

BAB 2

LANDASAN TEORI

2.1. Teori Umum

2.1.1. Sistem Informasi

Satzinger, Jackson, dan Burd (2012:p5), mengatakan bahwa sistem adalah sekumpulan kegiatan–kegiatan yang memungkinkan pengguna untuk mendefinisikan dan mendeskripsikan secara jelas di dalam memecahkan suatu permasalahan atas kebutuhan pengguna sistem tersebut. Menurut Romney dan Steinbart (2015:p3), sistem adalah suatu rangkaian yang terdiri dari dua atau lebih komponen yang saling berhubungan dan saling berinteraksi satu sama lain untuk mencapai tujuan dimana sistem biasanya terbagi dalam subsistem yang lebih kecil yang mendukung sistem yang lebih besar. Sedangkan Mulyadi (2016:p5) berpendapat bahwa sistem adalah jaringan prosedur yang dibuat menurut pola yang terpadu untuk melaksanakan kegiatan-kegiatan pokok perusahaan.

Menurut Satzinger, Jackson, dan Burd (2012:p4), sistem informasi adalah seperangkat dari kumpulan komponen–komponen yang saling terkait di dalam mengumpulkan, memproses, menyimpan (biasanya terjadi didalam database) dan menyediakan suatu keluaran berupa informasi yang dibutuhkan pengguna sistem di dalam menyelesaikan suatu tujuan dari tugas-tugas bisnis yang ada. Menurut Rainer, Prince, dan Cegielski (2014:p7), sistem informasi (*information system/IS*) merupakan sistem yang mengumpulkan, memproses, menyimpan, menganalisis, dan mendistribusikan informasi untuk tujuan tertentu. Sedangkan menurut O'Brien dan Marakas (2014:p4), sistem informasi dapat berupa kombinasi terorganisir dari beberapa orang, *hardware*, *software*, jaringan komunikasi, sumber daya, data, prosedur dan peraturan yang disimpan, dikumpulkan, ditransformasikan, dan dibagikan kepada organisasi.

Menurut Rainer, Prince, dan Cegielski (2014:p15-16), komponen-komponen dasar sistem informasi berbasis komputer tercantum di bawah ini, dimana empat pertama disebut komponen teknologi informasi.

- a. *Hardware*, terdiri dari perangkat seperti prosesor, monitor, keyboard, dan printer. Bersama-sama, perangkat ini menerima, memroses, dan menampilkan data dan informasi.
- b. *Software*, adalah sebuah program atau kumpulan program yang memungkinkan perangkat keras untuk memroses data.
- c. *Database*, adalah kumpulan *file* terkait atau tabel yang berisi data.
- d. Jaringan, adalah sistem yang menghubungkan (kabel atau nirkabel) yang memungkinkan komputer yang berbeda untuk berbagi sumber.
- e. Prosedur, adalah instruksi untuk menggabungkan komponen-komponen di atas untuk memroses informasi dan menghasilkan *output* yang diinginkan.
- f. Manusia, yaitu individu-individu yang menggunakan perangkat keras dan perangkat lunak, berinteraksi dengan perangkat keras dan perangkat lunak, atau memanfaatkan *output*-nya.

Menurut Rainer, Prince, dan Cegielski (2014:p17), keuntungan utama penggunaan sistem informasi adalah kemampuan sistem informasi dalam hal:

- a. Kemampuan melakukan perhitungan numerik dengan kecepatan dan volume yang tinggi.
- b. Menyediakan sarana komunikasi dan kolaborasi yang cepat dan akurat di dalam organisasi dan antar organisasi.
- c. Menyimpan sejumlah besar informasi dalam ruang yang relatif kecil namun mudah untuk diakses.
- d. Memungkinkan akses cepat dan murah untuk sejumlah besar informasi dari seluruh bagian dunia.
- e. Menafsirkan data dalam jumlah besar dengan cepat dan efisien.
- f. Mengotomatisasi baik proses bisnis yang semi-otomatis maupun tugas manual.

Menurut Rainer, Prince, dan Cegielski (2015:p38), sistem informasi dalam sebuah organisasi dapat dibagikan menjadi beberapa tipe, yaitu:

- a. *Functional Area IS*: Mendukung kegiatan dalam sebuah area fungsional tertentu.
- b. *Transaction Processing System*: Mengolah data transaksi dari kegiatan bisnis.

- c. *Enterprise Resource Planning*: Mengintegrasikan seluruh area fungsional yang berada dalam perusahaan.
- d. *Office Automation System*: Mendukung kegiatan sehari-hari individu maupun kelompok dalam perusahaan.
- e. *Management Information System*: Membuat laporan yang dirangkum dari data transaksi, umumnya pada area fungsional tertentu.
- f. *Decision Support System*: Memberikan akses untuk data dan alat analisis.
- g. *Expert System*: Meniru keahlian manusia di area tertentu dan membuat keputusan.
- h. *Executive Dashboard*: Menyajikan informasi yang terstruktur dan terangkum mengenai aspek dari kepentingan bisnis kepada para eksekutif.
- i. *Supply Chain Management System*: Mengatur alur berjalannya informasi, barang, dan jasa didalam perusahaan.
- j. *Eletronic Commerce System*: Memungkinkan transaksi antara perusahaan dengan pelanggan.

Menurut Rainer, Prince, dan Cegielski (2015:p282), sistem informasi bertujuan untuk mendukung kegiatan sehari-hari dari setiap area fungsional yang berada dalam perusahaan, yang dijelaskan sebagai berikut:

a. *Accounting*

Pada area *accounting*, sistem informasi membantu dalam melakukan kegiatan *profitability planning* pada tingkat *strategic*, kegiatan *auditing* dan *budgeting* pada tingkat *tactical*, dan kegiatan *payroll*, *accounts payable*, dan *accounts receivable* pada tingkat *operational*.

b. *Finance*

Pada area *finance*, sistem informasi membantu dalam melakukan kegiatan *financial planning* pada tingkat *strategic*, kegiatan *investment management* pada tingkat *tactical*, dan kegiatan *manage cash* dan *manage financial transaction* pada tingkat *operational*.

c. *Human Resources*

Pada area *human resources*, sistem informasi mendukung kegiatan *employment planning* dan *outsourcing* pada tingkat *strategic*, kegiatan *benefits*

administration dan *performance evaluation* pada tingkat *tactical*, dan kegiatan *maintain employee records* pada tingkat *operational*.

d. *Production/Operations*

Pada area *production/operations*, sistem informasi mendukung kegiatan *product life cycle management* pada tingkat *strategic*, kegiatan *quality control* dan *inventory management* pada tingkat *tactical*, dan kegiatan *order fulfillment* dan *processing* pada tingkat *operational*.

e. *Marketing*

Pada area *marketing*, sistem informasi mendukung kegiatan *sales forecasting*, *advertising* dan *planning* pada tingkat *strategic*, kegiatan *customer relations* dan *sales force automation* pada tingkat *tactical*, serta kegiatan *set pricing* dan *profile customers*.

2.1.2. Perancangan Sistem

Menurut Mulyadi (2008:p51), perancangan sistem adalah proses penerjemahan kebutuhan pemakai informasi ke dalam alternatif rancangan sistem informasi yang diajukan kepada pemakai informasi untuk dipertimbangkan. Sedangkan menurut Satzinger, Jackson, dan Burd (2012:p5), perancangan sistem merupakan suatu proses yang menjelaskan secara rinci bagaimana sistem akan bekerja untuk komponen-komponen sistem informasi dan bagaimana bekerjasama untuk memberikan solusi yang dapat diimplentasikan.

2.1.3. Sumber Daya Manusia

Sutrisno (2014:p3) menyatakan bahwa sumber daya manusia adalah satu-satunya sumber daya yang mempunyai perasaan, keterampilan, keinginan, dorongan, pengetahuan, karya, dan daya. Kesemua potensi tersebut berpengaruh terhadap upaya organisasi dalam mencapai tujuan. Sedangkan menurut Gaol (2014:p44), sumber daya manusia adalah orang yang bekerja dan berfungsi sebagai aset organisasi atau perusahaan yang dapat dihitung jumlahnya. Sumber daya manusia juga merupakan penggerak suatu organisasi.

Sunyoto D. (2015:p3) mengatakan bahwa sumber daya manusia adalah dengan keseluruhan penentuan dan pelaksanaan berbagai aktivitas, *policy*, dan program yang bertujuan untuk mendapatkan tenaga kerja, pengembangan, dan

pemeliharaan dalam usaha meningkatkan dukungannya terhadap peningkatan efektivitas organisasi dengan cara yang secara etis dan sosial dapat dipertanggungjawabkan.

2.1.4. Sistem Informasi Sumber Daya Manusia

Menurut Dessler (2003:p15), sistem informasi sumber daya manusia merupakan hasil integrasi dari sistem-sistem sumber daya manusia mereka yang terpisah-pisah pada saat perusahaan sedang berkembang. Sedangkan Laudon (2003:p7) berpendapat bahwa sistem informasi sumber daya manusia adalah komponen-komponen yang berhubungan dan bekerja bersama-sama untuk mengumpulkan, memroses, menyimpan, dan mendistribusikan informasi untuk mendukung dalam pengambilan keputusan, koordinasi, kontrol, analisis, dan visualisasi dari sebuah aktivitas manajemen sumber daya manusia.

2.2. Teori Khusus

Adapun teori khusus yang terdapat dalam penelitian skripsi ini adalah sebagai berikut:

2.2.1 *System Development Life Cycle (SDLC)*

Untuk melakukan pengembangan suatu sistem diperlukan analisis dan kerangka. Menurut Satzinger, Jackson, dan Burd (2012:p38) *System Development Life Cycle (SDLC)* dapat digunakan untuk mengidentifikasi seluruh aktivitas yang dibutuhkan guna mengembangkan sistem-sistem informasi yang telah terbentuk. Aktivitas yang ada pada SDLC mencakup analisis, desain, implementasi, pengujian, dan pengelolaan. Dalam pengembangan sistem informasi terdapat banyak pendekatan atau model pada SDLC yang dapat digunakan, bergantung pada kebutuhannya. Penggunaan SDLC ini akan mengurangi potensi konsekuensi negatif seperti kegagalan dalam pengembangan sistem skala besar yang menelan biaya tinggi, dan sebagainya.

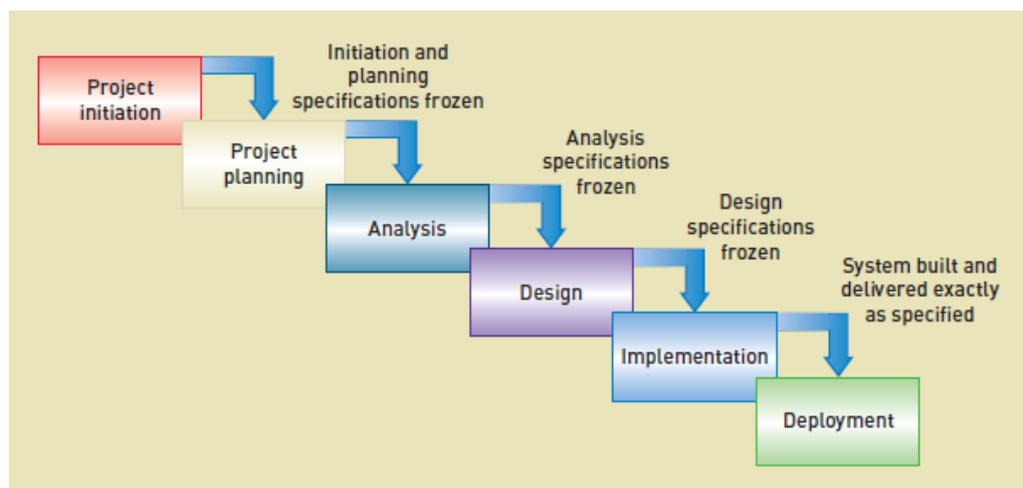
Selaras dengan apa yang dikemukakan oleh Wiley (2016:p300) mengenai pentingnya pengembangan sistem komputer yang berfungsi secara tepat waktu, tergabung dalam jaringan komunikasi, dan koordinasi antar pihak dengan berbagai sudut pandang dan prioritas yang berbeda. Tanpa rencana yang jelas untuk mendefinisikan, mengembangkan, menguji, dan mengimplementasikan

sistem, maka akan mudah sekali berakhir dengan kegagalan sistem dalam pemenuhan tujuan. SDLC dirancang untuk menyediakan rencana ini.

2.2.2 Metode *Waterfall*

Menurut Yurindra (2017:p42), karakteristik konsep *waterfall* merupakan pendekakatan yang dikendalikan oleh dokumen. Metode *waterfall* merupakan model yang membangun perangkat lunak berdasarkan pendekatan pada SDLC yang diibaratkan seperti air terjun yang terus mengalir dari atas sampai ke bawah. Model *waterfall* ini menjelaskan bahwasannya setiap proses saling berkelanjutan dengan proses yang berurutan dan sistematis dari satu proses ke proses berikutnya. Menurut Satzinger, Jackson, dan Burd (2012:p5) menyebutkan model *waterfall* terbagi menjadi enam tahap, yaitu:

1. *Project initiation*, yaitu tahap untuk mengetahui dan mendefinisikan potensi-potensi permasalahan yang terjadi.
2. *Project planning*, yaitu tahap untuk merencanakan sistem yang ingin dibangun sesuai dengan kebutuhan berdasarkan tujuan yang ingin dicapai.
3. *Analysis*, yaitu tahap pengolahan data dan pendefinisian persyaratan.
4. *Design*, yaitu tahap untuk mendesain sistem secara keseluruhan.
5. *Implementation*, merupakan tahap pembangunan sistem.
6. *Deployment*, merupakan tahap pemeliharaan sistem melalui evaluasi dan perbaikan berkala.



Gambar 2.1 SDLC *Waterfall*

Sumber: Satzinger, Jackson, dan Burd (2016:p299)

Disisi lain dalam pengembangan karakteristik model *waterfall* harus mengetahui sisi kelebihan dan kekurangan dalam pengaplikasiannya, diantaranya:

1. Pemrosesan pengerjaan menggunakan metode ini menjadi lebih terstruktur dan teratur dari satu tahap ke tahap selanjutnya.
2. Mengemudahkan sisi user dalam merencanakan dan menyiapkan seluruh kebutuhan data dan proses yang akan diperlukan.
3. Kesesuaian jadwal terhadap proses yang sedang berlangsung sehingga ada kepastian atau kejelasan waktu mengenai target penyelesaian pengembangan program. Dengan adanya urutan yang pasti, dan dapat dilihat pula progress untuk setiap tahap secara pasti.

Namun ada beberapa kelemahan yang perlu diperhatikan dalam penggunaan sistem ini:

1. Sifatnya kaku, sehingga susah dalam melakukan perubahan ditengah proses.
2. Jika terdapat kekurangan proses atau prosedur dari tahun sebelumnya, maka tahapan pengembangan harus dilakukan dari tahap awal, mengakibatkan penggunaan waktu yang cukup lama. Dikarenakan jika proses sebelumnya belum selesai hingga akhir maka tidak bisa melanjutkan ke proses selanjutnya.
3. Membutuhkan data secara terperinci pihak terkait dari tahap awal.

2.2.3 Unified Modeling Language (UML)

Menurut Mulyani (2016:p48) *Unified Modeling Language* (UML) adalah sebuah teknik pengembangan sistem yang menggunakan bahasa grafis sebagai alat untuk pendokumentasian dan melakukan spesifikasi pada sistem. Sedangkan menurut pandangan Rosa dan Shalahuddin (2014:p33) *Unified Modeling Language* (UML) adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek. Adapun jenis-jenis diagram UML adalah sebagai berikut:

1. *Use Case*

Menurut Dennis, Wixom, dan Roth (2012:p148), *use case analysis* adalah sarana untuk mengekspresikan kebutuhan pengguna. Digunakan secara ekstensif dalam tahap analisis dan mewakili bagaimana sistem berinteraksi dengan lingkungan. *Use Case* digunakan untuk mengetahui fungsi apa saja

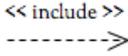
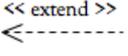
yang ada didalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu.

Dalam analisis *analysis use case*, kita mengidentifikasi suatu entitas yang disebut *actor*, yang berhubungan dengan sistem. Tujuan dari *diagram use case* adalah untuk mengidentifikasi bagaimana sistem akan digunakan oleh *user* Satzinger, Jackson, dan Burd (2012:p78). Berikut ini adalah komponen-komponen dari *use case diagram*:

- a. *System Boundary*, menggambarkan batasan antara sistem (*use case*) dengan *actor*.
- b. *Actor*, seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu.
- c. *Use Case*, menggambarkan perilaku interaksi antara *actor* dengan *system*.
- d. *Communication*, menggambarkan hubungan antara *use case* dengan *actor*.

Menentukan analisa dari kebutuhan sistem dalam melakukan pengembangan sistem atau rancang bangun sistem. Berikut penggambaran analisa mengenai penjelasan dari *use case diagram* diagram ditunjukkan pada tabel 2.1 di bawah ini.

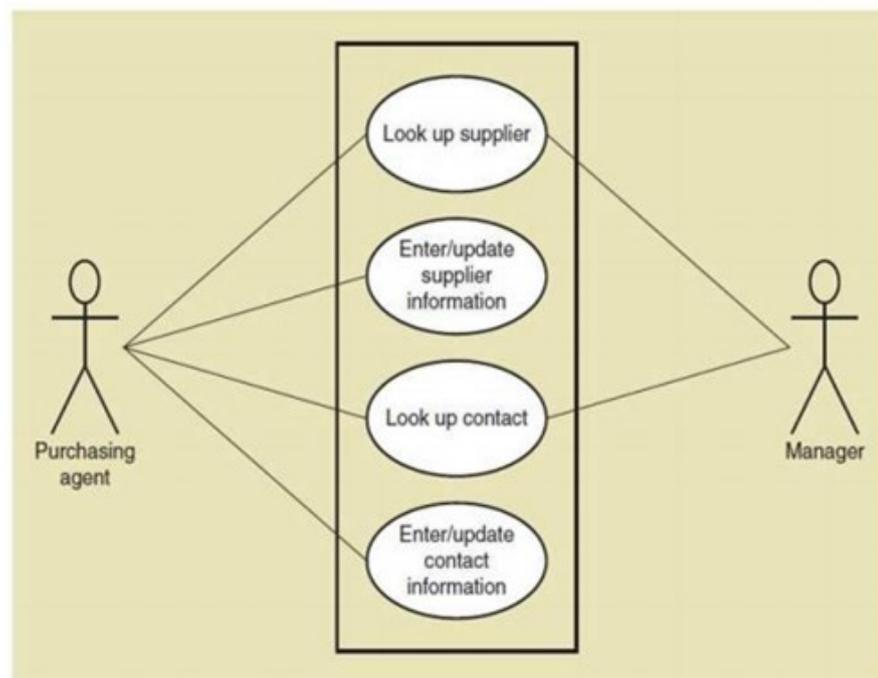
Tabel 2.1 Penjelasan *Use Case Diagram*

No	Gambar	Nama	Keterangan
1.		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i>
2.		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara eksplisit
3.		<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan
4.		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya

- | | | | |
|----|---|-----------------|---|
| 5. |  | <i>System</i> | Menspesifikasikan paket yang menampilkan sistem secara terbatas |
| 6. |  | <i>Use Case</i> | Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor |
-

Sumber: <https://widuri.raharja.info> (2015)

Use case analysis ini juga meliputi *use case diagram*, *use case description*, dan *use case fully developed description*. *Use case diagram* adalah diagram yang menunjukkan beragam peran user dan bagaimana peran tersebut di dalam sistem. Contoh *use case diagram* ditunjukkan pada gambar 2.2 di bawah ini.



Gambar 2.2 Contoh *Use Case Diagram*
 Sumber: Satzinger, Jackson, dan Burd (2012:p31)

Masih merupakan pendapat dari Satzinger, Jackson, dan Burd (2012:p121), *use case description* adalah sebuah model tekstual yang berisi daftar dan menguraikan mengenai rincian proses untuk *use case*. Terdapat tiga

tipe *use case description*, yaitu *brief description*, *intermediate description*, dan *fully developed use case description*.

Fully developed use case description adalah metode paling formal untuk mendokumentasikan sebuah *use case*. Dalam *fully developed use case description* berisi tentang;

1. *Use case name*, yang berguna untuk menuliskan nama dari suatu *use case*.
2. *Scenario*, yang berguna untuk menuliskan aktivitas dalam *use case*.
3. *Triggering event*, yang berguna untuk mengidentifikasi pemicu yang memulai suatu *use case*.
4. *Brief description*, yang berguna untuk menuliskan deskripsi dari suatu *use case*/skenario.
5. *Actors*, yang berguna untuk mengidentifikasi aktor yang berhubungan langsung dengan *use case*.
6. *Related use case*, yang berguna untuk mengidentifikasi *use case* lain dan cara *use case* lain tersebut berhubungan dengan suatu *use case*.
7. *Stakeholders*, yang berguna untuk mengidentifikasi kelompok/aktor yang menggunakan hasil dari *use case*.
8. *Precondition*, yang berguna untuk mengidentifikasi suatu tindakan yang harus dilakukan sebelum *use case* dimulai, termasuk objek yang sudah harus ada sebelum dimulainya suatu *use case*.
9. *Postcondition*, yang berguna untuk mengidentifikasi suatu tindakan yang terjadi setelah selesainya suatu *use case*.
10. *Flow of Events*, yang dibagi dalam dua kolom, yaitu kolom aktor dan kolom sistem. Kolom aktor untuk mengidentifikasi tindakan-tindakan yang dilakukan oleh aktor secara berurutan, sedangkan kolom sistem untuk mengidentifikasi respon yang diberikan oleh sistem.
11. *Exception conditions*, yang berguna untuk mengidentifikasi kondisi yang menjadi pengecualian terjadinya suatu tindakan dalam *use case*.

Mengingat bahwa *use case fully developed description* guna menjelaskan *use cases*, maka jumlah *use case fully developed description* yang dibangun pada dasarnya sama dengan jumlah keseluruhan *use case*. Berikut adalah contoh *use case fully developed description* yang ditunjukkan pada gambar 2.3 di bawah ini:

Use case name:	Create customer account.	
Scenario:	Create online customer account.	
Triggering event:	New customer wants to set up account online.	
Brief description:	Online customer creates customer account by entering basic information and then following up with one or more addresses and a credit or debit card.	
Actors:	Customer.	
Related use cases:	Might be invoked by the <i>Check out shopping cart</i> use case.	
Stakeholders:	Accounting, Marketing, Sales.	
Preconditions:	Customer account subsystem must be available. Credit/debit authorization services must be available.	
Postconditions:	Customer must be created and saved. One or more Addresses must be created and saved. Credit/debit card information must be validated. Account must be created and saved. Address and Account must be associated with Customer.	
Flow of activities:	Actor	System
	1. Customer indicates desire to create customer account and enters basic customer information.	1.1 System creates a new customer. 1.2 System prompts for customer addresses.
	2. Customer enters one or more addresses.	2.1 System creates addresses. 2.2 System prompts for credit/debit card.
	3. Customer enters credit/debit card information.	3.1 System creates account. 3.2 System verifies authorization for credit/debit card. 3.3 System associates customer, address, and account. 3.4 System returns valid customer account details.
Exception conditions:	1.1 Basic customer data are incomplete. 2.1 The address isn't valid. 3.2 Credit/debit information isn't valid.	

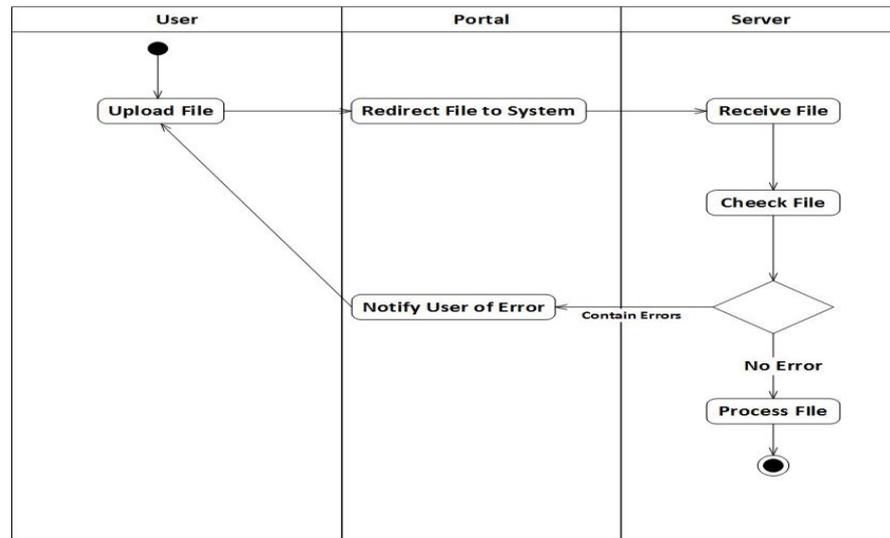
Gambar 2.3 Contoh *Use Case Fully Developed Description*
Sumber: Satzinger, Jackson, dan Burd (2012:p123)

2. Activity Diagram

Menurut Dennis, Wixom, dan Tegarden (2015:p130), *activity diagram* digunakan untuk memodelkan perilaku dari objek proses bisnis, alur kerja dari banyak kasus untuk mendetailkan individual *use case*, hingga individual *method*. Tujuan dari model *use case* yakni untuk mengidentifikasi dan mendefinisikan semua proses bisnis dasar yang harus didukung oleh sistem. Analisis mendefinisikan penggunaan permasalahan pada dua tingkatan tinjauan umum dan tinjauan terperinci. *Activity diagram* akan menyediakan gambaran umum dari semua aktivitas suatu sistem.

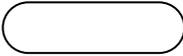
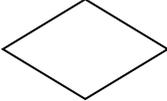
Jika ditarik sebuah garis kesimpulan, maka *activity diagram* dapat digunakan untuk memodelkan tipe proses apapun. Dalam penggambarannya, *activity diagram* adalah jenis diagram alur kerja yang menggambarkan kegiatan pengguna atau sistem dan aliran sekuensial kegiatan mereka. Bentuk oval dalam *activity diagram* menunjukkan kegiatan yang dilakukan dalam transaksi. Tanda panah menunjukkan hubungan kegiatan secara berurutan. Lingkaran hitam menjelaskan awal kegiatan hingga akhir kegiatan. Simbol *diamond* menjelaskan poin aliran proses dari hasil eksekusi (Satzinger, Jackson, dan

Burd, 2012:p57). Contoh dari *activity diagram* ditunjukkan oleh gambar 2.4 dan penjelasannya ditunjukkan oleh tabel 2.2.



Gambar 2.4 Contoh *Activity Diagram*
Sumber: Satzinger, Jackson, dan Burd (2012:p59)

Tabel 2.2 Penjelasan *Activity Diagram*

No.	Simbol	Deskripsi
1.		<i>Rounded Rectangle</i> sebagai bentuk untuk fungsi sistem yang spesifik.
2.		<i>Arrows/Tanda panah</i> berfungsi untuk menunjukkan aliran kerja pada sistem.
3.		<i>Decision Diamonds</i> sebagai bentuk kerja yang memiliki cabang, bisa memiliki lebih dari 1 bentuk kerja selanjutnya.
4.		<i>Horizontal Lines</i> untuk menjelaskan aktifitas paralel yang sedang terjadi.
5.		<i>Start node</i> berfungsi untuk menunjukkan awal dari proses yang akan berjalan.
6.		<i>End node</i> berfungsi untuk menunjukkan akhir dari suatu proses yang berjalan.

Sumber: Satzinger, Jackson, dan Burd (2012:p59)

3. *Class Diagram*

Menurut Bentley, Lonnie, dan Jeffrey (2007:p382), *class diagram* adalah sebuah diagram menggambarkan struktur objek dari sistem yang ada, dimana

class diagram ini memperlihatkan *object class* yang menyusun diagram ini beserta hubungan antara *object class* tersebut. Sedangkan menurut Satzinger, Jackson, dan Burd (2012:p101), *class diagram* atau juga disebut *domain class diagram* adalah sebuah diagram UML yang merepresentasikan kelas-kelas domain, atribut, pekerjaan pengguna serta hubungan antar kelas tersebut. Pada *class diagram*, bentuk kotak menggambarkan *classes* dan garis menunjukkan hubungan antar *class* tersebut. *Domain class diagram* digunakan untuk memahami hubungan antar *class* yang terdiri dari beberapa objek di dalam pengembangan dan perancangan sistem nantinya.

Dengan demikian, *class diagram* adalah diagram yang menggambarkan struktur objek dari sebuah sistem. *Class diagram* sistem yang bersifat terstruktur. Dimana pemikiran mengenai pembangunan sistem akan dicapai oleh objek yang dipakai dari beberapa unsur kelas. Dikarenakan unsur tersebut dapat membantu untuk memiliki beberapa gagasan tentang jumlah objek yang akan berkolaborasi satu sama lain dalam hubungan antar *class*.

4. *Sequence Diagram*

Menurut Satzinger, Jackson, dan Burd (2005:p316), *sequence diagram* merupakan sebuah diagram yang menunjukkan eksekusi *operation* di sebuah objek yang melibatkan pemanggilan *operations* di objek lain. *Sequence diagram* juga merupakan diagram yang menunjukkan urutan pesan antara aktor eksternal dan internal sistem di dalam *use case* atau *scenario* yang sudah dirancang sebelumnya (Satzinger, Jackson, dan Burd, 2012:p126). Jika ditarik garis besar *sequence diagram* mendeskripsikan bagaimana entitas dalam system berinteraksi, termasuk pesan yang digunakan dalam interaksi. Semua yang terdeskripsikan ada pada urutan eksekusi. Dikarenakan *sequence diagram* berhubungan erat dengan *use case diagram*. Penjelasan mengenai urutan eksekusi tersebut diantaranya:

1. Interaksi yang terjadi dalam kolaborasi dalam membangun *use case* atau operasi (diagram contoh atau diagram generic).
2. Interaksi tingkat tinggi antara pengguna sistem dan sistem itu sendiri, antara sistem satu dengan sistem lainnya, atau diantara subsistem.

Perwujudan *sequence diagram* pada dasarnya sebagai model interaksi tingkat tinggi antara objek aktif dalam suatu sistem. Selain itu sebagai kolaborasi yang menciptakan *use case* dan operasional sistem, baik dari model

interaksi generik dengan menunjukkan semua jalur yang mungkin melalui interaksi atau spesifikasi dari interaksi dengan menunjukkan data jalur interaksi.

2.2.4 Android

Menurut Safaat (2015:p1), android adalah sebuah sistem operasi untuk perangkat mobile berbasis *linux* yang mencakup sistem operasi, middleware dan aplikasi. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka. Android dipuji sebagai “*platform*” mobile pertama yang Lengkap, Terbuka, dan Bebas :

1. Lengkap (*Complete Platform*): Para desainer dapat melakukan pendekatan yang komprehensif ketika mereka sedang mengembangkan *platform Android*. Android merupakan sistem operasi yang aman dan banyak menyediakan *tools* dalam membangun software dan memungkinkan untuk peluang pengembangan aplikasi.
2. Terbuka (*Open Source Platform*): *Platform Android* disediakan melalui lisensi *open source*. Pengembang dapat dengan bebas untuk mengembangkan aplikasi. Android sendiri menggunakan *Linux Kernel 2.6*.
3. Bebas (*Free Platform*): Android adalah *platform / aplikasi* yang bebas untuk develop. Tidak ada lisensi atau biaya *royalty* untuk dikembangkan pada *platform Android*. Tidak ada biaya keanggotaan diperlukan. Aplikasi untuk Android dapat didistribusikan dan diperdagangkan dalam bentuk apapun.

Sistem operasi yang kuat dan mendukung dalam android membuat sejumlah besar digunakan sebagai aplikasi di *Smartphone*. Aplikasi ini lebih nyaman dan canggih bagi pengguna *Smartphone*. Perangkat keras yang mendukung perangkat lunak android didasarkan pada *platform* arsitektur *Acorn RISC Machine (ARM)*. Dengan begitu android memuat jutaan aplikasi yang tersedia guna membantu pengguna dalam mengelola aktivitas sesuai kebutuhannya baik dengan satu cara atau lain cara.

Android menggunakan *kernel Linux* dalam mendukung berbagai *driver* perangkat keras. *Kernel* sendiri merupakan jantung dari sistem operasi yang mengelola permintaan input dan *output* dari perangkat lunak. Hal ini sebagai

fungsionalitas sistem dasar seperti adanya manajemen proses, dan manajemen memori, manajemen perangkat.

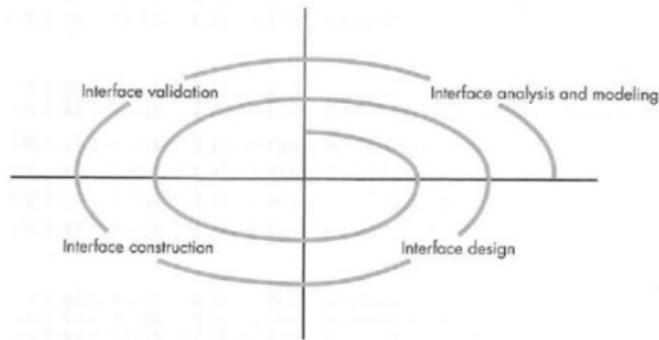
Menurut Sifaat (2015:p6), Arsitektur Android dapat digambarkan sebagai berikut :

1. *Applications dan Widgets*: adalah *layer* dimana kita berhubungan dengan aplikasi saja.
2. *Applications Frameworks*: adalah *layer* dimana para pembuat aplikasi melakukan pengembangan/pembuatan aplikasi yang akan dijalankan di *system* operasi Android
3. *Libraries*: adalah *layer* dimana fitur-fitur Android berada
4. *Android Run Time*
5. *Linux Kernel*: adalah *layer* dimana inti dari *operating system* dari Android itu berada. Berisi *file-file* sistem yang mengatur sistem *processing, memory, resource, drivers*, dan sistem-sistem operasi Android lainnya.

2.2.5 Use Interface

Menurut Satzinger, Jackson, dan Burd (2012:p189), *User Interface* adalah *input* dan *output* yang secara langsung melibatkan *user* sistem. *User Interface* bisa digunakan oleh pihak internal dan eksternal. Desain *user Interface* bisa bervariasi tergantung pada faktor-faktor seperti tujuan penggunaan *interface*, karakteristik *user*, dan karakteristik dari *device* untuk menampilkan *interface*.

Antar muka (*User Interface*) sangat diperlukan dalam suatu pembuatan sistem. Dengan adanya antar muka, diharapkan pengguna akan mudah memahami apa yang harus dilakukan dan menghindari terjadinya kebingungan saat menggunakan sistem. Pressman (2010:p312), Desain *user interface* adalah menciptakan sebuah media komunikasi yang efektif antara manusia dengan komputer.



Gambar 2.5 Proses *User Interface*
 Sumber: Pressman (2010:p319)

Terdapat langkah-langkah dalam pembuatan antar muka (*User Interface*):

1. *Interface analysis and modeling*
Interface analysis berfokus pada *user* untuk berinteraksi dengan sistem. Kemudian dianalisis untuk mendefinisikan satu set objek dan aksi interface. Informasi yang dikumpulkan digunakan untuk membuat model analisis untuk *interface*.
2. *Interface design*
Interface design mendefinisikan satu set obyek dan aksi *interface* yang memungkinkan *user* untuk melakukan semua tugas desain tata letak.
3. *Interface construction*
Interface construction biasanya dimulai dengan membuat sebuah *prototype* yang memungkinkan skenario penggunaan untuk dievaluasi dan digunakan untuk menyelesaikan konstruksi *interface*.
4. *Interface validation*
 Setelah pembuatan *prototype*, *interface validation* berfokus pada evaluasi secara keseluruhan untuk menentukan kemampuan *interface* untuk menjalankan setiap perintah berjalan dengan benar dan sejauh mana *interface* mudah digunakan dan mudah dipelajari serta memenuhi kebutuhan *user*.

2.2.6 Metode Pengumpulan Data

Teknik pengumpulan data merupakan langkah yang paling strategis dalam sebuah penelitian, sebab tujuan utama dari penelitian adalah mendapatkan data yang akurat, sehingga tanpa mengetahui teknik pengumpulan data peneliti tidak akan mendapatkan data yang memenuhi

standar yang ditetapkan (Sugiyono, 2016:p224). Teknik pengumpulan data yang digunakan dalam penelitian ini adalah dengan menggunakan metode *literature review* dan studi lapangan. Adapun teknik studi lapangan yang digunakan adalah wawancara, dan observasi.

Menurut sumber yang diambil dari presentasi Agusta (2007) mengenai metode penelitian, "*literature review is a critical analysis of the research conducted on a particular topic or question in the field of science*", yang artinya bahwa *literature review* merupakan analisis kritis dari penelitian yang sedang dilakukan terhadap topik khusus atau berupa pertanyaan terhadap suatu bagian dari keilmuan.

Adapun penjelasan mengenai studi lapangan adalah sebagai berikut:

1. Wawancara

Wawancara merupakan percakapan dengan maksud tertentu, yang dilakukan oleh dua pihak yaitu pewawancara (*interviewer*) yang mengajukan pertanyaan dan yang diwawancarai (*interviewee*) yang memberikan jawaban atas pertanyaan yang telah diberikan (Moleong, 2014:p186). Peneliti dalam penelitian ini menggunakan metode wawancara semi terstruktur yaitu wawancara yang dalam pelaksanaannya lebih bebas dibandingkan dengan wawancara terstruktur. Pewawancara memberikan pertanyaan kepada informan namun dapat berkembang dan lebih bebas sesuai dengan situasi dan informasi yang dibutuhkan oleh informan. Wawancara semi terstruktur bertujuan untuk menemukan permasalahan secara lebih terbuka, dimana pihak yang diwawancarai dimintai pendapat dan ide-idenya (Sugiyono, 2016:p233).

2. Observasi

Observasi adalah suatu metode atau cara untuk menganalisis dan melakukan pencatatan yang dilakukan secara sistematis, tidak hanya terbatas dari orang, tetapi juga obyek-obyek alam yang lain (Sugiyono:2010). Pada penelitian kualitatif teknik pengumpulan data dengan menggunakan metode observasi sangat dibutuhkan. Moleong (2014:p174) menyatakan salah satu alasan penggunaan metode observasi dalam penelitian kualitatif adalah memungkinkan melihat dan mengamati sendiri fenomena yang terjadi pada saat penelitian, kemudian mencatat perilaku dan kejadian sebagaimana yang terjadi pada keadaan sebenarnya.

2.3 Literature Review

Banyak penelitian yang telah dilakukan mengenai perancangan sistem informasi perizinan. Dalam upaya mengembangkan dan menyempurnakan perancangan sistem informasi sumber daya manusia perizinan tidak masuk kerja ini, perlu dilakukan *literature review* sebagai salah satu penerapan metode pengumpulan data. Adapun hasil dari *literature review* yang telah dilakukan adalah sebagai berikut:

Tabel 2.3 Tinjauan *Literatur Review*

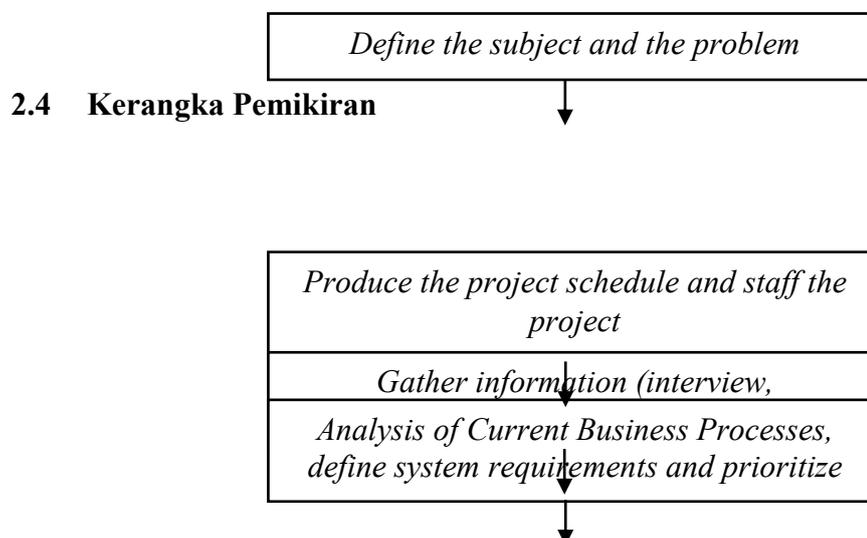
No	Penulis	Judul	Tahun Terbit	Penerbit	Abstrak	Sumber
1.	Lunario Rizal Halim, Youtaro Marshall Matsuda, Raffel Hindrali-jaya.	Pengembangan Aplikasi Surat Izin untuk Karyawan pada PT. Transcosmos Indonesia.	2017	Jurusan Teknik Informatika Universitas Bina Nusantara, Jakarta.	Laporan skripsi ini menjelaskan tentang metode analisis sistem berjalan menggunakan diagram <i>activity</i> beserta deskripsinya serta metode perancangan sistem yang jelas sehingga mudah dipelajari.	http://library.binus.ac.id/Collections/ethesis_detail/2017-1-01013
		(Skripsi)				
2.	Listina Nadhia Ningsih, Abert Tandilintin, dan Izza Ilma Insyifani.	Prototype Aplikasi Mobile Internship Management Application System (IMAS) pada PT. Garuda Maintenance Facility (GMF) AeroAsia.	2016	Perguruan Tinggi Raharja, Tangerang.	Jurnal ini menjelaskan petunjuk penulisan karya ilmiah yang baku dan sesuai dengan kaidah ketatabahasaan atau ejaan yang dibenarkan.	Jurnal <i>Creative Communication and Innovative Technology (CCIT)</i> Volume 2 - Nomor 2 - Agustus 2016.
		(Jurnal)				

3.	Fransiskus Adikara	Pengembangan Fungsi Pengajuan Cuti Karyawan pada Sistem Absensi <i>Mobile</i>	2016	Institut Teknologi Sepuluh Nopember (ITS), Surabaya.	Jurnal ini menjelaskan bahwa <i>smartphone</i> berbasis android dapat digunakan untuk mengajukan cuti karyawan secara mandiri.	Jurnal Sisfo Volume 06 Nomor 01 (2016) 78-86.
----	--------------------	---	------	--	--	---

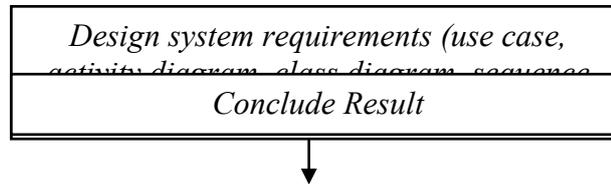
(Jurnal)

Literature review ini dilakukan dengan memulai studi di bidang terkait yang paling akhir dimuat dalam terbitan-terbitan terbaru, selanjutnya mundur ke terbitan-terbitan sebelumnya. Tujuan dilakukannya hal ini untuk memperoleh landasan teori yang dapat mengantarkan peneliti kepada pemecahan masalah dalam penelitian. Teori tersebut adalah langkah pertama sehingga permasalahan yang ada dapat dipahami dengan baik berdasarkan kerangka berpikir ilmiah. Selain itu, dengan melakukan *literature review*, didapat informasi berkenaan dengan penelitian sebelumnya.

Berdasarkan sumber-sumber *literature review* yang ada, pemahaman tentang perancangan sistem informasi perizinan berbasis *mobile* meningkat sehingga peneliti mampu menyusun kerangka konseptual dalam penelitian. Penguasaan teori mengenai perancangan sistem informasi perizinan berbasis *mobile* tersebut juga meningkatkan motivasi dan keyakinan peneliti sehingga dapat sampai kepada hasil penelitian yang direncanakan.



Gambar 2.6 Bagan Kerangka Pemikiran



Adapun penjelasan dari bagan kerangka pemikiran di atas, bahwasanya penelitian dilakukan dengan terlebih dahulu mendefinisikan subjek penelitian, kemudian mendefinisikan masalah yang akan menjadi pokok bahasan penelitian. Setelah mengidentifikasi masalah, peneliti melakukan perencanaan berupa penyusunan *project schedule* dan pembagian tugas.

Setelah melakukan perencanaan *project*, peneliti melakukan pengumpulan data dengan cara wawancara dengan calon *user*, observasi, dan studi pustaka terhadap semua sumber pembahasan yang berhubungan dengan permasalahan utama sehingga dapat dianalisis bagaimana proses bisnis sistem yang sedang berjalan untuk kemudian dapat ditentukan kebutuhan sistem dan prioritasnya.

Selanjutnya, hasil analisis dituangkan ke dalam suatu rancangan sistem menggunakan diagram *Unified Modeling Language* (UML) berupa diagram *use case*, *activity diagram*, *class diagram* dan *sequence diagram*. Adapun *user interface system* dirancang menggunakan *sitemap* dan *mockup*. Keseluruhan hasil dari penelitian lalu dijelaskan secara ringkas melalui bagian kesimpulan, sedangkan apa yang masih perlu diperbaiki atau apa yang masih bisa dikembangkan lagi diurai pada bagian saran.