

BAB 2

TINJAUAN PUSTAKA

Pada bab dua akan menjelaskan tentang semua pengertian atau teori yang terkait dengan penelitian dimana sumbernya bisa berasal dari buku, jurnal, artikel, dan lain-lain. Hal ini dilakukan untuk memahami dari teori atau pengertian yang akan digunakan pada penelitian. Teori yang terdapat pada penelitian yaitu, *Arsip, E-Arsip, System Development Life Cycle, Personal Extreme Programming, Unified Modeling Language, Use Case Diagram, Activity Diagram, Class Diagram, Entity Relationship Diagram, Blackbox Testing*, dan penelitian terdahulu.

2.1 Kelurahan Sepinggan

Kelurahan Sepinggan merupakan sebuah instansi pemerintahan yang memiliki tugas untuk menyelenggarakan urusan pemerintahan, pemberdayaan dan pelayanan masyarakat serta ketenteraman dan ketertiban umum serta lingkungan hidup dalam satu wilayah Kelurahan yang berada di wilayah kerja Kelurahan. Kelurahan berkedudukan dibawah Kecamatan dan bertanggung jawab kepada Camat. Kelurahan Sepinggan melayani masyarakat dalam hal pengurusan surat pernyataan ahli waris, surat kuasa ahli waris, surat keterangan domisili usaha, surat keterangan, surat pengantar akta kematian, surat keterangan bertempat tinggal sementara, dan surat keterangan untuk menikah. Berikut ini merupakan visi, misi, dan struktur organisasi dari Kelurahan Sepinggan.

VISI

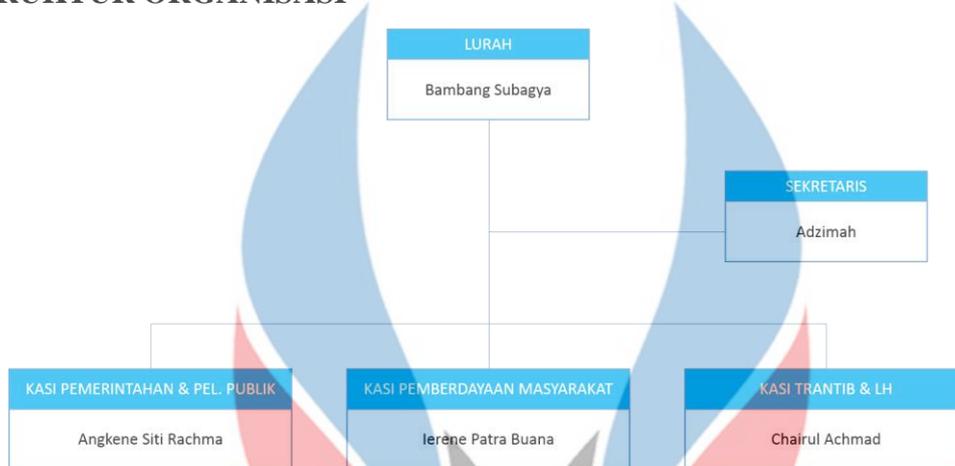
Terwujudnya Pelayanan Prima Kepada Masyarakat Kelurahan Sepinggan.

MISI

1. Meningkatkan kualitas pelayanan kepada masyarakat.
2. Melaksanakan program peningkatan kesejahteraan masyarakat.
3. Meningkatkan pemberdayaan masyarakat.
4. Menyediakan data yang akurat dan terkini.

5. Meningkatkan aparaturnya Kelurahan dalam rangka pengembangan sumber daya manusia.

STRUKTUR ORGANISASI



Gambar 2.1 Struktur Organisasi Kelurahan Sepinggan
(Sumber: Siti, 2021)

2.2 Arsip

Arsip dapat diartikan secara umum merupakan wujud atau berbentuk tulisan dalam bentuk bercorak teknis, keadaannya tunggal maupun berkelompok, atau suatu kesatuan yang bentuknya dari fungsi dalam usaha merencanakan, melaksanakan dan menyelenggarakan kehidupan secara umumnya, lalu arsip secara dapat diartikan secara khusus merupakan sekumpulan surat atau bahan sebagai suatu pengingat dalam administrasi negara yang bentuknya dapat berupa fisik atau yuridis (sesuai dengan ketentuan hukum yang berlaku) sesuai dengan aturan organisasi, yang disimpan secara baik dan dikelola atau dirawat selama dibutuhkan (Pritasari, Zaidiah, & Widi, 2020).

Arsip berbentuk sebuah catatan rekaman kegiatan yang telah dibuat oleh lembaga, organisasi, instansi, dan perorangan. Arsip dapat berbentuk piagam, buku, surat dan sebagainya. Kemudian terdapat arsip yang bentuknya teknologi yang mudah digunakan, atas bantuan kemajuan dan perkembangan teknologi maka arsip bisa berupa video, suara, dan digital. Kerasipan merupakan suatu bentuk mengelola perekapan jejak kegiatan serta bernilai kegunaannya yang telah diatur dan

www.itk.ac.id

direncanakan sehingga mudah ditemukan kembali jika diperlukan suatu hari nanti (Octafiona, Kesuma, & Bashori, 2020).

2.3 *E-Arsip*

E-Arsip merupakan bentuk sistem pengarsipan yang tentunya dapat digunakan oleh seluruh staf maupun pegawai yang ada instansi pemerintahan atau non pemerintah yang berguna dalam hal pemantauan maupun pengelolaan sesuatu yang ada kaitannya dengan sistem kearsipan yang berbentuk berkas, suara, *folder*, dan visual (Irawan & Simargolang, 2018).

Sistem kearsipan elektronik pada dasarnya memiliki konsep yang sama dengan teknik kearsipan konvensional. Jika pada kearsipan konvensional memiliki kabinet yang secara fisik berfungsi untuk menyimpan dokumen-dokumen penting yang dimiliki perusahaan, maka sistem kearsipan elektronik memiliki kabinet *virtual* yang di dalamnya berisi map *virtual* atau *folder*. Selanjutnya di dalam *folder* akan berisi lembaran-lembaran arsip yang telah dikonversi ke dalam bentuk *file* Gambar (*.bmp, jpg, dll) atau dokumen (*.doc, txt, dll). Beberapa manfaat penggunaan sistem pengelolaan secara elektronik yang mendorong sebagian besar organisasi untuk mengimplementasikan manajemen arsip elektronik diantaranya adalah cepat ditemukan, fleksibel, kecil kemungkinan akan hilang, risiko kerusakan surat kecil, mudah diawasi, dan mudah melakukan *recovery* data (Latif & Pratama, 2015).

Seperti yang telah dijelaskan sebelumnya sistem kearsipan elektronik berguna dalam mengarsipkan dokumen-dokumen penting dalam bentuk berkas, suara, *folder*, dan visual secara *virtual* dengan berbagai manfaatnya. Hal ini berkaitan juga dengan penelitian yang dilakukan dimana pembuatan sistem informasi pengelolaan surat ini bertujuan untuk memudahkan pekerjaan instansi dalam mengelola surat yang ada. Mulai dari pencatatan, pendisposisian, hingga pengarsipan. Sehingga risiko-risiko yang ada dapat diminimalisir, selain itu dapat mempercepat alur suratnya.

2.4 *System Development Life Cycle*

Systems Development Life Cycle (SDLC) digunakan untuk proses menciptakan maupun mengubah sistem yang ada saat ini atau yang berlaku. SDLC paling banyak digunakan dalam sistem informasi, rekayasa sistem, dan rekayasa perangkat lunak. SDLC dapat dianggap sebagai sebuah konsep yang telah digunakan oleh banyak metodologi pengembangan perangkat lunak, saat ini tersedia di pasar atau industri perangkat lunak. SDLC menyediakan kerangka kerja dalam membuat, merencanakan, dan mengendalikan sistem informasi apa pun yang akan dikembangkan dan dibangun (Amlani, 2012).

Empat tahapan yang terdapat didalam proses pembangunan maupun pengembangan sistem informasi dengan penggunaan SDLC adalah tahap *planning*, *analysis*, *design*, dan *implementation*. Proses pengimplementasian metode ini terdapat beberapa metodologi yang bisa digunakan. Proses pemakaian metodologinya nanti akan selalu bervariasi dan tergantung kepada kebutuhan pengembangnya, untuk probis atau data pendukungnya (Budi, Siswa, & Abijono, 2016).

2.5 *Personal Extreme Programming*

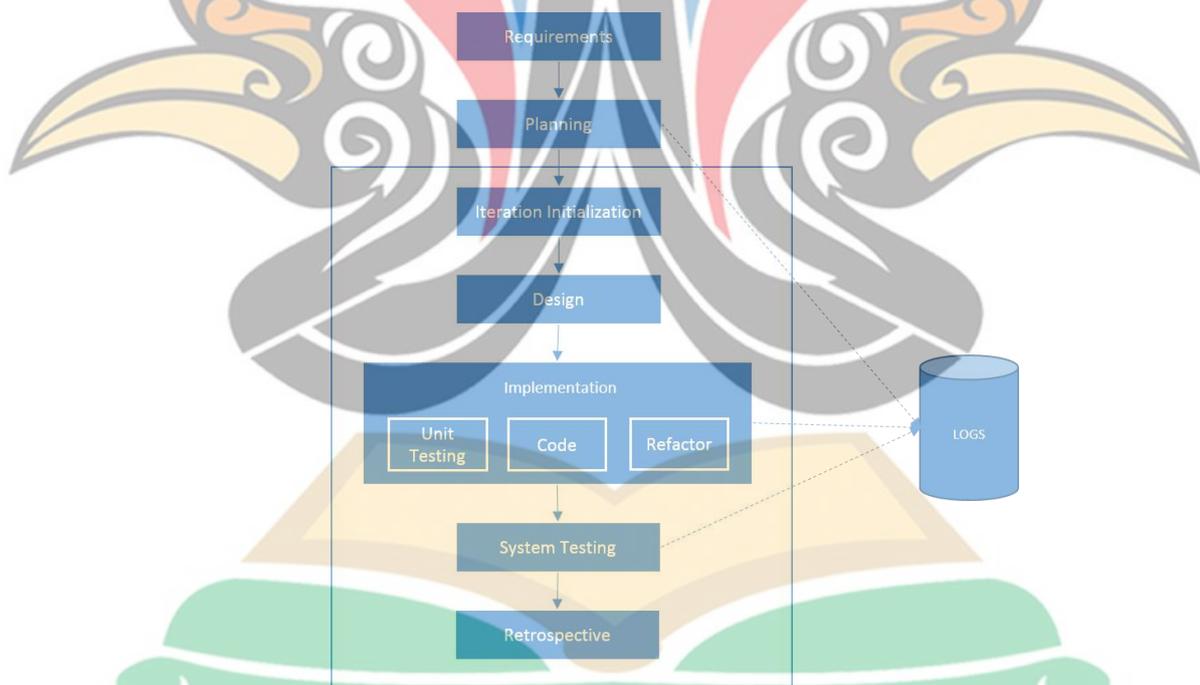
Personal Extreme Programming adalah sebuah pengembang perangkat lunak yang didesain untuk bisa dipakai oleh *programmer* secara individu. Pengembangan dalam metode *Personal Extreme Programming* bersifat iteratif dan memberikan kebebasan *programmer* dalam menangani setiap terjadinya perubahan *requirements*. Dengan menggunakan *Personal Extreme Programming* setiap *user story* dibagi kedalam *task-task* yang lebih kecil atau terperinci. *Personal Extreme Programming* menuntut seorang *programmer* atau *developer* untuk mampu bertanggungjawab atas seluruh tugas yang telah dibebankan, mengukur dan menganalisis hasil pekerjaannya sendiri. Dalam *Personal Extreme Programming* pengetesan sistem selalu dilakukan secara berkelanjutan, sehingga perubahan apapun dapat dilakukan pada saat awal dimana biaya masih rendah-rendahnya (Asri, 2015). Adapun keuntungan dan batasan pada *Personal Extreme Programming* sebagai berikut (Amlani, 2012).

Keuntungan:

- a. *Customer* berada di tempat dan langsung mendapatkan umpan balik karena setiap *developer* melakukan presentasi progress kepada *customer*, maka *customer* dapat mengetahui hal apa saja yang harus dikoreksi.
- b. Kesalahan kecil dapat cepat diketahui.
- c. Konsentrasi tetap terjaga.
- d. Cocok untuk tempat pelatihan proyek perangkat lunak besar dan kecil.
- e. *Customer* dapat melihat langsung pengerjaan proyek.
- f. Fleksibel.

Kekurangan:

- a. *Developer* bisa egois diakarenakan sistem yang dibuat harus sesuai dengan keinginannya sendiri bukan keinginan dari *customer*.
- b. Membutuhkan manajemen waktu yang baik.
- c. Pekerjaan bisa selesai, namun tidak terlalu maksimal.



Gambar 2.2 Tahapan *Personal Extreme Programming*

(Sumber: Dzhurov, 2009)

Pada Gambar 2.1 merupakan tahapan dari metode *personal extreme programming*. Tahapan dibagi menjadi beberapa bagian yang diawali dengan tahap

requirements, planning, iteration initialization, design, implementation, system testing, dan terakhir *retrospective*. Tahap pertama (*requirements*) adalah langkah dimana pengembang akan melakukan pengumpulan data kebutuhan sistem yang akan dibangun nantinya melalui teknik wawancara atau melakukan analisis dokumen, setelah melakukan hal tersebut kebutuhan sistem nantinya akan dibuat ke dalam bentuk *user story*. Tahap *planning* adalah tahap dimana pengembang melakukan pembuatan *user story* dalam bentuk fungsionalitas sistem, selain itu tahap ini pengembang akan melakukan pembuatan jadwal pembangunan sistem. Tahap *iteration initialization* adalah tahap melakukan penjabaran secara terperinci dari tahap *planning* ke dalam bentuk diagram *Unified Modeling Language* (UML). Tahap *design* adalah tahap dimana pengembang akan melakukan pemodelan sistem yang akan dibangun nanti berupa *user interface* dan ERD sebagai desain databasenya. Tahap *implementation* adalah tahap dimana sistem ini dibangun, pengembang akan melakukan pembangunan sistem berdasarkan fungsionalitas sistem yang telah dijabarkan dan apa yang telah di desain. Tahap *system testing* merupakan tahap dimana fungsionalitas yang sudah dibangun pada tahap sebelumnya akan diuji dan dilihat hasilnya apakah terdapat kekurangan atau tidak dengan metode *Black Box Testing*. Tahap terakhir (*retrospective*) adalah langkah dimana melakukan verifikasi kepada daftar yang ada di *user story* yang sudah diimplementasikan dalam tahap *implementation* serta telah diuji kedalam *system testing*. Proses verifikasi berdasarkan perbandingan antara waktu realisasi dengan waktu estimasi. Apabila telah ditemukan perbedaan, maka akan dilakukan pencarian mengapa terjadi *under* atau *over* estimasi agar tidak terulang pada proyek selanjutnya (Ardiansyah, 2018).

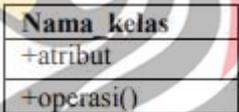
2.6 Unified Modeling Language

Unified Modeling Language (UML) merupakan sebuah spesifikasi standar atau bahasa yang digunakan dalam hal pendokumentasian, menspesifikasikan dan pembangunan perangkat lunak. UML adalah metodologi yang digunakan kedalam hal pengembangan sistem berorientasi objek sekaligus diagram pendukung pengembangan sistem (Hendini, 2016).

Unified Modelling Language (UML) adalah kumpulan dari pemodelan yang digunakan dalam proses penentuan atau penggambaran sebuah sistem yang terhubung dengan objek. *Unified Modelling Language* (UML) adalah salah satu diagram yang dapat membantu untuk digunakan dalam pengembangan sistem yang orientasinya objek. UML telah menyediakan bahasa pemodelan visual yang dapat memudahkan *developer* sistem mudah membuat *blueprint* berdasarkan visi misi kedalam bentuk yang lebih baku, sehingga gampang dimengerti dan dilengkapi dengan proses mekanisme dimana cukup efektif dalam hal pengkomunikasian rancangan dengan klien (Anisah & Kuswaya, 2017).

UML yang akan digunakan dalam penelitian ini adalah *class diagram*. *Class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan di buat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi. Berikut merupakan symbol-simbol yang ada pada *class diagram* (Aprianti & Maliha, 2016). Berikut merupakan simbol-simbol yang ada di *Class Diagram*, bisa dilihat pada Tabel 2.1.

Tabel 2.1 Simbol Class Diagram

Simbol	Deskripsi
 <p>Kelas</p>	Kelas pada struktur sistem
 <p>Interface</p>	Sama dengan konsep interface dalam pemrograman berorientasi objek
 <p>Asosiasi/association</p>	Relasi antar kelas dengan makna umum, asosiasi biasanya disertai dengan <i>multiplicity</i>
 <p>Asosiasi berarah/<i>directed association</i></p>	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>

Simbol	Deskripsi
 Generalisasi	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)
 Kebergantungan/ <i>dependency</i>	Relasi antar kelas dengan makna kebergantungan antar kelas
 Agregasi/ <i>aggregation</i>	Relasi antar kelas dengan makna semua bagian

2.7 User Story

User Story adalah suatu fungsionalitas yang bertugas untuk menuliskan kebutuhan klien serta menggantikan *requirements* (IEEE 830 dan *use case*). *Requirements* pada *waterfall* dengan *user story* berbeda, kemudian bisa dilihat *requirements* mengatakan bagaimana sebuah sistem dikatakan dapat berfungsi, dan *user story* mengatakan hal apa saja yang ingin dilakukan oleh *user*. Kemudian *requirements* digunakan selama aplikasi digunakan. Sedangkan *user story* tidak dibutuhkan lagi apabila pada tahap pembangunan aplikasi telah usai.

Didalam *user story* terdapat beberapa kelebihan yaitu dalam proses berkomunikasi antara pengembang dan klien dapat terbantu, dapat dipahami kedua belah pihak, sifat metodologinya iterasi dimana spesifikasi sistem mengalami banyak perubahan sistem. Rata-rata proyek *Personal Extreme Programming*, *user story* bisa ditulis, melakukan pengimplementasi kode, serta melakukan pengujian sistem secara langsung. *User story* dibentuk oleh klien menggunakan kertas dan sebagai media berdiskusi atau komunikasi antara *developer* dan klien selama masih didalam proses pengembangan aplikasi (Ardiansyah, 2018).

User story yang baik pastinya berbentuk model INVEST yaitu *Independent*, *Negotiable*, *Valuable*, *Estimable*, dan *Small*. *Independent* artinya tidak tergantung diantara *user story*, *Negotiable* artinya keseluruhan detail yang telah didapatkan melalui proses kerjasama, *Valuable* artinya memiliki nilai bagi penggunanya,

Estimable artinya dapat diprediksi dengan baik dalam segi waktu dan biaya, *TesTabel* artinya menentukan *acceptance criteria* (Azdy & Rini, 2018).

2.8 *Laravel*

Menurut Chen Et Al, *Laravel* merupakan *framework* web yang bersifat terbuka dan gratis. *Laravel* digunakan dalam hal pengembangan *web applications* yang dimana menggunakan standar arsitektur MVC (*Model, View, dan Controller*). *Framework Laravel* mudah untuk dipahami dalam hal *authentication, routing, session manager, caching*, dan sebagainya. Didalam *Laravel* telah menyediakan beberapa fitur *database* untuk *migration* maupun integrasi unit *testing support* sehingga dapat memudahkan pengembang dalam hal pembangunan aplikasi kompleks (Somya & Nathanael, 2019).

Laravel adalah sebuah *framework* web berbasis PHP yang *open-source* dan tidak berbayar, diciptakan oleh Taylor Otwell dan diperuntukkan untuk pengembangan aplikasi web yang menggunakan pola MVC. Struktur pola MVC pada *laravel* sedikit berbeda pada struktur pola MVC pada umumnya. Di *laravel* terdapat *routing* yang menjembatani antara *request* dari *user* dan *controller*. Jadi *controller* tidak langsung menerima *request* tersebut (Sari & Wijanarko, 2019).

Laravel Framework merupakan sebuah kerangka kerja pemrograman yang berbasis *open-source* yang dipakai oleh banyak *developer* dari seluruh dunia. *Laravel* mengikuti pola arsitektur *Model-View-Controller* (MVC). MVC memisahkan aplikasi berdasarkan komponen-komponen aplikasi, seperti manipulasi data, *controller*, dan *user interface*. Keuntungan pengembangan aplikasi ini adalah dalam proses *maintenance* dan *scalability* yang lebih mudah (Wijanarko & Budi, 2019).

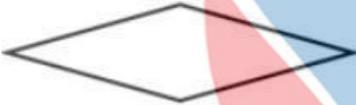
2.9 *Entity Relationship Diagram*

Entity Relationship Diagram (ERD) merupakan sebuah pendekatan yang berfungsi untuk mengGambarkan hubungan antar model. Hal paling utama dalam ERD adalah menunjukkan hubungan *entity* beserta atributnya.

Entity Relationship Diagram (ERD) merupakan sebuah bentuk pemodelan atau pengGambarkan data sehingga mambantu dalam hal mengorganisasi data dalam

suatu proyek. Nantinya data-data tersebut akan berbentuk entitas-entitas dan menentukan hubungannya. Simbol-simbol yang ada didalam ERD dapat dilihat sebagai berikut: (Fridayanthie & Mahdiati, 2016). Berikut merupakan simbol-simbol pada ERD, bias dilihat pada Tabel 2.2.

Tabel 2.2 Simbol ERD

Simbol	Deskripsi
 Entitas	Merupakan sesuatu yang nyata dan memiliki karakteristik dimana akan disimpan datanya
 Relasi	Merupakan hubungan yang dimiliki dan terjadi diantara entitas
 Atribut	Merupakan ciri-ciri khusus yang dimiliki oleh entitas
 Link	Merupakan garis penghubung antara atribut ke entitas dan antara entitas ke entitas

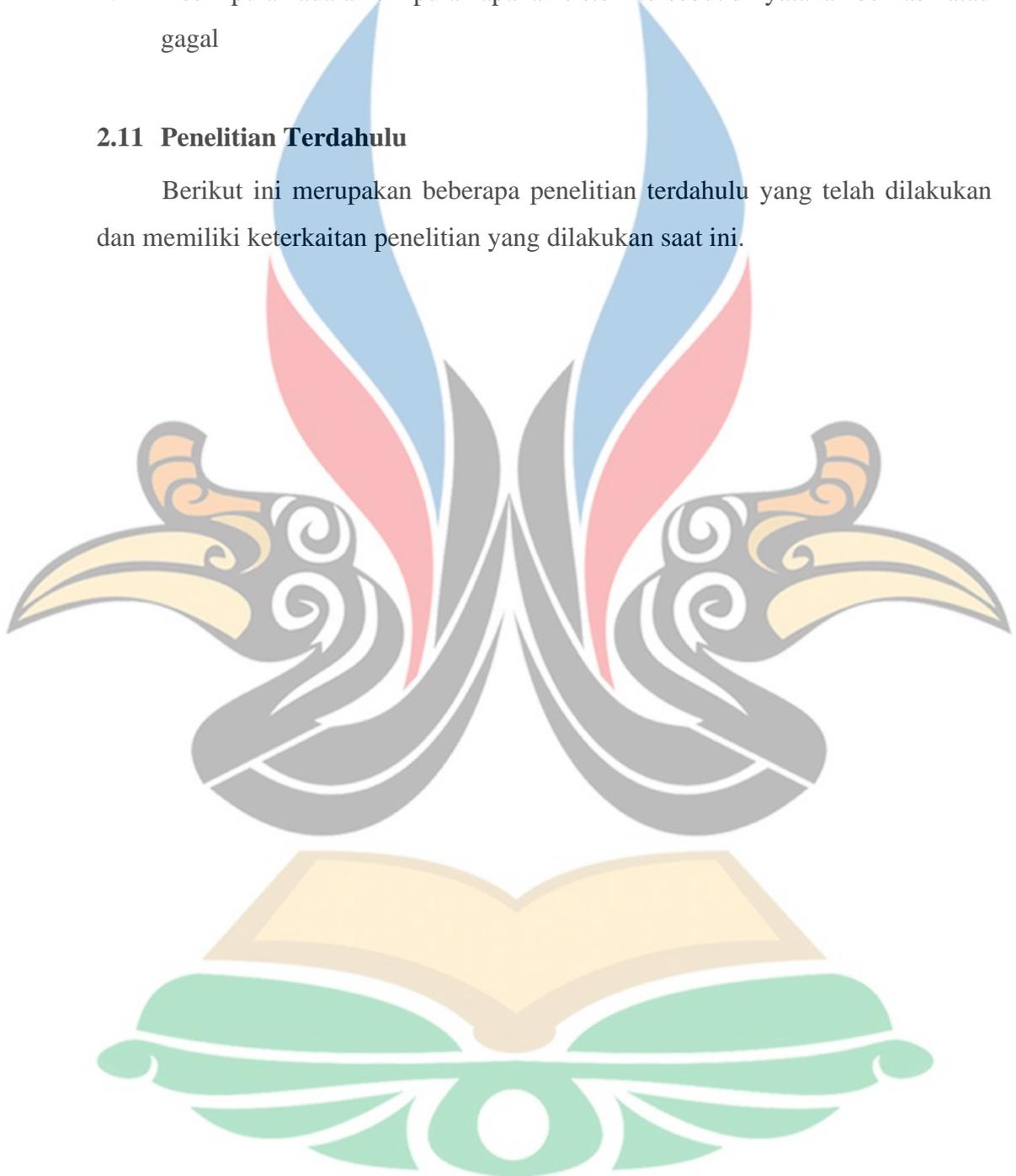
2.10 Blackbox Testing

Black Box Testing sendiri merupakan pengujian yang dilakukan hanya mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari perangkat lunak. Pengujian black box ini menitik beratkan pada fungsi sistem. Pengujian menggunakan metode ini terbagi menjadi dua yaitu, memasukkan data yang benar dan memasukkan data acak atau salah untuk memastikan sistem menolak untuk melakukan fungsinya. Pada hasil pengujian terdapat tabel *test case* yang berfungsi untuk menyimpulkan apakah sistem berhasil dalam pengujian tipe tersebut atau tidak. Berikut terdapat beberapa penjelasan dari tabel pengujian yang akan digunakan: (Hanifah, Alit, & Sugiarto, 2016)

1. Input adalah penjelasan tentang masukkan data benar atau data acak
2. Hasil harapan adalah hasil dari harus terjadi ketika dalam proses pengujian
3. *Output* adalah hasil dari pengujian setelah sistem selesai diujikan
4. Kesimpulan adalah simpulan apakah sistem tersebut dinyatakan berhasil atau gagal

2.11 Penelitian Terdahulu

Berikut ini merupakan beberapa penelitian terdahulu yang telah dilakukan dan memiliki keterkaitan penelitian yang dilakukan saat ini.



Tabel 2.3 Penelitian Terdahulu

No	Judul	Penulis	Tahun	Permasalahan	Metode	Hasil Penelitian	Fitur Aplikasi
1.	RANCANG BANGUN SISTEM PENGARSIPAN SURAT KEDINASAN BERBASIS WEB MENGGUNAKAN FRAMEWORK CODEIGNITER	Puja Irawan, Dimas Aulia Pudjie Prasetya, Petrus Sokibi	2020	Sekolah Menengah Kejuruan ASYIFA DEPOK yang terletak di Desa Cikeduk Kecamatan Depok Kabupaten Cirebon yang masih melakukan pengarsipan surat dalam bentuk kertas (<i>hardcopy</i>). Proses pengarsipan di sekolah tersebut tidak efektif, karena pengagendaan, disposisi, dan pengarsipan suratnya dilakukan dengan cara manual.	<i>Waterfall</i>	Terciptanya aplikasi yang dapat membantu pihak sekolah khususnya bagian TU dalam hal pencarian surat, pendataan surat dan pembuatan laporan surat serta meminimalisir risiko kehilangan surat.	<i>Login, logout</i> , tambah surat masuk dan keluar, tambah pengarsipan surat masuk dan keluar, pendisposisian surat, laporan grafik surat, laporan surat masuk dan keluar, cari surat masuk dan keluar.
2.	Rancang Bangun Sistem Informasi Pengelolaan Surat dan Kearsipan pada Sekretariat Daerah Provinsi NTB Berbasis Website	Sumiati, Sri Endang Anjarwani & Moh. Ali Albar	2018	Pada Sekretariat Daerah Provinsi NTB, pengelolaan dan pencatatan surat masuk maupun surat keluar masih manual yaitu menggunakan buku dan excel yang akan bisa menyebabkan terjadi kerusakan, hilang, dan lain-lain.	PXP	Menciptakan sebuah aplikasi bernama Sistem Informasi Pengelolaan Surat dan Kearsipan. Tujuan pembuatan adalah untuk mempermudah staff Ekspedisi dan Arsip dalam proses pengelolaan surat, pengarsipan surat, rekapitulasi surat, <i>monitoring</i> surat, verifikasi surat, dan pencarian surat.	<i>Login, logout</i> , CRUD surat masuk dan keluar, disposisi surat masuk, lampiran surat masuk dan keluar, cetak surat.

No	Judul	Penulis	Tahun	Permasalahan	Metode	Hasil Penelitian	Fitur Aplikasi
3.	RANCANG BANGUN SISTEM INFORMASI MANAJEMEN DISPOSISI SURAT MENYURAT DENGAN MENGGUNAKAN MODEL RAD	Istianti Elyana, Ishak Kholil, Frans Eduard Schaduw	2019	Kantor Desa Tanjungsari Kotawinangun Kebumen ini dimana bentuk pelayanan surat menyuratnya masih menggunakan sistem yang konvensional. Bentuk permasalahan yang sering timbul adalah sulitnya pelaporan surat masuk dan keluar dan pencarian data surat masuk dan surat keluar.	RAD	Terciptanya sebuah aplikasi disposisi surat yang dapat membantu pekerjaan Kantor Desa Tanjungsari dalam mengelola surat.	<i>Login, logout,</i> tambah surat, hapus surat, simpan surat, cari surat, disposisi surat, laporan surat masuk dan keluar.
4.	RANCANG BANGUN SISTEM INFORMASI PENGELOLAAN SURAT DAN KEARSIPAN PADA SEKRETARIAT DAERAH PROVINSI NTB BERBASIS WEB	Sumiati	2017	Pada Sekretariat Daerah Provinsi NTB khususnya pada sub bagian ekspedisi dan arsip yang menangani proses pengelolaan dan pencatatan data-data surat masuk dan surat keluar seperti tanggal terima, tanggal kirim, tanggal surat, nomor surat, isi surat atau perihal, penerima, pengirim, dan jenis surat masih menggunakan prosedur-prosedur kerja manual.	<i>Waterfall</i>	Terciptanay sebuah aplikasi yang dapat digunakan untuk mempermudah staff Ekspedisi dan Arsip dalam proses pengelolaan surat masuk dan surat keluar, pengarsipan surat masuk dan surat keluar, rekapitulasi laporan surat masuk dan surat keluar, monitoring disposisi surat masuk dan verifikasi status surat masuk.	<i>Login, logout,</i> cari surat, CRUD surat masuk dan keluar, disposisi surat, CRUD arsip surat masuk dan keluar, cetak laporan surat masuk dan keluar, grafik surat masuk dan keluar.
5.	Analisis dan Desain Sistem Informasi Pengarsipan Surat Masuk dan Surat Keluar di Perusahaan Daerah	Puji Rahayu	2017	Pengarsipan surat di PDAM Salatiga masih mempunyai kendala. Kendala yang sering terjadi yaitu sering terjadi kesalahan dalam pencatatan (<i>human error</i>), kehilangan	<i>Waterfall</i>	Menghasilkan sistem informai yang dapat mengelola surat masuk maupun surat keluar dan digunakan oleh Bagian Umum untuk membantu	<i>Login, logout,</i> CRUD surat masuk dan keluar, tombol <i>refresh, print, browse.</i>

No	Judul	Penulis	Tahun	Permasalahan	Metode	Hasil Penelitian	Fitur Aplikasi
	Air Minum (PDAM) Kota Salatiga			surat, rusaknya kertas surat, membutuhkan waktu yang cukup lama dalam melakukan pencarian surat, serta surat keluar hanya dilakukan dalam pencatatan saja sehingga tidak ada duplikasi dokumen. Kendala yang sering terjadi membuat kinerja dalam pengarsipan surat menjadi tidak efektif dan efisien.		meningkatkan efektifitas dan efisiensi dalam proses pengarsipan surat masuk dan surat keluar. Pencarian surat yang dilakukan berulang-ulang tidak akan terjadi lagi dan datanya aman.	



Pada tabel 2.1 adalah hasil penelitian sebelumnya yang sudah dilakukan. Dilihat dari penelitian diatas ditarik kesimpulan bahwa proses pengelolaan surat yang ada di perusahaan dan instansi masih dilakukan secara konvensional atau manual, akibatnya terjadi banyak kejadian seperti sulitnya mencari surat ketika kembali dibutuhkan, terdapat surat yang hilang, terdapat surat yang rusak, dan sebagainya. Hal tersebut yang menjadi akar permasalahan pada penelitian. Berdasarkan permasalahan di atas dapat diberikan solusi berupa pembuatan sistem informasi yang dapat mengelola proses surat-menyurat yang ada di perusahaan baik saat pencatatan surat maupun pengarsipannya, sehingga dapat mengurangi atau menghilangkan risiko-risikonya. Kemudian dari fitur-fitur diatas yang telah digunakan, tentunya ada beberapa fitur juga yang akan digunakan juga pada penelitian ini. Pertama fitur *login* dan *logout*, fitur ini digunakan karena nantinya Kepala Kelurahan dan Kepala tiap kasi bisa masuk ke dalam aplikasi menggunakan *username* dan *password* yang telah ditentukan, selain itu digunakan untuk keamanan aplikasi karena tidak sembarang orang bisa *login* ke dalam aplikasi. Kedua fitur CRUD baik di semua surat, dan saat pengarsipan surat. Fitur ini nantinya digunakan dalam proses pencatatan data, menghapus data, mengedit data, dan melihat data. Ketiga fitur *search*, fitur ini nantinya akan digunakan dalam mencari surat yang diinginkan secara singkat sehingga bisa menghemat waktu. Keempat fitur cetak laporan semua surat, nantinya fitur ini digunakan dalam proses perekapan surat secara otomatis.

