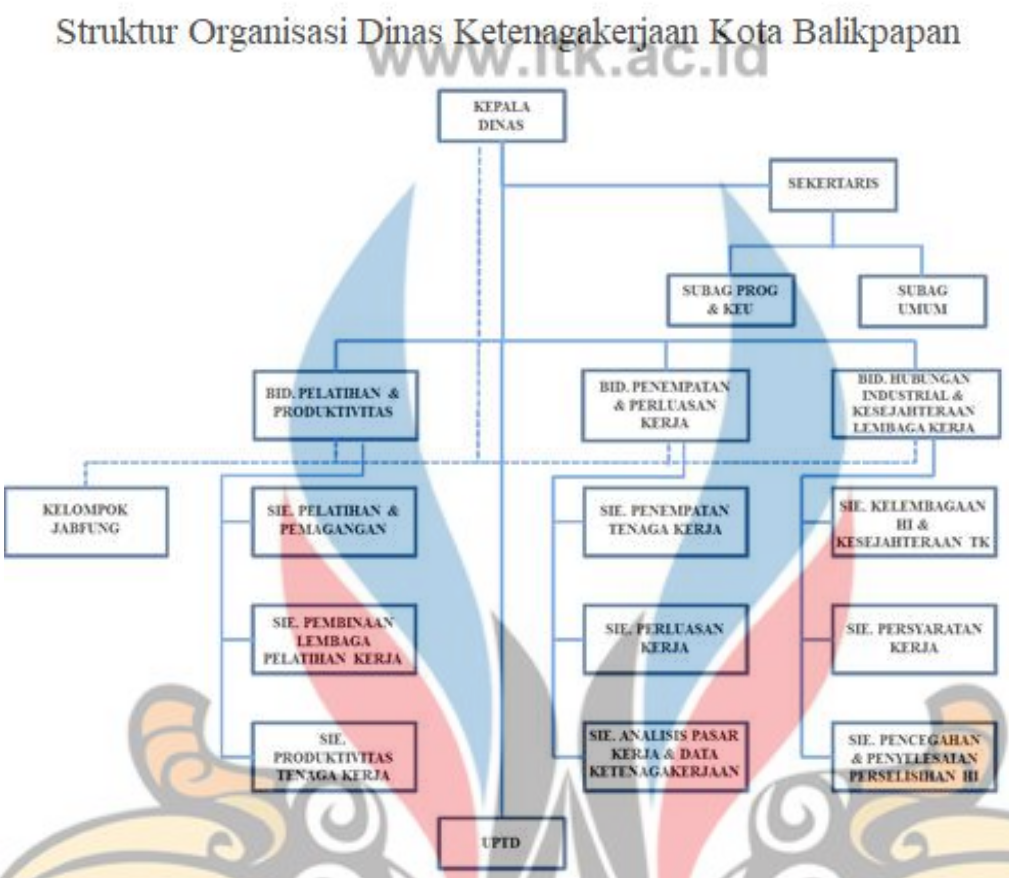
BAB 2
TINJAUAN PUSTAKA

Pada bab tinjauan Pustaka ini menjelaskan mengenai teori-teori terkait penelitian yang bersumber dari buku, jurnal, atau artikel. Tujuannya adalah untuk memahami konsep dan teori penyelesaian permasalahan yang digunakan. Teori yang terdapat pada bab ini adalah Profil Mitra Tugas Akhir, Sistem Informasi Manajemen HI Kesja, *Framework Bootstrap*, *Requirements Engineering*, UML, ERD, Metode *Waterfall*, Dokumen Perancangan Sistem, dan Penelitian Terdahulu.

2.1 Dinas Ketenagakerjaan Kota Balikpapan

Dinas Ketenagakerjaan Kota Balikpapan (Disnaker) adalah dinas milik Pemerintah Kota Balikpapan yang melaksanakan dan menjadi penunjang pemerintah di bidang Ketenagakerjaan yang dimana fungsi Disnaker sendiri yaitu membina, mengendalikan dan memberikan pengawasan dalam bidang ketenagakerjaan serta memberikan pelatihan kepada calon pekerja agar memiliki kemampuan yang sesuai dengan permintaan perusahaan yang sedang mencari tenaga kerja, memberikan kesempatan kerja secara luas, meningkatkan pelayanan penempatan tenaga kerja, serta memberikan informasi pasar dan bursa kerja. Berdasarkan Peraturan Walikota Balikpapan Nomor 37 tahun 2009, Dinas Ketenagakerjaan memiliki tugas menyelenggarakan urusan bidang ketenagakerjaan dan bidang sosial yang menjadi kewenangan Pemerintah Kota serta tugas pembantuan lainnya sesuai ketentuan peraturan perundangan yang berlaku. Dinas Ketenagakerjaan Kota Balikpapan memiliki 4 misi yaitu, meningkatkan kompetensi dan produktifitas tenaga kerja, meningkatkan perluasan kesempatan kerja, meningkatkan perlindungan tenaga kerja dan pengembangan kelembagaan Hubungan Industrial, serta meningkatkan kinerja organisasi. Terdapat 4 bidang diantaranya Bidang Sektertariat, Bidang Pelatihan dan Pemagangan, Bidang Penempatan Tenaga Kerja, dan Bidang Hubungan Industrial dan Kesejahteraan Tenaga Kerja (HI Kesja) Adapun struktur organisasi Dinas Ketenagakerjaan Kota Balikpapan dapat dilihat pada gambar dibawah ini (Disnaker, 2017).



Gambar 2.1 Struktur Organisasi Dinas Ketenagakerjaan Kota Balikpapan

Pada Gambar 2.1, Adapun struktur organisasi Dinas Ketenagakerjaan Kota Balikpapan dipimpin oleh seorang Kelapa Dinas yang dibawahi oleh Sekretariat yang membawahkan Subbagian Program dan Keuangan dan Subbagian Umum, Bidang Pelatihan dan Produktivitas Tenaga Kerja, yang membawahkan Seksi Pelatihan dan Pemagangan, Seksi Pembinaan Lembaga Pelatihan Kerja, dan Seksi Produktivitas Tenaga Kerja, Bidang Penempatan dan Perluasan Kerja, yang membawahkan Seksi Penempatan Tenaga Kerja, Seksi Perluasan Kerja, dan Seksi Analisis Pasar Kerja dan Data Ketenagakerjaan, Bidang Hubungan Industrial dan Kesejahteraan Tenaga Kerja, yang membawahkan Seksi Kelembagaan Hubungan Industrial dan Kesejahteraan Tenaga Kerja, Seksi Persyaratan Kerja, dan Seksi Pencegahan dan Penyelesaian Perselisihan Hubungan Industrial, UPT, dan Kelompok Jabatan Fungsional (Disnaker, 2017; Disnaker, 2017). Adapun yang

menjadi fokusan utama dalam penelitian ini adalah pada bidang Hubungan Industrial dan Kesejahteraan Tenaga Kerja (HI Kesja).

Bidang Hubungan Industrial dan Kesejahteraan Tenaga Kerja sebagaimana dimaksud dalam Pasal 3 ayat (1) huruf e mempunyai tugas merencanakan, mengoordinasikan, melaksanakan dan mengendalikan kegiatan bidang Hubungan Industrial dan Kesejahteraan Tenaga Kerja. Bidang Hubungan Industrial dan Kesejahteraan Tenaga Kerja sebagaimana dimaksud pada ayat (1) dipimpin oleh kepala bidang yang berada di bawah dan bertanggung jawab kepada Kepala Dinas. Bidang Hubungan Industrial dan Kesejahteraan Tenaga Kerja membawahkan seksi dan setiap seksi dipimpin oleh kepala seksi yang bertanggungjawab kepada kepala bidang (Effendi, 2016). Terdapat 3 Seksi pada bidang HI Kesja yakni, Seksi Kelembagaan Hubungan Industrial dan Kesejahteraan Tenaga Kerja (KHI Kesja), Seksi Persyaratan Kerja (Syaker), dan Seksi Pencegahan dan Penyelesaian Perselisihan Hubungan Industrial (P3HI).

Seksi Kelembagaan Hubungan Industrial dan Kesejahteraan Tenaga Kerja sebagaimana dimaksud dalam Pasal 3 ayat (1) huruf e angka 1, mempunyai tugas (Effendi, 2016):

1. Menyusun program dan kegiatan seksi Kelembagaan Hubungan Industrial dan Kesejahteraan Tenaga Kerja;
2. Menyiapkan bahan kebijakan teknis di bidang Kelembagaan Hubungan Industrial dan Kesejahteraan Tenaga Kerja;
3. Menyiapkan pembinaan dan bimbingan teknis hubungan industrial;
4. Menyiapkan pembinaan pembentukan dan bimbingan teknis Lembaga Kerjasama Bipartit dan lembaga lainnya di perusahaan;
5. Melaksanakan fasilitasi Lembaga Kerjasama Tripartit;
6. Menyiapkan pembinaan Organisasi Pekerja dan Organisasi Pengusaha;
7. Melaksanakan verifikasi pencatatan Organisasi Pekerja;
8. Melaksanakan pencatatan Organisasi Pekerja;
9. Melaksanakan verifikasi keanggotaan Organisasi Pekerja/Buruh;
10. Menyiapkan pembinaan kesejahteraan tenaga kerja;
11. Menyiapkan pembinaan fasilitas kesejahteraan pekerja/buruh;

12. Melaksanakan pemantauan dan pelaporan kelembagaan Hubungan Industrial dan Kesejahteraan Tenaga Kerja;
13. Melaksanakan monitoring, evaluasi, pengendalian dan pelaporan pertanggungjawaban pelaksanaan tugas; dan
14. Melaksanakan tugas lainnya yang diberikan oleh pimpinan/atasan sesuai dengan bidang tugasnya.

Seksi Persyaratan Kerja sebagaimana dimaksud dalam Pasal 3 ayat (1) huruf e angka 2 mempunyai tugas (Effendi, 2016):

1. Menyusun program dan kegiatan seksi Persyaratan Kerja;
2. Menyiapkan bahan kebijakan teknis di bidang persyaratan kerja;
3. Melaksanakan penyuluhan dan bimbingan teknis penyusunan Perjanjian Kerja, Peraturan Perusahaan dan Perjanjian Kerja Bersama;
4. Melaksanakan verifikasi Perjanjian Kerja, Peraturan Perusahaan dan Perjanjian Kerja Bersama;
5. Melaksanakan pencatatan Perjanjian Kerja, pengesahan Peraturan Perusahaan dan pendaftaran Perjanjian Kerja Bersama;
6. Melaksanakan penyuluhan dan pembinaan penyerahan sebagian pelaksanaan pekerjaan kepada perusahaan lain;
7. Menyiapkan bahan pembinaan pelaporan jenis pekerjaan penunjang (alur kegiatan proses pelaksanaan pekerjaan) perusahaan pemberi pekerjaan;
8. Menyiapkan bahan pembinaan pendaftaran perjanjian antara Perusahaan pemberi pekerjaan dengan perusahaan penerima pekerjaan;
9. Menyiapkan bahan pembinaan dan bimbingan teknis Upah Minimum dan struktur skala upah;
10. Melaksanakan fasilitasi Dewan Pengupahan;
11. Melaksanakan pemantauan dan pelaporan syarat kerja dan penyerahan sebagian pelaksanaan pekerjaan;
12. Melaksanakan monitoring, evaluasi, pengendalian dan pelaporan pertanggungjawaban pelaksanaan tugas; dan
13. Melaksanakan tugas lainnya yang diberikan oleh pimpinan/atasan sesuai dengan bidang tugasnya.

2.2 Sistem Informasi Manajemen HI Kesja

Menurut Hartanto, Sistem Informasi Manajemen (SIM) merupakan rangkaian yang terstruktur dari sejumlah bagian yang secara bersama-sama berfungsi menghasilkan informasi yang selanjutnya akan digunakan dalam memanajemen sebuah perusahaan (Hartono, 2013). Sedangkan menurut Heryati, Sistem Informasi Manajemen (SIM) merupakan suatu sistem informasi yang bertujuan untuk menyajikan berbagai informasi secara lebih luas. Sistem infromasi manajemen menyediakan informasi yang digunakan untuk mendukung manajemen dalam mengambil keputusan. Konsep dalam SIM adalah sistem informasi merupakan unsur meningkatkan nilai perusahaan (Heryati, 2017). Sistem Informasi Manajemen HI Kesja yang dimaksud adalah suatu sistem berbasis komputer yang dapat memanajemen pendataan dan menyediakan informasi untuk pada pegawai/seksi yang ada pada bidang HI Kesja dengan kebutuhan yang serupa. SIM HI Kesja yang akan dibangun merupakan sebuah sistem berbasis *website*, dengan menggunakan bahasa pemrograman PHP (*Hypertext Preprocessor)*, database server MySQL (*My Structure Query Languange*) dan *framework bootstrap.*

PHP singkatan dari *Perl Hypertext Preprocessor* yaitu bahasa pemrograman yang bersifat open source. PHP adalah script yang terintergrasi dalam HTML dan digunakan untuk membuat halaman *web* dinamis. Dinamis berarti halaman yang akan ditampilkan dibuat saat halaman itu diminta oleh *client*. Semua script PHP akan dieksekusi pada *server* dimana *script* tersebut dijalankan (Fridayanthie, 2016).

My Structure Query Languange atau yang biasa disingkat dengan MySQL merupakan salah satu jenis *database server* yang digunakan dalam membuat sebuah *software* dalam bentuk *web* yang menggunakan database sebagai pengelolaan dan sumber datanya. MySQL jugs bersifat *open source* dan menggunakan *Structured Query Languange* atau yang biasa disingkat dengan SQL. MySQL dapat dijalankan diberbagai platform seperti windows Linux, dan lain sebagainya (Fridayanthie, 2016).

2.3 *Framework Bootstrap*

Bootstrap adalah seperangkat aplikasi atau template desain web yang dipakai untuk membuat *front-end* sebuah *website*. *Bootstrap* digunakan agar

mempermudah proses mendesain sebuah *website* untuk berbagai tingkat pengguna. *Bootstrap* dapat digunakan dengan cukup bermodalkan pengetahuan dasar mengenai CSS dan HTML dan CSS (Christian, 2018).

Bootstrap memiliki keunggulan serta kekurangan antara lain sebagai beriut (Adhiasta, 2018):

1. Keunggulan dari *Bootstrap* diantaranya adalah waktu dalam pembuatan *website* menjadi lebih cepat karena elemen umum yang biasa ada dalam sebuah *website* sudah dibuatkan *class*-nya oleh *Bootstrap*. Selain itu, template dengan menggunakan *Bootstrap* lebih rapi dan ringan, serta tersedia template *Bootstrap* yang *responsive* dan tidak *responsive.*
2. Kekurangan dari *Bootstrap* adalah mudah dikenai. Hal ini akan menjadi buruk ketika sesuatu telah menjadi terkenal, maka akan semakin banyak orang yang mudah mengenali *bootstrap*. Misalnya bagi para pengembang *front-end website* akan merasa tidak asing ketika melihat sebuah template yang dibuat dengan menggunakan *bootstrap*. Hal ini akan menjadi sedikit masalah ketika diminta untuk membuat sebuah template yang berbeda dari template *boostrap* lainnya.

2.4 *Requirements Engineering*

Requirements engineering merupakan proses mencari tahu, menganalisis, mendokumentasikan dan memeriksa layanan yang dibutuhkan dan kendala yang terdapat pada pengembangan sistem. Suatu sistem memiliki kebutuhan yang menguraikan tentang apa yang harus dilakukan oleh sistem beserta dengan layanan yang disediakannya dan kendala dalam operasinya. Selain itu kebutuhan dari sistem memperlihatkan kebutuhan pelanggan untuk sistem yang melayani suatu tujuan seperti mengendalikan perangkat, melakukan pemesanan, ataupun mencari informasi. Hasil dari *requirements engineering* ini digunakan sebagai dasar penawaran suatu kontrak dengan penyajian item-item yang terbuka untuk masukan dalam pembangunan *software* nantinya dan digunakan sebagai dasar kontrak dengan penyajian yang didefinisikan secara detail sehingga apa yang akan dilakukan oleh *developer* menjadi jelas (Sommerville, 2011).

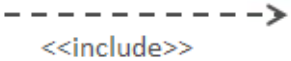
2.5 *Unified Modeling Language* (UML)

UML merupakan bahasa yang digunakan untuk mendetailkan, membangun, dan mendokumentasikan *software*. UML adalah metodologi untuk mengembangkan sebuah sistem berorientasi objek serta sebagai alat untuk mendukung pengembangan sistem (Hendini, 2016). Diagram UML yang digunakan dalam penelitian ini antara lain sebagai berikut:

1. *Use Case Diagram*

Use Case Diagram adalah gambaran perilaku (*behavior)* pada sistem yang akan dibuat. *Use case* digunakan untuk mengetahui aktifitas apa dan siapa saja yang dapat menggunakan aktifitas tersebut (Hendini, 2016). Adapun simbol yang terdapat pada *Use Case Diagram* adalah sebagai berikut yaitu:

Tabel 2.1 Simbol *Use Case*

Gambar	Keterangan
	*Use Case*: digunakan sebagai unit yang bertukar pesan antar unit dengan aktor. *Use case* dinyatakan dengan menggunakan kata kerja.
	*Actor:* menggambarkan orang atau sistem yang lain yang mengaktifkan fungsi dari system yang ditargetkan. *Actor* bisa muncul dalam beberapa peran.
	Association: komunikasi antar aktor dan *use case* yang berpartisipasi pada *use case* atau use case memiliki interaksi dengan aktor.
	Generalization: hubungan generalisasi dan spesialisasi antara dua buah *use case* dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
	*Include*: *Use case* tambahan dimana *use case* yang ditambahkan memerlukan *use case* ini untuk menjalankan fungsinya sebagai syarat.

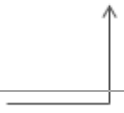
Gambar	Keterangan
<<extends>>	*Extend*: Relasi *use case* tambahan antar *use case* dimana *use case* yang ditambahkan dapat berdiri sendiri.

(Hendini, 2016)

2. *Activity Diagram*

Activity Diagram menggambarkan aktivitas atau aliran kerja dari sebuah system (Hendini, 2016). *Activity diagram* menyediakan analis yang berguna untuk memodelkan proses dalam suatu sistem informasi. Adapun simbol yang digunakan dalam *Activity Diagram* yaitu:

Tabel 2.2 Simbol *Activity Diagram*

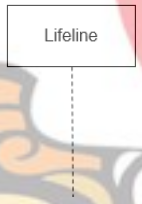
Gambar	Keterangan
	*Start Point*: menunjukkan mulainya dari aktivitas
	*End Point*: menunjukkan diakhir aktivitas
	*Activities:* menunjukkan suatu kegiatan atau proses bisnis
	*Fork*/percabangan: Digunakan untuk menggabungkan dua kegiatan paralel menjadi satu atau untuk menunjukkan kegiatan yang dilakukan secara parallel
	*Join* (penggabungan) atau *rake*: Digunakan untuk menunjukkan adanya penggabungan kegiatan
	*Decision Points*: menggambar kan pilihan untuk pengambilan keputusan.
	*Control Flow*: menunjukkan aliran dari satu kegiatan menuju kegiatan lainnya.

(Hendini, 2016)

3. *Sequence Diagram*

Sequence diagram menggambarkan interaksi yang terjadi disekitar sistem termasuk pengguna berupa *message* terhadap waktu. Biasanya *sequence diagram* digunakan untuk menggambarkan rangkaian skenario sebagai respon dari sebuah *event* dan akan menghasilkan *output* tertentu (Dharwiyanti, 2003). Simbol-simbol yang terdapat pada *Sequence Diagram* adalah sebagai berikut:

Tabel 2.3 Simbol *Sequence Diagram*

Gambar	Keterangan
	*Lifeline:* Elemen yang merepresentasikan partisipan dalam interaksi. Hanya mewakili satu entitas yang berinteraksi
	Message: Menentukan pesan dari pemanggil ke penerima dalam *sequence diagram.*
	Reply Message: Menunjukkan bahwa penerima pesan selesai memproses pesan dan mengembalikan kontrol ke pemanggil.
	*Self Message:* Objek mengirim pesan ke dirinya sendiri.

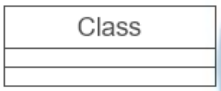
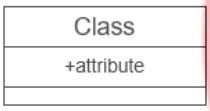
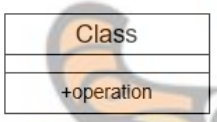
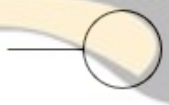
(Dharwiyanti, 2003)

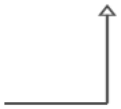
4. *Class Diagram*

Class diagram merupakan sebuah perincian yang menggambarkan struktur dan deskripsi keadaan (atribut/properti) sistem sekaligus memberikan layanan untuk mengubah keadaan tersebut (metode/fungsi). Terdapat area pokok pada sebuah *class* yaitu nama (dan *stereotype*), atribut dan metode. Atribut dan metode

dapat memiliki salah satu sifat *private* atau tidak dapat dipanggil dari luar *class* yang bersangkutan, *protected* atau hanya dapat dipanggil oleh *class* yang bersangkutan dan *child* yang mewarisinya, dan *public* atau dapat dipanggil oleh siapa saja (Dharwiyanti, 2003).

Tabel 2.4 Simbol *Class Diagram*

Gambar	Keterangan
	*Class*: sekumpulan objek yang berbagi operasi, atribut, dan hubungan yang sama. Suatu class dapat mengimplementasikan satu atau lebih *interface*.
	*Attribute*: properti dari *class* yang berisi tipe data yang ada dalam suatu *class*.
	*Operation*: kegiatan yang akan dilakukan oleh suatu *class*.
	*Interface*: antarmuka dengan konsep yang sama seperti pada pemrograman berorientasi objek.
	*Association*: relasi tiap-tiap *class* dengan makna umum yang biasanya disertai dengan *multiplicity*.
	*Directed Association: r*elasi antar *class* dengan maksud kelas yang satu digunakan oleh kelas yang lain.
	*Aggregation*: relasi antara dua atau lebih objek yang dimana salah satu objek adalah bagian dari objek yang lain.
	*Dependency*: menunjukkan operasi pada suatu *class* dengan menggunakan *class* lain.

	*Generalization*: relasi antar kelas yang lebih umum dengan kelas yang lebih spesifik dimana kelas yang spesifik mewarisi fitur pada kelas yang lebih umum.

(Dharwiyanti, 2003)

2.6 *Entity Relationship Diagram* (ERD)

ERD merupakan model teknik pendekatan yang menggambarkan atau menyatakan relasi suatu model. Relasi ini dinyatakan dengan menunjukan objek data (*Entity*) dan hubungan (*Relationship*), yang ada pada *Entity* berikutnya (Fridayanthie, 2016).

Tabel 2.5 Simbol *Entity Relationship Diagram* (ERD)

Gambar	Keterangan
	Entitas: menunjukkan entitas yang terhubungan dengan sistem
	Atribut: menunjukkan atribut yang dimiliki oleh entitas
	Relasi: menunjukkan relasi antar entitas
	Link: Sebagai penghubung antara entitas, atribut, dan relasi

(Malik, 2017)

2.7 Systems Development Life Cycle (SDLC)

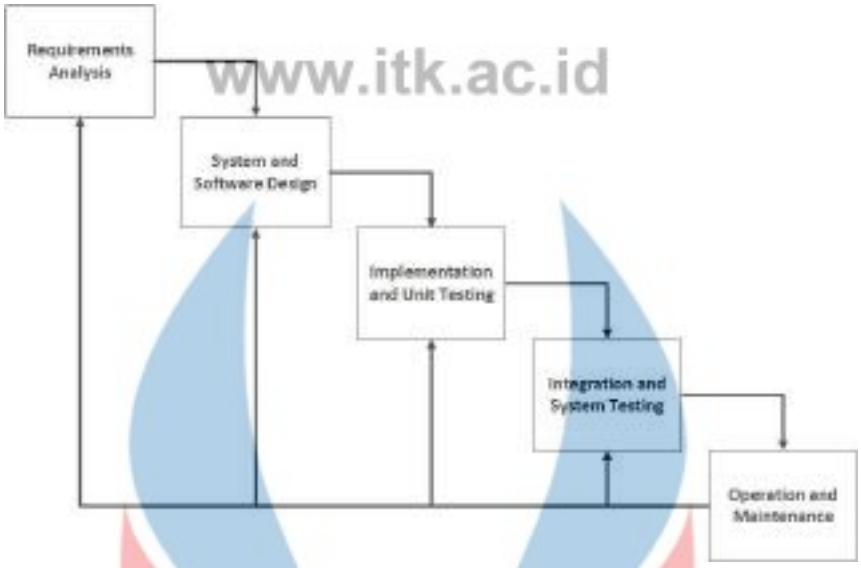
SDLC (*Systems Development Life Cycle*) dalam rekayasa sistem dan rekayasa perangkat lunak adalah proses pembuatan dan pengubahan sistem serta model dan metodologi yang digunakan untuk mengembangkan sistem-sistem tersebut. Konsep ini umumnya merujuk pada sistem komputer atau informasi. SDLC juga merupakan pola yang diambil untuk mengembangkan sistem perangkat lunak, yang terdiri dari tahap-tahap: rencana (*planning*), analisis (*analysis*), desain (*design*), implementasi (*implementation*), uji coba (*testing*) dan pengelolaan (*maintenance*).

Berikut beberapa macam metode yang digunakan dalam pengembangan sistem informasi atau aplikasi yakni *Waterfall, Prototype, RAD, Spiral, PXP,* dan *Scrum*. Masing-masing metode tersebut memiliki perbedaan yang mendasar sebagaimana dijelaskan pada sub bab 2.7.1 sampai 2.7.6. Dari perbedaan pada masing-masing metode tersebut, metode *waterfall* digunakan pada penelitian tugas akhir ini atas dasar pertimbangan sebagai berikut.

1. *Output* yang dihasilkan pada penelitian ini merupakan sebuah sistem informasi, sedangkan *output* dari *prototype model* hanya sampai *mockup* saja.
2. Penelitian ini tidak dapat diselesaikan dalam waktu kurang dari 90 hari karena harus menganalisis serta mendesain sistem sedemikian rupa agar mempermudah dan meminimalisir kesalahan pada saat pembangunan/pembuatan sistemnya.
3. Sistem yang akan dibangun termasuk sekala kecil, berdasarkan kelebihan *waterfall model* dimana model ini cocok untuk proyek-proyek berskala kecil dibandingkan *spiral model* yang akan bekerja dengan baik pada proyek berskala besar.
4. Dokumen pengembangan sistem menggunakan *waterfall model* sangat terorganisir, karena setiap fase harus terselesaikan dengan lengkap sebelum melangkah ke fase berikutnya. Dokumen-dokumen pengembangan tersebut akan mempermudah *developer* selanjutnya ketika ingin melakukan pengembangan sistem kedepannya, sedangkan dokumentasi pada metode PXP hanya secara informal.

2.7.1 *Waterfall Model*

Waterfall Model merupakan pendekatan yang dilakukan secara teratur dan berurutan untuk mengembangkan *software*. *Waterfall model* terkadang disebut *classic life cycle* (siklus hidup klasik) yang dimulai dengan perincian kebutuhan melalui *communication* yang berkembang dengan *planning, modeling, construction, deployment,* dan perawatan yang dilakukan secara terus-menerus setelah *software* selesai (Pressman, 2012).



Gambar 2.2 Model SDLC *Waterfall*

Metode ini dipilih karena proses perancangan *website* dilakukan secara bertahap, berikut ini merupakan deskripsi tahapan metode *waterfall* (Fatimah, 2018):

1. *Requirements analysis* yang mendefinisikan kebutuhan sistem yang sesuai dengan keinginan *user* yang terdapat dalam dokumen SRS.
2. *Ssystem and software design* mendefinisikan gambaran dari rancangan sistem baik perangkat keras maupun perangkat lunak yang terdapat dalam dokumen SDD yang didalamnya terdapat serangkaian fungsi-fungsi dari *software* dan gambaran dari desain *software*.
3. *Implementation* mendefinsikan hasil perancangan diubah kedalam kedalam bahasa pemograman yang dimengerti komputer.
4. *Integration and system testing* mendefinisikan hasil dari program yang selesai dibuat dilakukan pengujian apakah terdapat kesalahan atau tidak, sebelum diberikan ke *user*.
5. *Operation and maintenance* mendefinisikan bahwa sistem telah digunakan secara nyata, tahapan ini merupakan final dalam pembuatan sebuah *software* atau sistem. Setelah melakukan analisis, desain, perancangan maka sistem yang sudah jadi akan digunakan oleh *user*. Kemudian *software* yang telah dibuat dilakukan hosting pada server dan pemeliharaan secara berkala serta pengembangan.

Kelebihan dari metode *Waterfall* adalah sebagai berikut (Setiyani, 2019).

1. *Waterfall* sangat sederhana dan mudah dimengerti yang sangat bermanfaat untuk pemula dalam merancang dan membagun sebuah web.
2. Pada fase model ini diproses dan diselesaikan sekaligus dalam satu waktu sehingga dapat menghemat banyak waktu.
3. *Waterfall* ini bekerja lebih efektif dalam proyek-proyek yang berskala kecil dimana persyaratan mudah dipahami.
4. Cocok digunakan untuk produk *software*/program yang sudah jelas kebutuhannya di awal, sehingga minim kesalahannya.
5. Dokumen pengembangan sistem sangat terorganisir, karena setiap fase harus terselesaikan dengan lengkap sebelum melangkah ke fase berikutnya.

Kekurangan dari metode *Waterfall* adalah sebagai berikut (Pricillia, 2021).

1. Proyek yang sebenarnya jarang mengikuti alur sekuensial seperti diusulkan, sehingga perubahan yang terjadi dapat menyebabkan hasil yang sudah didapatkan tim pengembang harus diubah kembali/iterasi sering menyebabkan masalah baru.
2. Sulit untuk mengalami perubahan kebutuhan yang diinginkan oleh *customer*/pelanggan.
3. Perubahan ditengah-tengah pengerjaan produk akan membuat bingung tim pengembang yang sedang membuat produk

2.7.2 *Prototype Model*

Model *prototyping* merupakan suatu teknik untuk mengumpulkan informasi tertentu mengenai kebutuhan-kebutuhan informasi pengguna secara cepat. Berfokus pada penyajian dari aspek-aspek perangkat lunak tersebut yang akan nampak bagi pelanggan atau pemakai. Prototipe tersebut akan dievaluasi oleh pelanggan/pemakai dan dipakai untuk menyaring kebutuhan pengembangan perangkat lunak. *Prototype* didefinisikan sebagai alat yang memberikan ide bagi pembuat maupun pemakai potensial tentang cara sistem berfungsi dalam bentuk

lengkapnya, dan proses untuk menghasilkan sebuah *prototype* disebut *prototyping*. Sistem dengan model prototype mengijinkan pengguna agar mengetahui seperti apa tahapan sistem dibuat sehingga sistem mampu beroperasi dengan baik. Metode *prototype* digunakan untuk mendapatkan representasi dari pemodelan aplikasi yang akan dibuat. Rancangan aplikasi awal mulanya berbentuk *mockup* selanjutnya akan dievaluasi oleh pengguna. Setelah mockup dievaluasi pengguna tahap selanjutnya *mockup* menjadi bahan rujukan bagi pengembang *software* untuk merancang aplikasi (Pricillia, 2021).

Gambar 2.3 Model SDLC *Prototype*

Kelebihan dari metode *Prototype* adalah sebagai berikut (Pricillia, 2021).

1. Penentuan kebutuhan lebih mudah diwujudkan.
2. Mempersingkat waktu pengembangan produk perangkat lunak.
3. Penerapan menjadi lebih mudah karena pelanggan mengetahui apa yang diharapkannya.
4. Pelanggan berpartisipasi aktif dalam pengembangan sistem, sehingga hasil produk pengembangan akan semakin mudah disesuaikan dengan keinginan dan kebutuhan pelanggan.

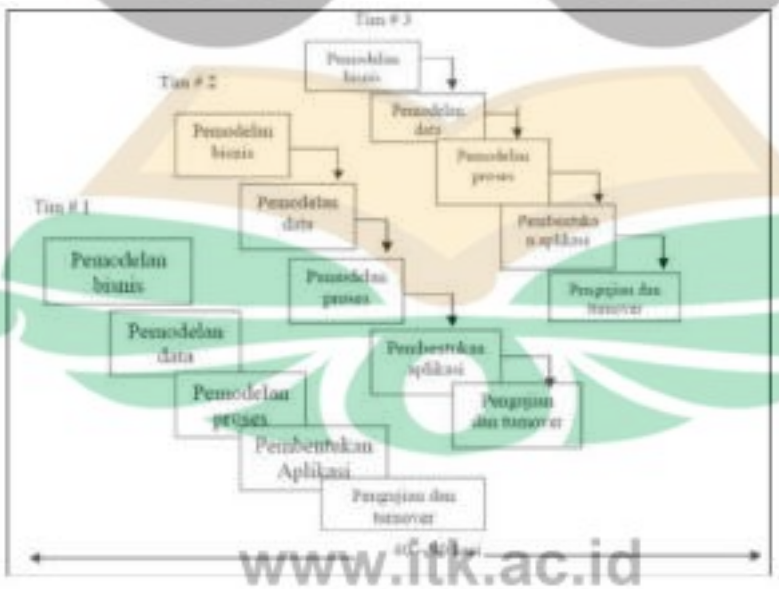
Kekurangan dari metode *Prototype* adalah sebagai berikut (Pricillia, 2021).

1. Proses analisis dan perancangan terlalu singkat.
2. Biasanya kurang fleksibel dalam mengahadapi perubahan.

3. Walaupun pemakai melihat berbagai perbaikan dari setiap versi prototype, tetapi pemakai mungkin tidak menyadari bahwa versi tersebut dibuat tanpa memperhatikan kualitas dan pemeliharaan jangka panjang.

2.7.3 *Rapid Aplication Development (RAD) Model*

Rapid Aplication Development (RAD) adalah sebuah model proses perkembangan perangkat lunak sekuensial linier yang menekankan siklus perkembangan yang sangat pendek (kira-kira 60 sampai 90 hari). Model RAD ini merupakan sebuah adaptasi "kecepatan tinggi" dari model sekuensial linier dimana perkembangan cepat dicapai dengan menggunakan pendekatan konstruksi berbasis komponen. RAD bertujuan mempersingkat waktu yang biasanya diperlukan dalam siklus hidup pengembangan sistem tradisional antara perancangan dan penerapan suatu sistem informasi. Pada akhirnya, RAD sama-sama berusaha memenuhi syarat-syarat bisnis yang berubah secara cepat RAD menggunakan metode iteratif (berulang) dalam mengembangkan system dimana working model (model bekerja) sistem dikonstruksikan di awal tahap pengembangan dengan tujuan menetapkan kebutuhan (*requirement*) pengguna dan selanjutnya disingkirkan. Dalam pengembangan sistem informasi normal, memerlukan waktu minimal 180 hari, namun dengan menggunakan metode RAD, sistem dapat diselesaikan dalam waktu 30-90 hari (Pricillia, 2021).



Gambar 2.4 Model SDLC RAD

Kelebihan dari metode RAD adalah sebagai berikut (Pricillia, 2021).

1. Cocok untuk proyek yang memerlukan waktu yang singkat.
2. Model RAD mengikuti tahap pengembangan sistem seperti pada umumnya, tetapi mempunyai kemampuan untuk menggunakan kembali komponen yang ada sehingga pengembang tidak perlu membuatnya dari awal lagi sehingga waktu pengembangan menjadi lebih singkat dan efisien.

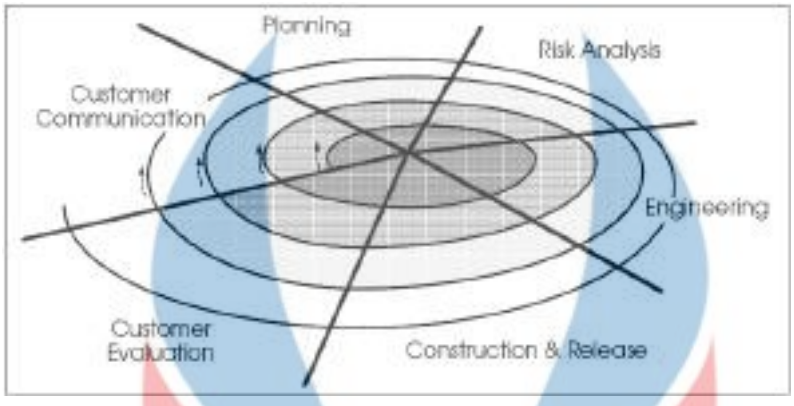
Kekurangan dari metode RAD adalah sebagai berikut (Pricillia, 2021).

1. Model RAD menuntut pengembangan dan pelanggan memiliki komitmen di dalam aktivitas *rapid-fire* yang diperlukan untuk melengkapi sebuah sistem, di dalam kerangka waktu yang sangat diperpendek. Jika komitmen tersebut tidak ada, proyek RAD akan gagal.
2. Tidak semua aplikasi sesuai untuk RAD, bila system tidak dapat dimodulkan dengan teratur, pembangunan komponen penting pada RAD akan menjadi sangat bermasalah.
3. Membutuhkan tenaga kerja yang banyak untuk menyelesaikan sebuah proyek dalam skala besar.
4. Jika ada perubahan di tengah-tengah pengerjaan maka harus membuat kontrak baru antara pengembang dan pelanggan.

2.7.4 *Spiral Model*

Model ini mengambil fitur penting dari model waterfall dan prototyping, dengan menambah elemen baru yaitu analisa resiko (*risk analysis*). Model ini memiliki 6 aktivitas penting, yaitu *Customer Communication, Planning, Risk Analysis, Engineering, Construction and release,* dan *Customer Evaluation*. Bentuk *spiral* memberikan gambaran bahwa jika iterasinya semakin besar, maka semakin lengkap versi dari perangkat lunak yang digunakan. Jika analisa resiko ada yang menunjukkan ketidakpastian terhadap kebutuhan, maka *prototyping* harus dibuat pada *kuadran engineering*. Pelanggan mengevaluasi hasil *engineering* (*kuadran customer evaluation*) dan membuat usulan untuk perbaikan. Berdasarkan masukan

dari pelanggan, fase berikutnya adalah *planning* dan analisis resiko. Setelah analisis resiko, selalu diperiksa apakah proyek diteruskan atau tidak, jika resiko terlalu besar, maka proyek dapat dihentikan (Budi, 2016).



Gambar 2.5 Model Spiral

Kelebihan dari metode Spiral adalah sebagai berikut (Budi, 2016).

1. Jumlah analisis risiko yang tinggi
2. Baik untuk proyek-proyek besar dan *mission-critical*,
3. Software diproduksi di awal siklus hidup perangkat lunak.

Kekurangan dari metode Spiral adalah sebagai berikut (Budi, 2016).

1. Dapat menjadi model mahal untuk digunakan.
2. Analisis risiko membutuhkan keahlian yang sangat spesifik.
3. Keberhasilan proyek sangat tergantung pada tahap analisis risiko.
4. Tidak bekerja dengan baik untuk proyek-proyek yang lebih kecil.

2.7.5 *Personal Extreme Programming (PXP)*

Extreme Programming (XP) adalah proses pengembangan perangkat lunak dirancang untuk diterapkan oleh pengembang perangkat lunak secara individu. XP menjaga prinsip dasar mengurangi jumlah dokumentasi dan upaya pemeliharaan. Proses pengembangan XP bersifat fleksibel dan responsif untuk perubahan. Metodologi ini ditujukan untuk meningkatkan kinerja *programmer* dan

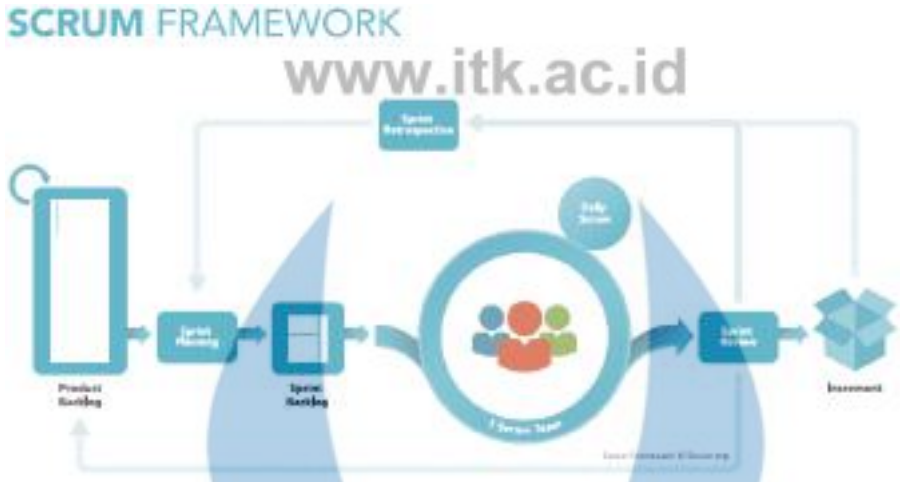
mempersingkat pengkodean dan waktu yang digunakan untuk mendukung sistem perangkat lunak (Pamungkas, 2018).

Metodologi XP didasarkan pada beberapa prinsip yaitu, PXP memerlukan pendekatan disiplin, pengembang bertanggung jawab untuk mengikuti proses dan menerapkan praktik XP, pengembang harus mengukur, melacak dan menganalisis pekerjaan sehari- hari mereka, pengembang harus belajar dari variasi kinerja mereka dan memperbaiki proses berdasarkan data proyek yang terkumpul, XP melibatkan pengujian terus menerus, perbaikan cacat harus terjadi pada tahap pengembangan awal, bila biaya itu lebih rendah, pengembang harus mencoba mengotomatisasi sebanyak mungkin hari mereka kerja (Pamungkas, 2018).

Gambar 2.6 Life Cycle PXP

2.7.6 *Scrum*

Scrum merupakan sebuah kerangka kerja dimana orang-orang dapat mengatasi masalah kompleks adaptif, dimana pada saat bersamaan dapat menghantarkan produk dengan nilai setinggi mungkin secara produktif dan kreatif. *Scrum* bersifat sederhana untuk dipahami namun sulit untuk dikuasai. *Scrum* mengekspos ketidak-efektifan dari manajemen produk dan teknik kerja anda, sehingga anda dapat secara terus-menerus meningkatkan kinerja produk, tim, dan lingkungan kerja (Pradhana, 2020).



Gambar 2.7 Framework SCRUM

Kelebihan dari metode Scrum adalah sebagai berikut (Pradhana, 2020).

1. Ukuran tim yang kecil melancarkan komunikasi, mengurangi biaya, dan memberdayakan satu sama lain.
2. Proses dapat beradaptasi terhadap perubahan teknis dan bisnis.
3. Proses menghasilkan beberapa software increment.
4. Pembangunan dan orang yang membangun dibagi dalam tim yang kecil.
5. Dokumentasi dan pengujian terus menerus dilakukan setelah software dibangun.

Kekurangan dari metode Scrum adalah sebagai berikut (Pradhana, 2020).

1. Jika tugas tidak didefinisikan dengan baik, perkirakan biaya proyek dan waktu tidak akan akurat. Dalam kasus seperti itu, tugas dapat tersebar di beberapa *Sprint*.
2. Jika anggota tim tidak berkomitmen, maka proyek tidak akan selesai atau bahkan gagal.
3. *Scrum* dapat bekerja dengan baik jika *Scrum* master mempercayai tim yang mereka kelola, jika *Scrum* master terlalu mengontrol secara ketat setiap anggota tim ini dapat menyebabkan tim menjadi stress yang mengakibatkan demoralisasi dan kegagalan dari proyek tersebut.
4. *Project quality management* sangat sulit untuk diimplementasikan dan diukur kecuali tim dapat melakukan pengujian regresi setelah beberapa *Sprint*.

2.8 Dokumen Perancangan dan Pengujian Sistem

Dalam penelitian ini terdapat dokumen perancangan dan pengujian yaitu *Software Requirements Specification* (SRS), *Software Design Description* (*SDD*), dan *Software Design Description* (*SDD*).

1. *Software Requirement Specification* (SRS)

Software Requirements Specification atau yang disingkat dengan SRS merupakan dokumen yang menjelaskan kebutuhanapa saja yang harus dipenuhi oleh suatu *software*. Dokumen SRS berisi mengenai kebutuhan pada *software* sebagai hasil dari proses analisis yang telah dilakukan. Dokumen ini dibuat setelah menggali informasi dari calon pemakai *software* (Putra, 2016). Karakterisitik yang ada pada *SRS* yang baik dijelaskan sebagai berikut (IEEE, 1984):

1. Unambiguous

Dokumen *SRS* dikatakan tidak ambigu (*unambiguous*) jika di dalamnya hanya memiliki satu tujuan dengan setiap karakteristik produk yang digambarkan menggunakan satu istilah unik. Dalam kasus tertentu terdapat istilah yang memiliki banyak arti sehingga istilah tersebut harus dimasukkan dalam daftar istilah yang artinya dibuat lebih spesifik.

2. Complete

SRS selesai (*complete*) jika semua kebutuhan penting yang berkaitan dengan fungsionalitas, kinerja, atribut, atau *eksternal interface* dimasukkan. Selain itu penjelasan tentang respon yang dilakukan *software* untuk semua kelas input data dapatt direalisasikan. Terdapat kesesuaian dengan standar *SRS* yang ada dengan pelabelan yang lengkap.

3. Verifiable

SRS dapat diverifikasi (*verifiable*) apabila setiap kebutuhan untuk melihat *software* memenuhi persyaratan atau tidak. Kebutuhan yang tidak dapat diverifikasi seperti pernyataan bahwa produk harus bekerja dengan baik atau produk harus memiliki *interface* yang bagus, hal tersebut tidak dapat diverifikasi karena tidak dapat didefinisikan lebih baik. Apabila suatu metode yang digunakan tidak dapat dirancang untuk menentukan kebutuhan *software*, maka kebutuhan tersebut harus dihapus atau direvisi.

4. *Consistent*

SRS yang konsisten (*consistent*) tidak memiliki kebutuhan individual yang dijelaskan dalam permasalahan. Terdapat 2 (dua) atau lebih kebutuhan yang menggambarkan objek yang sama tetapi dengan istilah yang berbeda. Selain itu tidak ada karakteristik yang bertentangan dari objek.

5. *Modifiable*

SRS dapat dimodifikasi (*modifiable*) apabila strukturnya sedemikian rupa sehingga setiap perubahan yang diperlukan terhadap kebutuhan dapat dilakukan dengan mudah, lengkap dan konsisten. Untuk melakukan modifikasi diperlukan *SRS* yang memiliki organisasi yang berkaitan dan mudah digunakan dengan daftar isi, indeks, dan referensi. Kebutuhan yang sama tidak boleh muncul lebih dari 1 (satu) tempat dalam *SRS*.

6. *Traceable*

SRS dapat dilacak (*traceable*) apabila asal dari masing-masing kebutuhan jelas. *Backward traceability* yaitu pelacakan ke tahap pengembangan sebelumnya tergantung pada setiap kebutuhan yang merujuk pada sumbernya pada dokumen sebelumnya. *Forward traceability* yaitu pelacakan untuk semua dokumen yang dihasilkan oleh *SRS* yang tergantung dengan kebutuhan dalam *SRS* yang memiliki nama atau nomor referensi yang unik. Ketika suatu kebutuhan dalam *SRS* mewakili pembagian dari kebutuhan lain, maka harus disediakan *backward traceability* dan *forward traceability*.

7. *Usable During the Operation and Maintenance Phase*

SRS harus memenuhi kebutuhan fase operasi dan pemeliharaan termasuk juga apabila terdapat penggantian *software*. Biasanya pemeliharaan dilakukan oleh pihak yang tidak terkait dengan pengembangan asli. Apabila terdapat perubahan kecil dapat dilakukan koreksi menggunakan *code* yang lebih baik, sementara pada perubahan dengan cakupan yang lebih luas diperlukan dokumentasi desain dan kebutuhan. Untuk melakukan pemeliharaan diperlukan pemahaman suatu fungsi agar pemeliharaan dapat dilakukan dengan baik.

SRS tidak mengikuti garis besar yang ada dalam panduan tersebut, akan tetapi harus mencakup semua informasi berikut (IEEE, 1984).

1. *Introduction*
    a. Tujuan
    b. Definisi, Istilah dan Singkatan
    c. Aturan Penomoran
    d. Referensi
    e. Deskripsi Umum Dokumen
2. Kebutuhan Perangkat Lunak
    a. Deskripsi Umum Sistem
    b. Kebutuhan Fungsional dan Non-fungsional
    c. Batasan Sistem
    d. Lingkungan Operasi
    e. Model Use Case
3. Lampiran

2. *Software Design Description* (SDD)

Software Design Description (*SDD*) merupakan gambaran dari desain software yang akan digunakan untuk menyimpan informasi desain dengan bahasan berbagai masalah desain *software*. Selama masa pemakaiannya, *SDD* digunakan oleh *project manager*, staf penjamin kualitas, *programmers*, penguji dan pengelola dengan kebutuhannya masing-masing baik dalam informasi desain, maupun pengaturan informasi yang optimal dan menjadi pemandu dalam pemilihan, organisasi dan penyajian informasi desain. Oleh karena itu deskripsi desain akan berisi informasi desain yang dibutuhkan semua *stakeholder* tersebut (IEEE, 2005).

Setiap tampilan desain ditentukan oleh sudut pandang desain yang mengidentifikasi masalah desain ke dalam bahasa desain. Beberapa bagian dalam *SDD* adalah sebagai berikut (IEEE, 2005).

1. Pendahuluan
    1.1 Tujuan
    1.2 Ruang Lingkup
    1.3 Aturan Penomoran
    1.4 Referensi
    1.5 Definisi Istilah

2. Model Analisis
    2.1 *System Sequence* Tahap Analisis
    2.2 *Class Diagram* Tahap Analisis
    2.3 *Package Diagram* Tahap Analisis
    2.4 Deskripsi Arsitektur
    2.5 Pedoman Perancangan
3. Model Perancangan
    3.1 *System Sequence* Tahap Perancangan
    3.2 *Class Diagram* Tahap Perancangan
    3.3 Perancangan Representasi Kelas Persisten
    3.4 Perancangan Antar Muka

3. *Software Test Documentation* (STD)

Software Test Documentation (STD) adalah dokumen yang berisi mengenai hasil uji fitur dari sistem yang dikembangkan atau dibangun. Selain itu, dokumen STD berisi mengenai pengujian fitur sistem, baik fitur yang berhasil atau gagal. Dokumen STD dibuat untuk memastikan apakah fitur sudah berjalan dengan baik atau sudah sesuai dengan kebutuhan serta untuk mengevaluasi fitur dari sistem yang dibuat.

2.9 Penelitian Terdahulu

Penelitian terdahulu digunakan sebagai reverensi dalam pengerjaan penelitian untuk melihat hasil penelitian yang sebelumnya sebagai referensi dalam penelitian ini. Adapun penelitian terdahulu yang memiliki keterkaitan dengan penelitian yang telah dilakukan adalah sebagai berikut:

Tabel 2.6 Penelitian terdahulu

No	Nama	Tahun	Metode	Studi Kasus	Hasil
1	Ella Pratiwi	2015	*Waterfall*	Dinas Perhubungan Provinsi Kepuauan Bangka Belitung	Sistem informasi yang dibangun dapat digunakan untuk membantu bagian kepegawaian dalam proses menjalankan tugas pengolahan data agar lebih mudah.

No	Nama	Tahun	Metode	Studi Kasus	Hasil
2	Mohamad Topan, dkk	2015	*Waterfall*	Rumah Sakit TNI AU Lanud Sam Ratulangi	Sistem Informasi yang dibangun dapat digunakan untuk mempermudah pengelolaan data pasien rumah sakit untuk pelayanan kasir, pelayanan apotik, rawat jalan, rawat inap.
3	Nely Zahroh	2015	*Rapid Aplication Development* (RAD)	Dinas Pendidikan Kab. Kulon Progo	Sistem yang dibangun dapat digunakan untuk mempermudah melakukan pendataan siswa pada Dinas Pendidikan Kab. Kulon Progo.
4	I Gede Agus Sanjaya	2016	*Waterfall*	CV. Intra Media Bali	Sistem informasi yang dibangun dapat digunakan untuk mengelola data pada CV. Intra Media Bali
5	Nila Elzha Pratiwi	2016	*Waterfall*	Dinas Ketenagakerjaan Kota Surabaya	Sistem informasi yang dibagun dapat mempermudah perusahaan yang ingin melakukan wajib lapor perusahaan. Selain itu, Sistem memfasilitasi serta mempercepat proses persetujuan wajib lapor ketenagakerjaan bagi para pihak yang bersangkutan.
6	Sukron Amin, Kondar Siahaan	2016	*Waterfall*	Sekolah Tinggi Ilmu Tarbiyah (Stit) Kabupaten Tebo	Prototipe yang dihasilkan dapat digunakan sebagai gambaran sistem yang menampilkan informasi yang berkaitan dengan peminjaman arsip dan juga pengelolaan arsip.
7	Maulana Malik	2017	*Waterfall*	LBH Makassar	Sistem informasi yang dibangun dapat

No	Nama	Tahun	Metode	Studi Kasus	Hasil
					digunakan untuk pengelolaan data serta penjadwalan konsultasi agar mempermudah para karyawan dalam menata ataupun mengelola dokumentasi - dokumentasi kasus
8	Reza Trimahardhika, Entin Sutinah	2017	*Rapid Aplication Development* (RAD)	Perpustakaan Yayasan Putra Asih Malida	Sistem informasi yang dibangun dapat digunakan untuk mempermudah petugas dalam mengelola data perpustakaan serta pembuatan laporan lebih optimal
9	Dian Puspitasari	2017	*Waterfall*	Pasar Kuwu, Grobongan	Sistem informasi yang dihasilkan dapat mempermudah penyampaian serta pengelolaan sistem informasi retribusi dan lokasi kios terkait pasar Kuwu secara online.
10	Vivin Ambar, Arisdandy Ambarita	2017	*Waterfall*	Dinas Pendidikan Nasional Kota Ternate	Sistem informasi yang dibangun dapat memberikan kemudahan dalam melakukan pengolahan data hasil kelulusan siswa no-formal serta mempermudah siswa dan orangtua dalam hal mendapatkan informasi kelulusan.
11	Yana Iqbal Maulana	2017	*Waterfall*	Dinas Pendidikan Kota Tanggerang Selatan	Sistem informasi yang dibangun dapat digunakan untuk mempermudah pendataan guru dan sekolah yang ada pada Dinas Pendidikan

No	Nama	Tahun	Metode	Studi Kasus	Hasil
					Kota Tanggerang Selatan.
12	Elgamar Syam	2018	*Waterfall*	Universitas Islam Kuantan Singingi (UNIKS)	Sistem informasi yang dibangun dapat digunakan untuk membantu pengelolaan data dosen dan mahasiswa yang selanjutnya data yang dikelola dapat memberikan gambaran serta perumusan kepada pimpinan dalam hal penerapan kebijakan terhadap dosen dan mahasiswa.
13	Anista Yulia R, Edy Sudanto, Edy Susena	2019	*Waterfall*	Dinas Perindustrian dan Perdagangan Kab. Sragen	Rancangan sistem yang dihasilkan adalah gambaran sistem informasi manajemen yang dapat mempermudah manajemen pendataan serta penyajian data laporan industri kecil dan menengah
14	Elvi Yanti, Effiyaldi	2019	*Waterfall*	Pengadilan Agama Jambi Kelas 1A	Prototype yang dihasilkan, merupakan gambaran sistem informasi layanan persidangan yang meliputi pendaftaran perkara, informasi panjar biaya perkara, dan jadwal persidangan.
15	Hendra K, Wicakso Bandung B	2019	*Waterfall*	Institut Informatika dan Bisnis Darmajaya	Sistem informasi yang dihasilkan meliputi aktifitas mulai dari pengajuan judul, pendaftaran, penjadwalan sampai penentuan nilai akhir TA.

No	Nama	Tahun	Metode	Studi Kasus	Hasil
16	Raditya Wardhana, Ade Pujianto	2019		Perpustakaan Univ. AMIKOM Yogyakarta	Sistem infromasi yang dihasilkan dapat mempermudah dan mempercepat pengunjung dalam hal menemukan lokasi rak koleksi buku yang dicari
17	Wafa Pamulasari, Nana Suryana	2020	*Waterfall*	BPJS Ketenagakerjaan Cabang Sukabumi	Sistem informasi yang dibangun dapat mempermudah para bidang terkait untuk melihat disposisi surat dan mengelola data persuratan yang ada.

Penentuan penelitian terdahulu berdasarkan 3 aspek, yaitu mengenai pengelolaan data, penjadwalan, dan penyusunan tata letak arsip. Pada aspek pengelolaan data, mengacu pada penelitian Ella Pratiwi tahun 2015 dengan permasalahan yaitu pengolahan data pegawai, kenaikan jabatan, permohonan cuti, SK mutasi, kenaikan pangkat, kenaikan gaji berkala masih dilakukan secara manual dan tertulis. Tahapan pada penelitian ini adalah *Project Execution Plan* (PEP), Identifikasi *Stakeholder*, Identifikasi *Deliverables*, Penjadwalan Proyek, dengan menggunakan (Pratiwi, 2015). Selanjutnya adalah pada penelitian Moh. Topan, dkk pada tahun 2015, dengan permasalahan yaitu pengelolaan data pasien untuk kasir, rawat inap, rawat jalan, dan apotik masih dilakukan secara. Tahapan pada penelitian ini adalah *communication, planning, modeling,* dan *construction* (Topan, 2015). Selanjutnya pada penelitian Nely Zahroh pada tahun 2015 dengan permasalahan yaitu, model sistem informasi pendataan siswa belum optimal dan bersifat semi-otomatis. Sistem informasi pendataan siswa meliputi data sekolah, data penerimaan siswa, daftar sekolah asal, data siswa total, data siswa mengulang, putus sekolah, dan mutasi, data siswa inklusi, dan data siswa penerima KPS. Tahapan pada penelitian ini yaitu, masalah dan potensi, pengumpulan data, analisis sistem, perancangan sistem, uji coba struktur kendali, revisi, uji coba unit, revisi, uji coba integrasi, revisi, uji coba sistem, revisi, ujicoba penerimaan, revisi, validasi produk, model produk final (Zahroh, 2015). Selanjutnya pada penelitian I Gede Agus Sanjaya pada tahun 2015 dengan permasalahan yaitu sistem yang ada pada saat ini

terkomputerisasi sangat rendah dalam hal mengelola data pembelian, data karyawan, data barang, data servis, data biaya operasional, dan data penjualan pada perusahan. Tahapan pada penelitian ini yaitu, deskripsi sistem, spesifikasi sistem, perancangan sistem, data flow diagram, design antar muka aplikasi (Sanjaya, 2015). Selanjutnya pada penelitian Sukron Amin dan Kondar Siahaan pada tahun 2016 dengan permasalahan yaitu, sistem informasi yang ada pada saat ini belum optimal dan dalam hal penataan arsip kampus masih dilakukan secara manual. Pendataan yang terdapat pada penelitian ini yaitu, data admin, data dosen, dan, data dokumen. Tahapan yang ada pada penelitian ini yaitu, identifikasi masalah, literatur, pengumpulan data, analisis sistem, protoype (Amin, 2016). Selanjutnya pada penelitian Dian P pada tahun 2017 dengan permasalah sistem yang ada pada saat ini masih bersifat manual yang menyulitkan penyebaran informasi terkait pasar sampai letak lokas kios. Tahapan pada penelitian ini yaitu studi Pustaka, observasi, wawancara, analisa kebutuhan, perancangan sistem, pembuatan sistem, pengujian sistem, implementasi (Puspitasari, 2017). Selanjutnya pada penelitian Vivin A dan Arsandy A pada tahun 2017 dengan permasalahan pengelolaan data hasil kelulusan siswa non-formal yang meliputi proses penginputan, penyimpanan dan informasi hasil kelulusan masih dilakukan secara manual dan tertulis sehingga pencarian data menjadi lama (Ambar, 2017).Selanjutnya pada penelitian Yana I pada tahun 2017 dengan permasalahan yaitu pendataan guru dan sekolah masih dilakukan secara manual pada Ms. Excel sehingga pendataan kurang optimal. Tahapan pada penelitian ini yaitu, perancangan, analisis, desain, implementasi (Maulana, 2017). Selanjutnya pada penelitian Elgamar S pada tahun 2018 dengan permasahan yaitu, Belum ada sistem yang dapat memanajemen pengelolaan data dikarenakan institusi masih terbilang perguruan tinggi yang baru. Tahapan pada penelitian ini yaitu, identifikasi masalah, perumusan masalah, studi literatur, analisa sistem, pengumpulan data, perancangan sistem (desain dan program), pengolagan data, dan hasil (Syam, 2018). Selanjutnya pada penelitian Anista Y, dkk pada tahun 2019 dengan permasalahan yaitu, pada saat ini pengelolaan data Industri Kecil dan Menengah (IKM) masih dilakukan secara manual dengan menggunakan media Ms. Excel sehinggal dalam melakukan pendataan, pengelolaan data, pengarsipan, dan penyajian laporan masih belum opotimal. Tahapan pada penelitian ini yaitu,

perumusan masalah, pengumpulan data, analisis sistem, desain sistem, pengkodean, pengujian, hasil penelitian (Ratnawati, 2019). Selanjutnya pada penelitian Wafa P dan Nana S pada tahun 2020 dengan permasalahan pengelolaan data surat masih dilakukan secara manual dengan menggunakan selembar kertas berupa kartu selanjutnya menyalin data surat keluar dan masuk kedalam Ms. Excel sehingga proses administrasi belum optimal. Tahapan pada penelitian ini yaitu pengumpulan dan indentifikasi kebutuhan sistem, membuat prototype, mengevaluasi prototype, pengkodean, pengujian sistem, evaluasi sistem, dan implementasi sistem (Pamulasari, 2020).

Mengenai aspek penjadwalan mengacu pada penelitian Elvi Y dan Effriyaldi pada tahun 2019 dengan permasalahan terjadinya keterlambatan penyelesaian proses perkara dikarenakan minimnya informasi yang diperoleh pihak yang berkara. Cakupan data pada penelitian ini meliputi jadwal persidangan, laporan tahap perkara, laporan biaya perkara, dan laporan perkara masuk perbulan. Tahapan pada penelitian ini yaitu, identifikasi masalah, studi literatur, pengumpulan data, analisa dan perancangan, dan pembuatan laporan (Yanti, 2019). Selanjutnya pada penelitian Hendara K dan Wicaksono B pada tahun 2019 dengan permasalahan proses kegiatan TA masih dilakukan dalam bentuk *hard* dokumen yang pembuat proses kegiatan TA kurang optimal. Sistem informasi yang ada pada penelitian ini mencakup informasi mahasiswa, informasi jadwal ujian, informasi ujian, informasi dosen penguji, informasi nilai ujian. Tahapan pada penelitian ini yaitu, analisis, desain, code, dan test (Kurniawan, 2019).

Mengenai aspek penyusunan tata letak arsip mengacu pada penelitian Maulana Malik pada tahun 2017 dengan permasalahan yaitu, penyimpanan serta dokumentasi kasus belum dikelola dengan baik. Pendataan pada penelitian ini meliputi data kasus, data pemohon, data konsultasi. Tahapan pada penelitian ini yaitu, analisis masalah, analisis kebutuhan, perancangan sistem, implementasi, dan pengujian sistem (Malik, 2017). Selanjutntya pada penelitian Reza T dan Entin S pada tahun 2017 dengan permasalahan yaitu, pengelolaan data perpustakaan masih dilakukan secara manual dan menggunakan media kertas untuk pengolahan data anggota, peminjaman dan pengendalian buku, dan pembuatan laporan. Tahapan

pada penelitian ini yaitu, *Fase Requirements Planning, Fase RAD Design Workshop, Fase Instruction,* dan *Fase Implementation* (Trimahardhika, 2017). Selanjutnya pada penelitian Raditya W dan Ade P pada tahun 2019 dengan permasalahan yang terjadi sulitnya pengunjung baru ketika ingin mencari buku yang diinginkan dikarnakan sistim yang ada saat ini tidak mencantumkan informasi detail serta tidak ada lokasi rak buku. Penelitian ini mencakup pendataan pengelolaan buku, rak, kategori, pengarang, dan penerbit. Tahapan pada penelitian ini yaitu analisis sistem yang berjalan, analisis kelemahan sistem, solusi terhadap masalah, penggunaan aplikasi, perancangan proses, perancangan basis data, perancangan table, perancangan antarmuka sistem, dan rancangan pengujian (Wardhana, 2019).