Umbara, 2018. Universitas Jenderal Achmad Yani	Sistem <i>Monitoring</i> Pelayanan Di Badan Pertanahan Nasional Kota Cimahi.	1.Berbasis <i>website</i> . 2.Tujuan pembuatan sistem informasi ini yaitu untuk mengelola semua data mengenai <i>monitoring</i> kegiatan jenis penerimaan pelayanan di Badan Pertanahan Nasional Kota Cimahi serta memberikan keluaran berupa informasi yang dapat direpresentasikan dalam bentuk grafik untuk memudahkan setiap bagian kelola pengerjaan melakukan pencarian dampak yang sering menjadi masalah di Badan Pertanahan Nasional tersebut sehingga harus dilakukan <i>monitoring</i> sistem.
3.	Suryana,Agustina, & Fauziah, 2021. Teknik Informatika	Rancang Bangun Sistem Pengolahan Data Buku Sertifikat Tanah Berbasis Web.	1.Berbasi <i>website</i> . 2.Bertujuan untuk mengoptimalkan sistem pada kantor Badan Pertanahan Nasional Kabupaten Bandung terutama pada sistem pendataan yang terdiri dari peta bidang tanah (PBT), berkas permohonan, sertifikat tanah, dan pendaftaran permohonan pembuatan sertifikat tanah.

No	Penulis	Judul	Keterangan
			<p>3. Dalam pengujian sistem menggunakan metode <i>Black Box</i>.</p> <p>4. UML sebagai analisis dan perancangan sistem yang akan dibuat.</p>
4.	Wiyono, Nugroho, & Astri, 2019. STIKOM Dinamika Bangsa	Perancangan Aplikasi Pengarsipan Dan Pendaftaran Pada Seksi hubungan Hukum.	<p>1. Berbasis <i>website</i>.</p> <p>2. Metode penelitian yang digunakan adalah Model Sekuensial Linier.</p> <p>3. Bertujuan untuk mengoptimalkan pengarsipan dan pendaftaran tanah pada Seksi hubungan Hukum.</p> <p>4. Dalam pengujian sistem menggunakan metode <i>Black Box</i>.</p>

Tabel 2.2 Penelitian Yang Dilakukan Penulis

No	Penulis	Judul	Keterangan
1.	Eri Kurniawan (2022), Universitas Tanjungpura Pontianak	Aplikasi Layanan Pengukuran Bidang Tanah Pada Kantor Badan Pertanahan Nasional Kabupaten Sekadau Berbasis <i>Website</i> .	<p>1. Aplikasi Berbasis <i>Website</i>.</p> <p>2. Tujuan aplikasi yaitu sebagai pengelolaan data pengukuran dan pemetaan bidang tanah serta dapat <i>me-monitoring</i> petugas</p>

No	Penulis	Judul	Keterangan
			<p>lapangan dalam melakukan pengukuran dan pemetaan.</p> <p>3. Metode pengembangan sistem aplikasi yang digunakan yaitu metode waterfall.</p> <p>4. Dalam melakukan pengujian sistem, dilakukan dengan menggunakan metode <i>Black Box</i>.</p>

2.2 Aplikasi

Menurut (Pramana, 2006), aplikasi merupakan suatu unit perangkat lunak yang dibuat untuk melayani kebutuhan akan beberapa aktivitas seperti sistem perniagaan, game, pelayanan masyarakat, periklanan, atau semua proses yang hampir dilakukan manusia.

2.3 Pelayanan

Pelayanan mempunyai makna yang melayani orang yang dilayani. Jika melayani, maka sejatinya adalah memberikan pelayanan atau pengabdian secara profesional dan proporsional. Bentuk dan cara pelayanan juga merupakan bagian dari makna yang terpisahkan dari pelayanan itu sendiri.

Pelayanan adalah “Kegiatan untuk memenuhi kebutuhan seseorang atau menyediakan keperluan orang. Pelayanan juga merupakan suatu kegiatan atau urutan kegiatan yang terjadi dalam interaksi langsung antara seseorang dengan orang lain atau mesin secara fisik dan menyediakan kepuasan pelanggan. Pelayanan sebagai hal, cara atau hasil pekerjaan yang melayani” (Susepti, 2017).

Pelayanan adalah “Tindakan atau perbuatan seseorang atau suatu organisasi untuk memberikan kepuasan kepada pelanggan, sesama karyawan, dan juga pimpinan” (Kasmir, 2016). Pelayanan dan pemberian dukungan kepada

masyarakat adalah kemampuan petugas dalam melaksanakan tugasnya yaitu memberikan layanan dan dukungan dengan penuh komitmen serta kemampuan memecahkan masalah pada saat pemberian layanan itu berlangsung.

Berdasarkan penjelasan diatas, dapat penulis simpulkan bahwa pelayanan merupakan suatu kegiatan atau urutan kegiatan yang terjadi dalam interaksi langsung antara seseorang dengan orang lain atau mesin secara fisik dan menyediakan kepuasan pelanggan dalam bentuk barang atau jasa dalam rangka upaya pemenuhan kebutuhan masyarakat.

2.4 Pemetaan

Pemetaan adalah pengelompokkan suatu kumpulan wilayah yang berkaitan dengan beberapa letak geografis wilayah yang meliputi dataran tinggi, pegunungan, sumber daya dan potensi penduduk yang berpengaruh terhadap sosial kultural yang memiliki ciri khas khusus dalam penggunaan skala yang tepat (Munir, 2014).

2.5 Monitoring

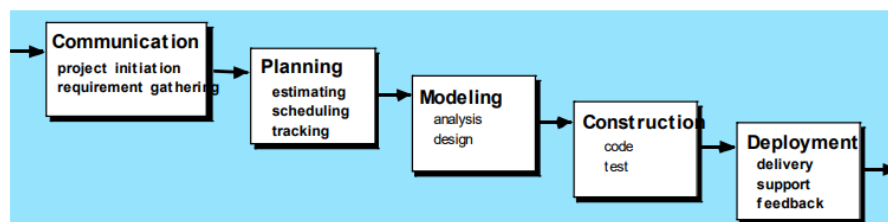
Menurut (Mudjahidin & Putra, 2010), tujuan dari *monitoring* adalah mengkaji apakah kegiatan-kegiatan yang dilaksanakan telah sesuai dengan rencana, mengidentifikasi masalah yang timbul agar langsung dapat diatasi, melakukan penilaian apakah pola kerja dan manajemen yang digunakan sudah tepat untuk mencapai tujuan, mengetahui kaitan antara kegiatan dengan tujuan untuk memperoleh ukuran kemajuan.

2.6 Bidang Tanah

Bidang tanah adalah bagian permukaan bumi yang merupakan satuan bidang terbatas. Karena bidang tanah dipermukaan bumi merupakan bagian dari ruang yang keberadaannya sangat terbatas, untuk itu pemanfaatannya harus dilakukan secara terencana dan terkendali. Pemetaan bidang tanah dilakukan dengan cara melakukan pengukuran posisi titik-titik batas dari bidang tanah untuk mendapatkan kepastian letak bidang tanah tersebut di permukaan Bumi. Pemetaan suatu bidang tanah dilaksanakan dengan cara terestrial, fotogrametris, atau metode lainnya (Yuwono, Ningsih, & Awaluddin, 2014).

2.7 Waterfall

Model *waterfall* adalah model klasik yang bersifat sistematis, berurutan dalam membangun software. Nama model ini sebenarnya adalah “*Linear Sequential Model*”. Model ini disebut juga dengan “*Classic Life Cycle*” atau metode waterfall. Model ini termasuk ke dalam model generic pada rekayasa perangkat lunak dan pertama kali diperkenalkan oleh Winson Royce sekitar tahun 1970 sehingga sering dianggap kuno, tetapi merupakan model yang paling banyak dipakai dalam *Software Engineering* (SE). model ini melakukan pendekatan secara sistematis dan berurutan. Disebut dengan waterfall karena tahap demi tahap yang dilalui harus menunggu selesainya tahap sebelumnya dan berjalan berurutan (Pressman R. , 2015). Fase-fase dalam Model *Waterfall* :



Gambar 2.1 *Waterfall*

Model air terjun (*waterfall*) biasa disebut siklus hidup perangkat lunak. Mengambil kegiatan dasar seperti spesifikasi, pengembangan, validasi, evolusi dan mempresentasikan sebagai fase-fase proses yang berada seperti spesifikasi persyaratan, perancangan perangkat lunak., implementasi, pengujian dan seterusnya.

Keuntungan menggunakan metode *waterfall* adalah prosesnya lebih terstruktur, hal ini membuat kualitas *software* baik dan tetap terjaga. Dari sisi *user* dan juga lebih menguntungkan, karena dapat merencanakan dan menyiapkan kebutuhan data dan proses yang diperlukan sejak awal. Penjadwalan juga menjadi lebih menentu, karena jadwal setiap proses dapat ditentukan secara pasti.

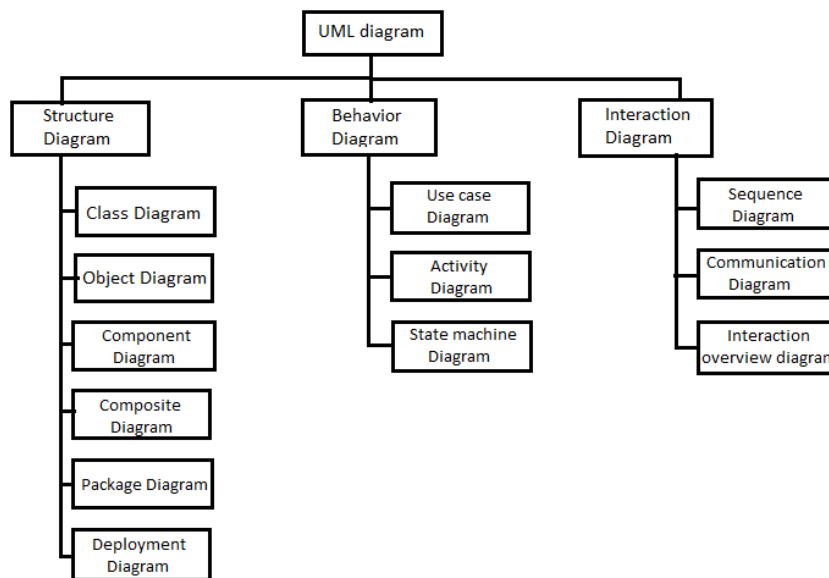
2.8 OpenStreetMaps

OpenStreetMap (OSM) adalah sebuah proyek berbasis web untuk membuat peta seluruh dunia yang gratis dan terbuka, dibangun sepenuhnya oleh sukarelawan dengan melakukan survei menggunakan GPS, mendigitasi citra satelit, dan mengumpulkan serta membebaskan data geografis yang tersedia di publik.

2.9 Unified Modeling Language (UML)

Menurut (Rosa & Shalahuddin, 2013), UML adalah salah satu standar bahasa visual yang banyak digunakan di dunia industri untuk mengidentifikasi requirement, membuat analisis & desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek.

Menurut (Sukamto, R.A., & Shalahuddin, 2013), dalam UML terdapat 13 diagram yang dikelompokkan dalam 3 kategori. Pembagian kategori dan macam-macam diagram tersebut dapat dilihat pada gambar 2.2 di bawah ini :



Gambar 2.2 Klasifikasi Diagram UML

Berikut penjelasan singkat dari Gambar 2.2 :

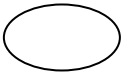
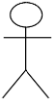

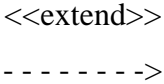
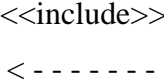

1. *Structure Diagram*, yaitu kumpulan diagram-diagram yang menggambarkan struktur statis dari sistem yang dimodelkan.
2. *Behavior Diagram*, yaitu kumpulan diagram-diagram yang menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi di dalam sistem.
3. *Interaction Diagram*, yaitu kumpulan diagram-diagram yang menggambarkan interaksi sistem dengan sistem lain ataupun interaksi antara subsistem dalam sebuah sistem.

Secara garis besar, beberapa diagram utama sudah dapat menggambarkan keseluruhan sistem. *Diagram* tersebut antara lain *use case diagram*, *class Diagram*, *sequence Diagram*, dan *activity Diagram*.

2.9.1 Use Case

Menurut (Sukamto, R.A., & Shalahuddin, 2013), diagram *use case* merupakan pemodelan untuk kelakuan sistem informasi yang akan dibuat. *Use Case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu. Berikut adalah simbol-simbol yang ada pada diagram *use case*.

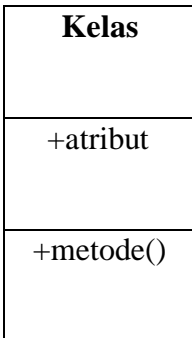

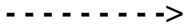
Tabel 2.3 Deskripsi Notasi Pada *Use Case Diagram*

No	Notasi	Nama	Deskripsi
1		<i>Use Case</i>	Menggambarkan fungsionalitas yang dimiliki sistem.
2		<i>Actor</i>	Menggambarkan semua objek di luar sistem (bukan hanya pengguna sistem/perangkat lunak) yang berinteraksi dengan sistem yang dikembangkan.
3		<i>Association</i>	Lintasan komunikasi antara <i>actor</i> dengan <i>use case</i> .
4		<i>Extended</i> (Ekstensi)	Penambahan perilaku ke suatu <i>use case</i> dasar.
5		<i>Include</i> (Menggunakan)	Penambahan perilaku ke suatu <i>use case</i> dasar yang secara <i>explicit</i> mendeskripsikan penambahan tersebut.
6		<i>Generalization</i> (Generalisasi)	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu merupakan fungsi yang lebih umum dari lainnya.

2.9.2 Class Diagram

Class Diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem (Sukanto, R.A., & Shalahuddin, 2013). *Class Diagram* memiliki apa yang disebut atribut dan operasi. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas. Berikut adalah simbol-simbol yang ada pada tabel 2.4.

Tabel 2.4 Simbol Class Diagram



No	Notasi	Nama	Deskripsi
1		<i>Class</i>	Menggambarkan konsep dasar pemodelan sistem.
2		Asosiasi (<i>Association</i>)	Sebuah garis solid antara dua <i>class</i> , ditarik dari <i>class</i> sumber ke <i>class</i> target lebih spesifik, digunakan dalam struktur pewarisan.
3		Ketergantungan (<i>Dependency</i>)	Relasi antara dua elemen jika perubahan definisi sebuah elemen (<i>supplier</i> atau sumber) dapat menyebabkan perubahan pada

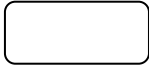
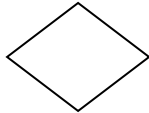

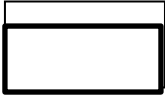
No	Notasi	Nama	Deskripsi
			elemen lainnya (<i>Client</i> atau target).

2.9.3 Activity Diagram

Activity Diagram menggambarkan *workflow* (aliran kerja) proses bisnis dan urutan aktivitas dalam sebuah proses. *Activity Diagram* sangat mirip dengan *flowchart* karena memodelkan *workflow* dari satu aktivitas ke aktivitas lainnya atau dari aktivitas ke status. Membuat *activity Diagram* pada awal pemodelan proses cukup menguntungkan untuk membantu memahami keseluruhan proses. *Activity Diagram* juga bermanfaat untuk menggambarkan *parallel behavior* atau menggambarkan interaksi antara beberapa *use case*. Berikut merupakan simbol notasi *Activity Diagram* pada Tabel 2.5 (Rosa & Shalahuddin, 2013).

Tabel 2.5 Simbol Activity Diagram

No.	Notasi	Nama	Deskripsi
1		Status Awal (<i>initial node</i>)	Status awal aktivitas, sebuah diagram aktivitas memiliki status awal.
2		Status Akhir (<i>final node</i>)	Status akhir yang dilakukan sistem.

No.	Notasi	Nama	Deskripsi
3		Aktivitas (<i>activity</i>)	Aktivitas yang dilakukan oleh sistem, biasanya diawali oleh kata kerja.
4		Percabangan (<i>decision</i>)	Asosiasi percabangan di mana jika ada pilihan aktivitas lebih dari satu.
5		Penggabungan (<i>join</i>)	Asosiasi penggabungan di mana lebih dari satu aktivitas digabungkan menjadi satu.
6		<i>Swimlane</i>	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

2.10 Teknologi Pendukung

2.10.1 Website

Menurut (Bekti, 2015), mengemukakan bahwa Website merupakan kumpulan halaman-halaman yang digunakan untuk menampilkan informasi teks, gambar diam atau gerak, animasi, suara, dan atau gabungan dari semuanya, baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait, yang masing-masing dihubungkan dengan jaringan-jaringan halaman.

2.10.2 HTML

Menurut (Wardana, 2016), *Hypertext* markup language (HTML) merupakan bahasa pemrograman dasar untuk mengelola *website*. Akan tetapi HTML hanya terbatas pada pembuatan *website* statis (*website* yang tidak dapat berinteraksi aktif dengan *user*). Maka dari itu HTML biasa dikombinasikan dengan Bahasa pemrograman web lainnya.

2.10.3 PHP

Menurut (MADCOM, 2016), “PHP (*Hypertext Preprocessor*) adalah bahasa *script* yang dapat ditanamkan atau disisipkan ke dalam HTML. PHP banyak dipakai untuk membuat program situs web dinamis”. PHP dapat digunakan dengan gratis (*free*) dan bersifat *Open Source*. PHP dirilis dalam lisensi *PHP license*. Untuk membuat program PHP kita diharuskan untuk menginstal web *server* terlebih dahulu. PHP mendukung komentar seperti pada bahasa ‘C’, ‘C++’, dan Unix shell-style. (*Perl style*).

2.10.4 CSS

CSS adalah suatu bahasa pemrograman web yang digunakan untuk mengendalikan dan membangun berbagai komponen dalam web sehingga tampilan web akan lebih rapi, terstruktur, dan seragam (Saputra, Agus, & Agustin, 2013).

2.10.5 Laragon

Laragon adalah *universal* development environment portabel, terisolasi, cepat & kuat untuk PHP, Node.js, Python, Java, Go, *Ruby*. Aplikasi ini cepat, ringan dan mudah digunakan. Aplikasi ini juga sangat bagus untuk membangun dan mengelola aplikasi web modern.

2.10.6 MySql

Menurut (Raharjo, 2011) “*MySQL* merupakan software RDBMS (atau software database) yang dapat mengelola database dengan sangat cepat, dapat menampung data dalam jumlah sangat besar, dapat diakses oleh banyak *user* (*multi-user*), dan dapat melakukan suatu proses secara sinkron atau berbarengan (*multi-treaded*).

2.10.7 *JavaScript*

JavaScript adalah bahasa *script* yang disisipkan pada kode HTML dan diproses di sisi klien yaitu sering disebut *client side*. Bahasa ini menjadikan dokumen HTML menjadi semakin luas (Riandi, 2018).

2.10.8 *Bootstrap*

Menurut (Nugroho & Setiyawati, 2019), *bootstrap* adalah *framework* css untuk membuat tampilan web. *Bootstrap* menyediakan *class* dan komponen yang sudah siap dipakai.

2.10.9 *Laravel*

Laravel merupakan *framework* bahasa pemrograman PHP yang memiliki banyak fitur dan sangat membantu *developer* dalam membangun sebuah aplikasi berbasis web (Tahir, Rais, & HS, 2019). *Laravel* adalah sebuah web *development framework* yang didesain untuk meningkatkan kualitas aplikasi dengan mengurangi beban biaya pengembangan dan memudahkan proses *maintenance* serta meningkatkan produktifitas pekerjaan dengan kode program yang rapi dan terstruktur (Luthfi, 2017). *Laravel* memiliki beberapa kelebihan diantaranya: menggunakan *Command Line Interface* (CLI) *Artisan*, dapat menggunakan *package* manager PHP *Composer*, Penulisan kode program yang rapi, singkat dan terstruktur, dan mudah dimengerti *developer*.

2.11 Pengujian Perangkat Lunak

2.11.1 Pengujian *Black Box*

Pengujian *Black Box* berfokus pada persyaratan fungsional perangkat lunak yang memungkinkan *engineers* untuk memperoleh set kondisi *input* yang sepenuhnya akan melaksanakan persyaratan fungsional untuk sebuah program (Pressman R. , 2010).

Menurut (Mustaqbal & Firdaus, 2015) *Black Box Testing* berfokus pada spesifikasi fungsional dari perangkat lunak. *Tester* dapat mendefinisikan kumpulan kondisi *input* dan melakukan pengujian pada spesifikasi fungsional program. *Black Box Testing* bukanlah solusi alternatif dari *White Box Testing* tapi lebih

merupakan pelengkap untuk menguji hal-hal yang tidak dicakup oleh *White Box Testing*. *Black Box Testing* cenderung untuk menemukan hal-hal berikut:

1. Fungsi yang tidak benar atau tidak ada
2. Kesalahan antarmuka (*interface errors*)
3. Kesalahan pada struktur data dan akses basis data
4. Kesalahan performansi (*performance errors*)
5. Kesalahan inisialisasi dan terminasi