

BAB II

LANDASAN TEORI

2.1. Konsep Dasar

Umumnya setiap organisasi atau sebuah instansi mempunyai sebuah alur sistem informasi dalam mengumpulkan, menyimpan, melihat, mengolah, dan menyalurkan informasi yang berjalan secara teratur terus menerus sesuai dengan alurnya.

Sistem pada dasarnya adalah kelompok berbagai unsur-unsur yang erat kaitannya antara satu dengan yang lainnya, yang berfungsi bersama-sama untuk mencapai suatu tujuan tertentu. Unsur-unsur yang mewakili suatu sistem secara umum adalah masukan (*input*), pengolahan (*process*), dan keluaran (*output*).

2.1.1. Pengertian Sistem

“Sistem adalah kumpulan orang yang saling bekerja sama dengan ketentuan-ketentuan aturan yang sistematis dan terstruktur untuk membentuk satu kesatuan yang melaksanakan suatu fungsi untuk mencapai suatu tujuan” (Anggraeni dan Irviani, 2017). Romney mengatakan sistem adalah “kumpulan dari dua atau lebih komponen yang saling bekerja dan berhubungan untuk mencapai tujuan tertentu” (Mulyani, 2018).

Kamus Teknologi dan Informasi mempunyai pendapat lain tentang sistem, yaitu sistem adalah “suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu” (Mulyani, 2018).

Berdasarkan pengertian di atas dapat disimpulkan bahwa sistem merupakan suatu kumpulan dari berbagai komponen dan elemen yang memiliki hubungan dan bekerja sama untuk mencapai tujuan dalam suatu kegiatan.

2.1.2. Karakteristik Sistem

Untuk mengetahui sesuatu itu adalah sebuah sistem atau bukan dapat dilihat dari beberapa ciri khusus yang menyertainya. Berikut ciri khusus dari sistem agar dapat dibedakan (Nafiudin, 2019):

1. Komponen sistem

Sistem terdiri dari sejumlah komponen yang saling berinteraksi atau bekerja sama membentuk satu kesatuan. Komponen = elemen = bagian = sub sistem.

2. Batas sistem (*Boundary*)

Daerah yang membatasi antara satu sistem dengan sistem lainnya atau dengan lingkungan luarnya. Batas sistem memungkinkan dipandang sebagai satu kesatuan serta menunjukkan *scope* sistem.

3. Lingkungan Luar Sistem (*Environment*)

Lingkungan adalah apa pun di luar batas sistem yang mempengaruhi operasi sistem. Lingkungan ini dapat bersifat menguntungkan/merugikan.

4. Penghubung Sistem (*Interface*)

Penghubung sistem merupakan medio penghubung antara satu sub sistem dengan sub sistem lainnya.

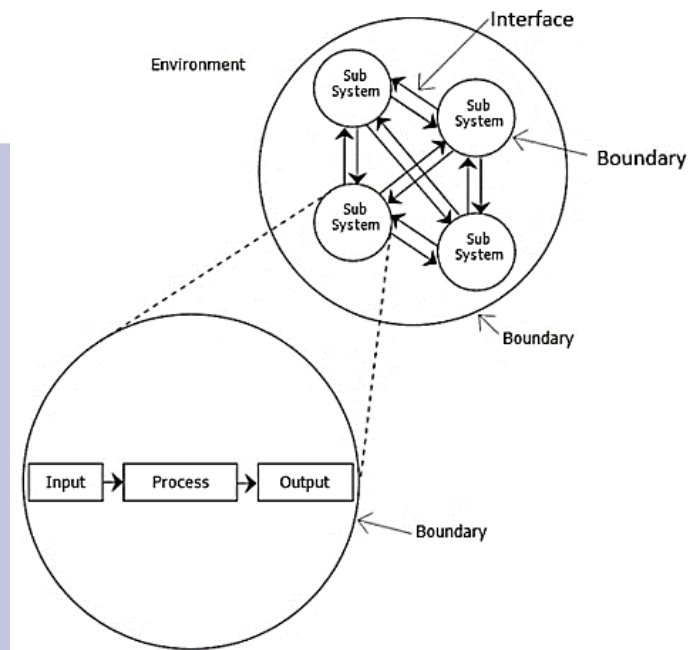
5. Masukan Sistem (*Input*)

Masukan adalah energi yang dimasukkan ke dalam sistem. Masukan ini dapat berupa *maintenance input* dan *signal input*.

6. Keluaran Sistem (*Output*)

Keluaran sistem adalah energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan. Keluaran sistem dapat menjadi *input* bagi sub sistem lainnya atau kepada supra sistem.

Gambaran karakteristik sistem:



Sumber : (Nafiudin, 2019)

Gambar II.1.
Karakteristik Sistem

2.1.3. Klasifikasi Sistem

Sistem dapat diklasifikasikan dalam beberapa sudut pandang, di antaranya sebagai berikut (Hutahaeen, 2015):

1. Sistem Abstrak (*Abstract System*)

Sistem Abstrak adalah sistem yang berupa pemikiran-pemikiran atau ide-ide yang tidak tampak secara fisik.

2. Sistem Fisik (*Physical System*)

Sistem fisik adalah sistem yang ada secara fisik.

3. Sistem Alamiah (*Natural System*)

Sistem alamiah adalah sistem yang terjadi melalui proses alam, tidak dibuat oleh manusia. Misalnya sistem perputaran bumi.

4. Sistem Buatan Manusia (*Human Made System*)

Sistem buatan manusia adalah sistem yang dibuat oleh manusia yang melibatkan interaksi antara manusia dengan mesin (*Human machine system*)

5. Sistem Tertentu

Sistem tertentu adalah sistem yang beroperasi dengan tingkah laku yang sudah dapat diprediksi, sebagai keluaran sistem yang dapat diramalkan.

6. Sistem Tak Tentu (*Probalistic System*)

Sistem tak tentu adalah sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probabilitas.

7. Sistem Tertutup (*Close System*)

Sistem tertutup adalah sistem yang tidak terpengaruh dan tidak berhubungan dengan lingkungan luar, sistem bekerja otomatis tanpa ada turut campur lingkungan luar. Secara teoritis sistem tertutup ini ada, kenyataannya tidak ada sistem yang benar-benar tertutup, yang ada hanya *relatively closed system*.

8. Sistem Terbuka (*Open System*)

Sistem terbuka adalah sistem yang berhubungan dan terpengaruh dengan lingkungan luarnya. Sistem ini menerima input dan output dari lingkungan luar atau subsistem lainnya. Karena sistem terbuka terpengaruh lingkungan luar maka harus mempunyai pengendali yang baik”.

2.1.4. Pengertian Informasi

Menurut (Nafiudin, 2019) informasi dapat diartikan sebagai hasil pengolahan data yang berarti dan bermanfaat. Informasi juga dapat didefinisikan sebagai “data yang telah diorganisir dan diproses sehingga bermanfaat bagi proses pengambilan keputusan. Semakin banyak dan semakin berkualitas informasi yang tersedia, maka pengambilan keputusan menjadi semakin baik” (TM Books, 2017). Informasi diperoleh dari sekumpulan data yang diolah dan menghasilkan keluaran yang bermanfaat.

Nilai informasi (*Value Of Information*) ditentukan dari dua hal, yaitu manfaat dan biaya mendapatkannya. Suatu informasi disebut bernilai jika manfaatnya lebih efektif dibandingkan dengan biaya mendapatkannya.

Kualitas informasi tergantung dari 3 (tiga) hal, yaitu:

1. Akurat (*Accurate*)

Informasi harus bebas dari kesalahan dan tidak bias atau menyesatkan, dimana informasi harus jelas dan mencerminkan maksudnya.

2. Tepat Waktu (*Timelines*)

Informasi yang datang kepada penerima tidak boleh terlambat, sebab informasi merupakan landasan dalam pengambilan keputusan. Jika informasi terlambat akan berakibat fatal bagi suatu organisasi.

3. Relevan (*Relevance*)

Informasi yang diberikan harus sesuai dengan apa yang dibutuhkan. kegunaan informasi untuk mengurangi hal ketidakpastian di dalam proses pengambilan keputusan tentang suatu fakta.

2.1.5. Pengertian Sistem Informasi

Azhar Susanto memberi definisi sistem informasi sebagai “kumpulan dari sub sistem apa pun baik fisik ataupun non fisik yang saling berhubungan satu sama lain dan bekerja sama secara harmonis untuk mencapai satu tujuan yaitu mengolah data menjadi informasi yang berarti dan berguna” (Nafiudin, 2019). Pendapat lain dari Laudon tentang sistem informasi yaitu “komponen-komponen yang saling berhubungan dan bekerja sama untuk mengumpulkan, memproses, menyimpan, dan mendistribusikan informasi untuk mendukung proses pengambilan keputusan, koordinasi, pengendalian, dan untuk memberikan gambaran aktivitas di dalam perusahaan” (Nafiudin, 2019)”.


Sistem informasi juga dapat diartikan “kumpulan dari elemen- elemen yang berinteraksi untuk menghasilkan informasi yang berguna bagi penggunanya” (Nafiudin, 2019).

Dari pengertian di atas dapat disimpulkan bahwa sistem informasi merupakan suatu grup elemen-elemen fisik maupun bukan fisik yang menunjukkan suatu kumpulan saling berhubungan dan berinteraksi sehingga menghasilkan keluaran berupa informasi yang bermanfaat.

2.1.6. Pengertian Akuntansi

Akuntansi berasal dari bahasa inggris yaitu “*to account*” yang artinya menghitung atau mempertanggungjawabkan sesuatu yang ada kaitannya dengan pengelolaan bidang keuangan dari suatu perusahaan kepada pemiliknya atas kepercayaan yang telah diberikan kepada pengelola tersebut untuk menjalankan kegiatan perusahaan (Sujarweni, 2016).

Ikatan Akuntan Amerika menyatakan bahwa akuntansi “adalah suatu potongan, estimasi, dan pelaporan data keuangan, yang memungkinkan sebuah penilaian dan pilihan membuat yang jelas dan tegas bagi individu yang

memanfaatkan data” (Bitar, 2020). Definisi lain yang dipaparkan Soemarsono memberi batasan bahwa akuntansi adalah “proses mengidentifikasi, mengukur dan melaporkan informasi ekonomi untuk memungkinkan adanya penilaian dan keputusan yang jelas dan tegas bagi mereka yang menggunakan informasi tersebut” (Bitar, 2020).

Dapat ditarik kesimpulan bahwa akuntansi merupakan sebuah proses pengidentifikasian, pengukuran, pemisahan atau klasifikasi serta pelaporan yang berkaitan dengan data keuangan dengan cara terstruktur dan terperinci.

2.1.7. Pengertian Sistem Informasi Akuntansi

Sistem akuntansi adalah “organisasi formulir, catatan, dan laporan yang dikoordinasi sedemikian rupa untuk menyediakan informasi keuangan yang dibutuhkan oleh manajemen guna memudahkan pengelolaan perusahaan” (Mulyadi, 2016). Sistem Informasi Akuntansi merupakan “jaringan dari seluruh prosedur, formulir-formulir, catatan-catatan, dan alat-alat yang digunakan untuk mengolah data keuangan menjadi suatu bentuk laporan yang akan digunakan oleh pihak manajemen dalam mengendalikan kegiatan usahanya dan selanjutnya digunakan sebagai alat pengambilan keputusan manajemen” (Marina et al., 2017).

Tujuan pokok dari diselenggarakannya SIA antara lain (Marina et al., 2017):

1. Mengumpulkan dan menyimpan data tentang aktivitas dan kegiatan keuangan perusahaan.
2. Memproses data menjadi informasi yang dapat digunakan dalam proses pengambilan keputusan perusahaan.
3. Melakukan pengendalian terhadap seluruh aspek perusahaan.

SIA bertanggung jawab untuk memberikan informasi keuangan dan informasi lain yang dibutuhkan dengan cara mengumpulkan dan mengolah data transaksi kemudian di disampaikan kepada pemakai. SIA dapat dipakai oleh perusahaan yang telah terkomputerisasi maupun perusahaan yang masih bersifat manual dalam pengolahan data akuntansinya.

2.1.8. Pengertian Penjualan Kredit

Soemarsono menjelaskan bahwa penjualan kredit adalah “transaksi antara perusahaan dengan pembeli untuk menyerahkan barang atau jasa yang berakibat timbulnya piutang, kas aktiva (Hidayah et al., 2016), definisi lain dipaparkan oleh Kieso yang menerangkan penjualan kredit adalah “janji lisan dari pembeli untuk membayar jasa yang dijual” (Hidayah et al., 2016).

Berdasarkan penjelasan di atas, dapat ditarik kesimpulan penjualan kredit yaitu transaksi penjualan yang dilakukan tetapi pembayarannya menggunakan jangka waktu tertentu sesuai kesepakatan bersama.

2.2. Peralatan Pendukung (*Tools System*)

Peralatan pendukung (*tools system*) merupakan alat yang digunakan untuk menggambarkan bentuk logika model dari suatu sistem dengan menggunakan simbol-simbol, lambang-lambang, diagram-diagram yang menunjukkan secara tepat arti dan fungsinya

Dalam proses rancang bangun sistem diperlukan adanya peralatan yang mendukung untuk menentukan jenis data yang akan menjadi masukan (*input*) dan keluaran (*output*) didalam suatu pembuatan program.

2.2.1. Pengertian *Unified Modelling Language* (UML)

Fowler mendefinisikan UML (*Unified Modeling Language*) sebagai “Keluarga notasi grafis yang didukung oleh meta-model tunggal, yang membantu pendeskripsian dan desain sistem perangkat lunak, khususnya sistem yang dibangun menggunakan pemrograman berorientasi objek (OO)” (Fridayanthie & Mahdiati, 2016). UML merupakan salah standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan kebutuhan, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek (Sukamto & Shalahuddin, 2018).

Dapat disimpulkan bahwa UML adalah sebuah permodelan yang dijadikan standard yang digunakan untuk kepentingan pembuatan sistem, terutama sistem dengan bahasa pemrograman berbasis obyek.

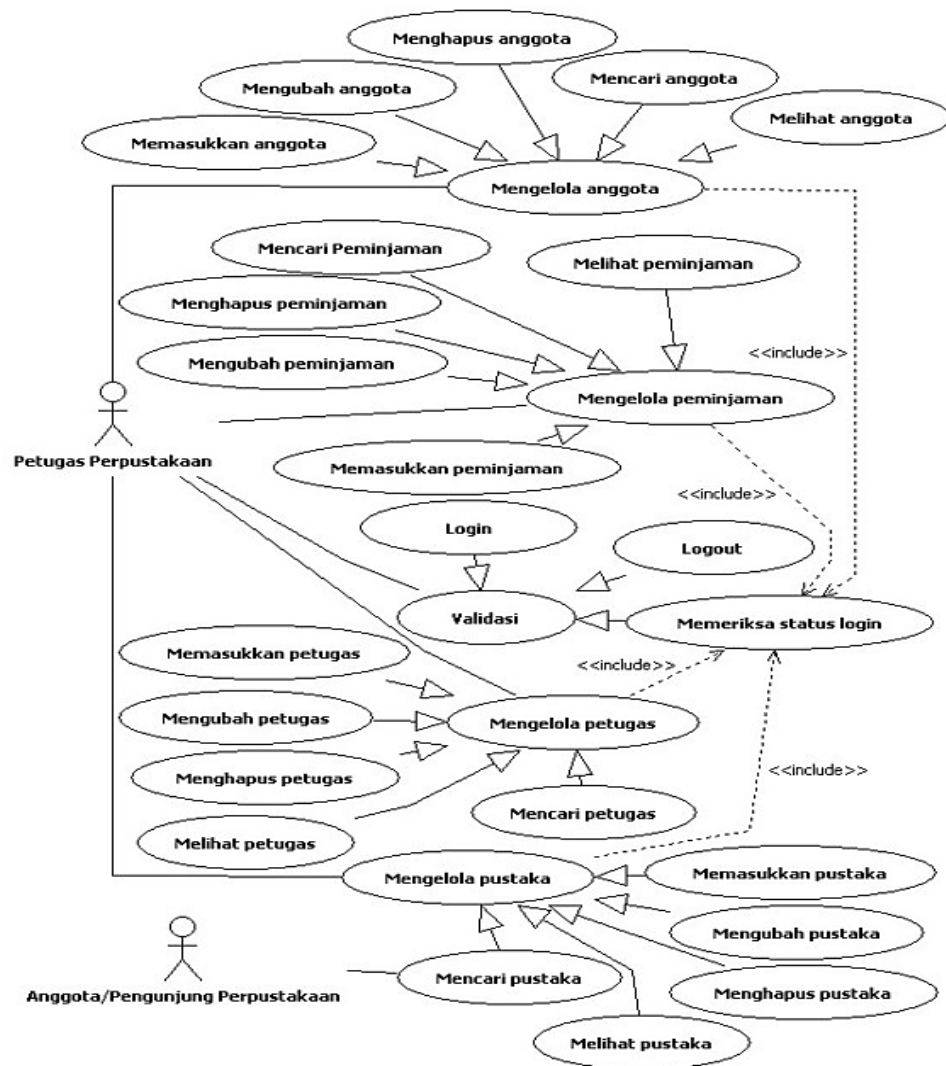
Adapun jenis-jenis diagram UML yang digunakan ada 6 buah, diantaranya sebagai berikut:

1. *Use Case Diagram*

“*Use case* atau *diagram use case* merupakan pemodelan untuk untuk kelakuan (*behavior*) sistem informasi yang akan dibuat” *Use Case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Selain itu *Use Case* digunakan untuk mengetahui fungsi apa saja yang ada didalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu (Sukamto, Rosa A. dan Shalahudin, 2015).

Use case atau *diagram use case* merupakan pemodelan untuk memodelkan kelakuan (*behavior*) dari sebuah sistem informasi (Sukamto & Shalahuddin, 2018). *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor

dengan sistem informasi yang akan dibuat. Aktor tidak harus berupa personal manusia, dapat juga berupa sistem atau sub sistem. Ada 3 komponen utama dari *use case diagram*, yaitu : Aktor, *use case*, dan relasi. Berikut contoh dari *use case diagram*:

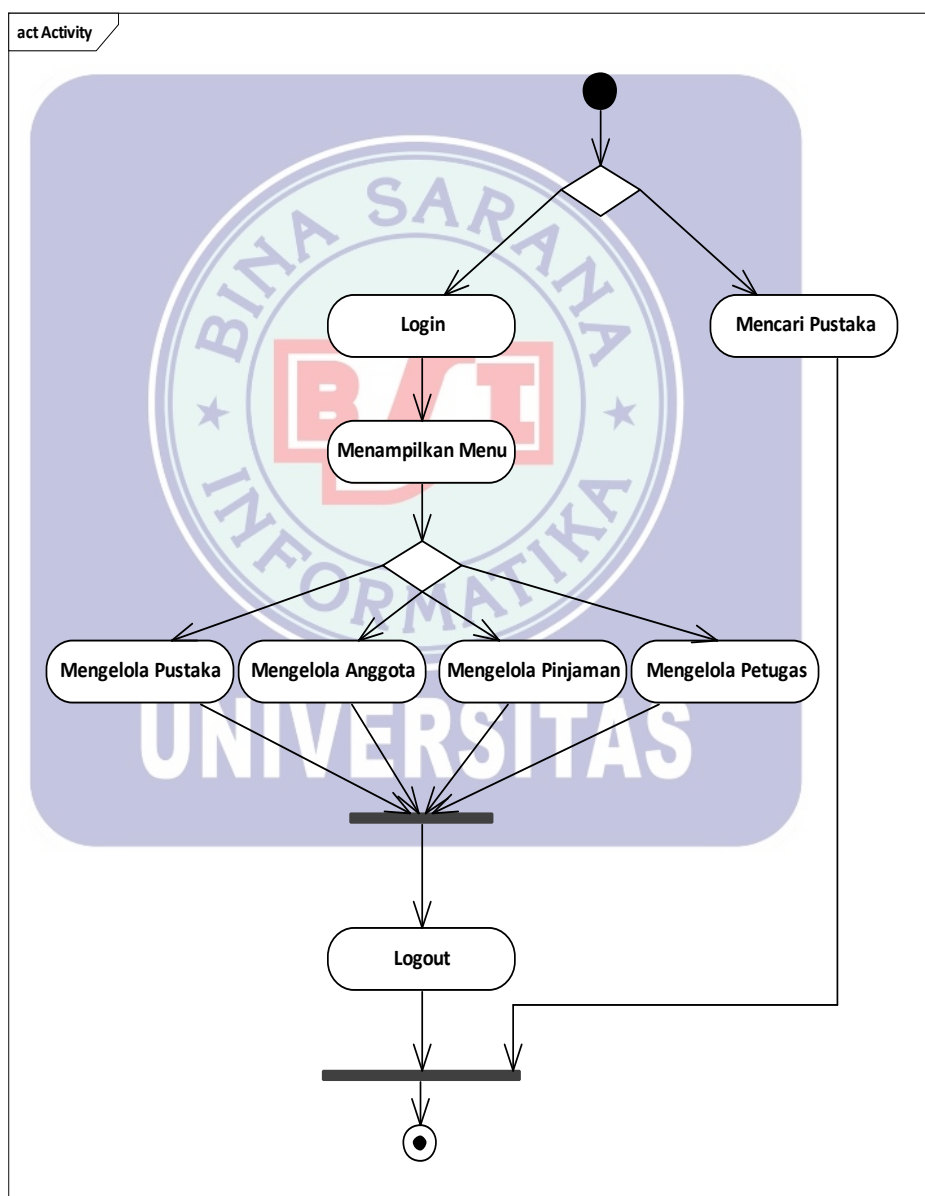


Sumber: (Sukamto & Shalahuddin, 2018)

Gambar II.2.
Use Case Diagram

2. Activity Diagram

Activity diagram digunakan untuk menggambarkan aliran kerja atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak (Sukamto & Shalahuddin, 2018). *Activity diagram* lebih menekankan terhadap apa yang sistem lakukan, bukan apa yang aktor lakukan. Berikut contoh *activity diagram*:

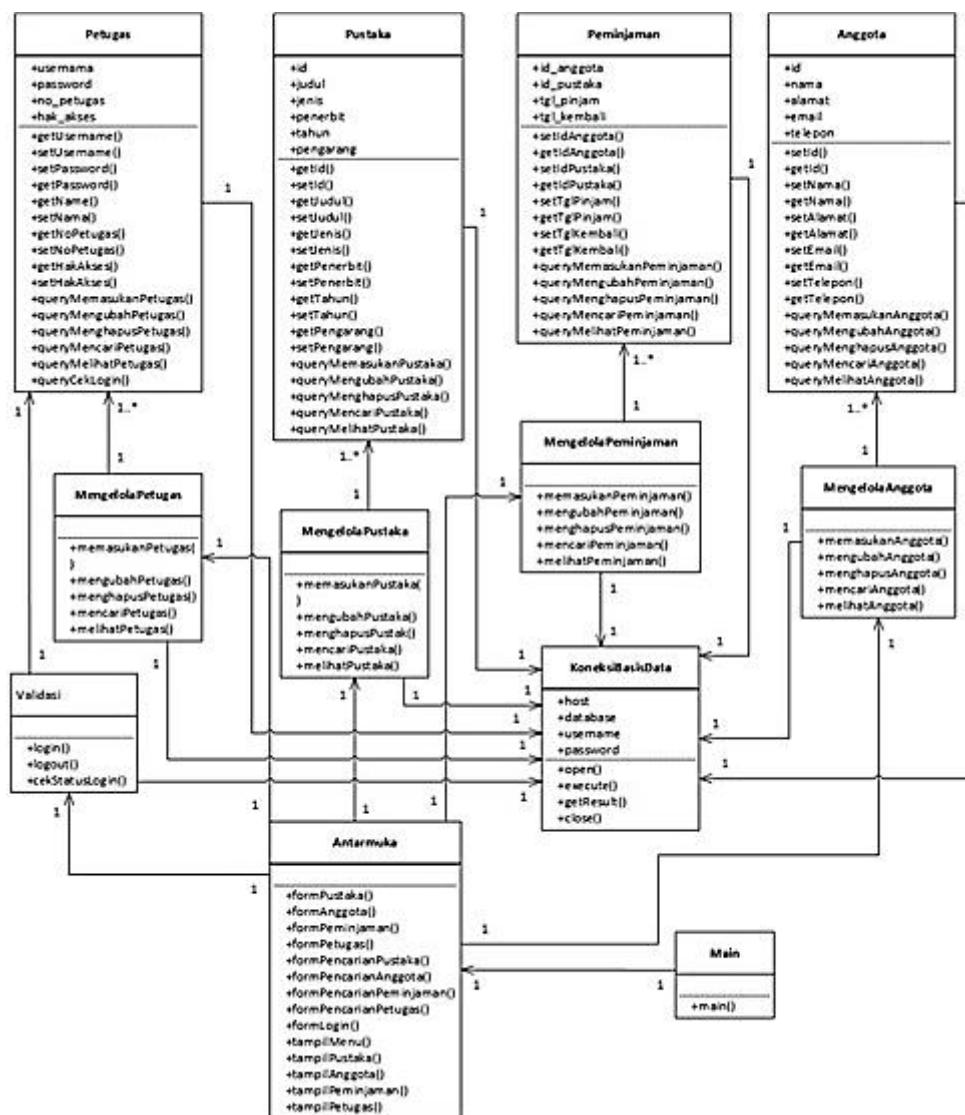


Sumber: (Sukamto & Shalahuddin, 2018)

Gambar II.3.
Activity Diagram

3. Class Diagram

Class diagram menggambarkan struktur kelas-kelas yang akan dibuat dalam membangun sistem (Sukamto & Shalahuddin, 2018). Kelas memiliki apa yang disebut atribut dan *metode* atau operasi. Berikut contoh *class diagram*:



Sumber: (Sukamto & Shalahuddin, 2018)

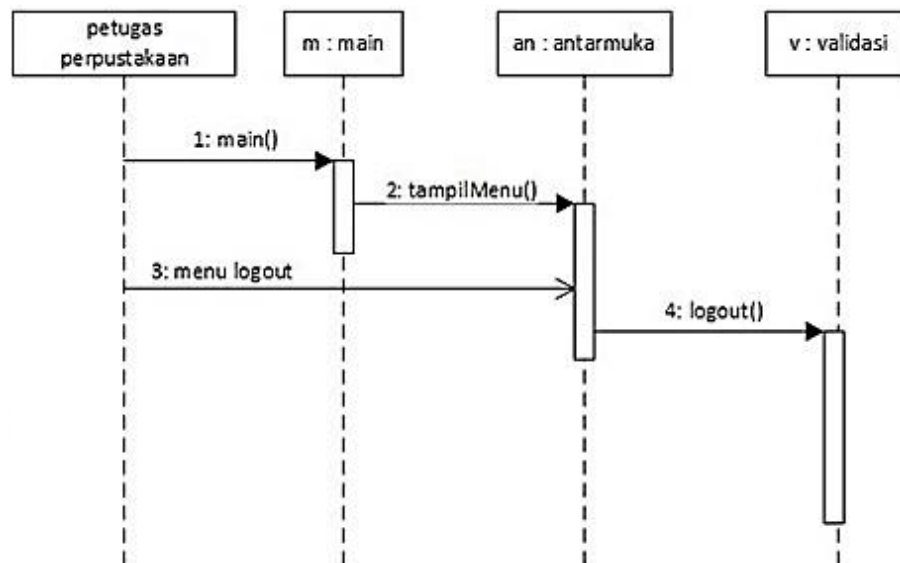
Gambar II.4.
Class Diagram

4. Sequence Diagram

Diagram sekuen menggambarkan kelakuan (*behavior*) objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan

diterima antar objek berdasarkan urutan tertentu (Sukamto & Shalahuddin, 2018).

Berikut contoh sequence diagram:

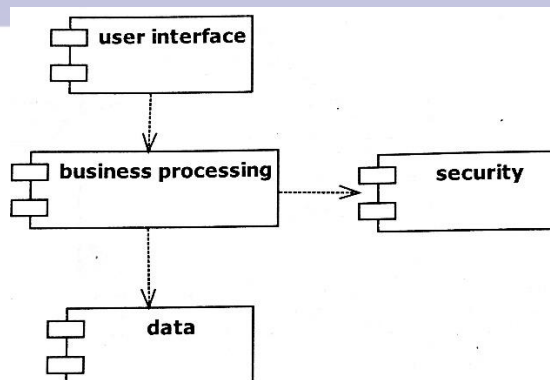


Sumber: (Sukamto & Shalahuddin, 2018)

Gambar II.5.
Sequence Diagram

5. Component Diagram

Component diagram dibuat untuk menunjukkan organisasi dan ketergantungan diantara kumpulan komponen dalam sebuah sistem (Sukamto & Shalahuddin, 2018). Berikut contoh *component diagram*:

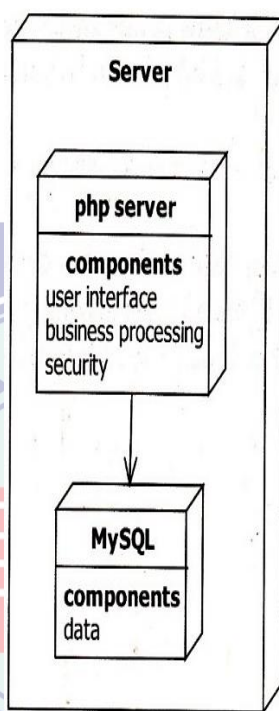


Sumber: (Sukamto & Shalahuddin, 2018)

Gambar II.6.
Component Diagram

6. *Deployment Diagram*

Deployment diagram lebih menunjukkan konfigurasi komponen dalam proses eksekusi aplikasi (Sukamto & Shalahuddin, 2018).



Sumber: (Sukamto & Shalahuddin, 2018)

Gambar II.7.
Component Diagram

2.2.2. *Entity Relationship Diagram (ERD)*

Simarmata mendefinisikan *entity relationship diagram* adalah “alat pemodelan data utama dan akan membantu mengorganisasi data dalam suatu proyek ke dalam entitas-entitas dan menentukan hubungan antar entitas” (Fridayanthie & Mahdiati, 2016). Menurut (Indrajani, 2015) *entity relationship diagram (ERD)* adalah “sebuah pendekatan *top-bottom* dalam perancangan basis data yang dimulai dengan mengidentifikasi data-data terpenting yang disebut entitas dan hubungan antara entitas-entitas tersebut digambarkan dalam suatu model atau diagram”.

Hubungan pada ERD yang paling umum terdiri dari:

1. *One to one* (satu ke satu)

Setiap elemen dari Entitas A memiliki hubungan paling banyak satu dengan elemen pada Entitas B. Begitu juga sebaliknya.

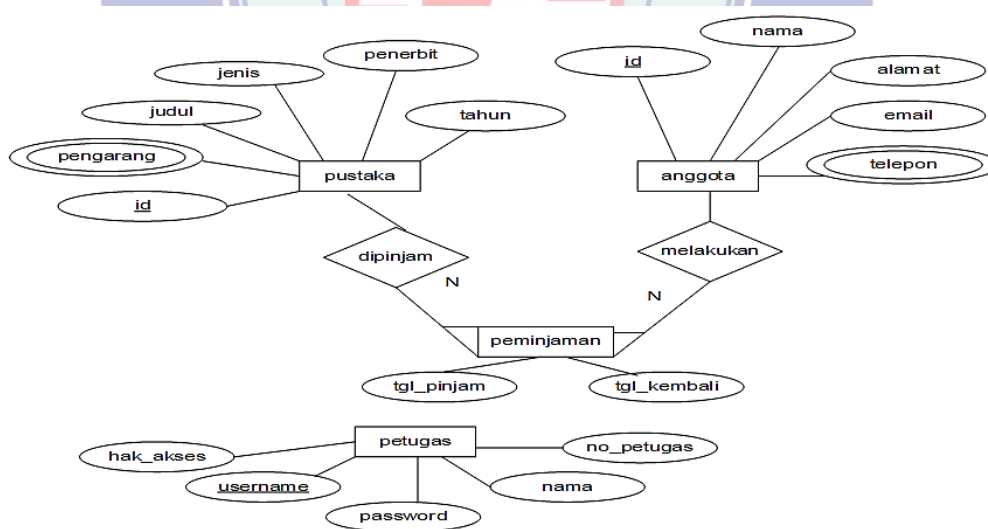
2. *One to many* (satu ke banyak)

Setiap elemen atau salah satu elemen pada Entitas A, dapat memiliki hubungan minimal satu sampai tak hingga dengan elemen pada entitas B.

3. *Many to many* (banyak ke banyak)

Setiap elemen pada Entitas A, dapat memiliki hubungan tak hingga dengan elemen pada Entitas B, dan berlaku pula setiap elemen pada Entitas B memiliki hubungan tak hingga dengan elemen pada Entitas A.

Berikut contoh *entity relationship diagram*:



Sumber : (Sukamto & Shalahuddin, 2018)

Gambar II.8.
Entity Relationship Diagram

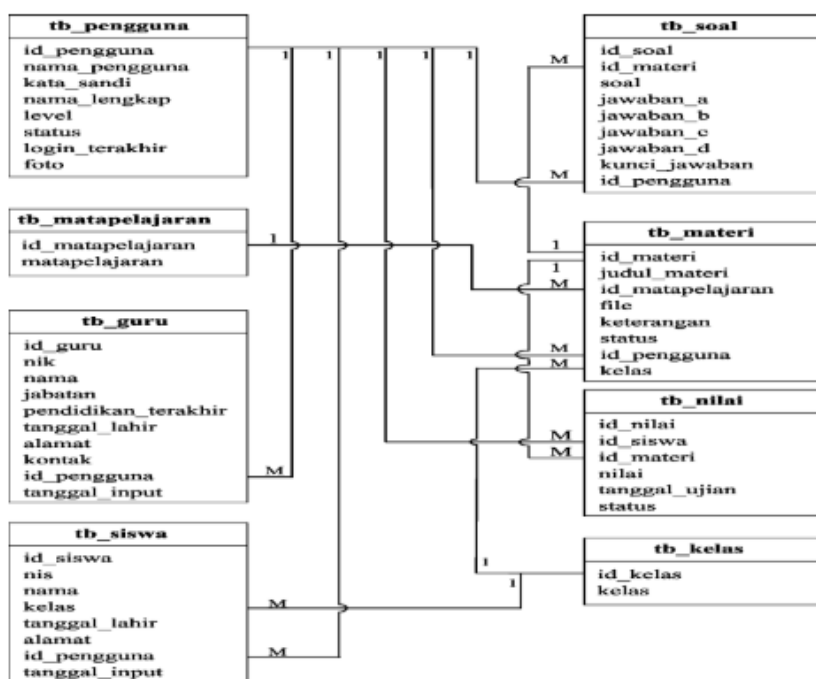
Berdasarkan definisi diatas dapat disimpulkan bahwa ERD merupakan pemodelan basis data dalam suatu sistem.

2.2.3. Logical Relational Structure (LRS)

Hasugian dan Shidiq Memberikan batasan bahwa LRS adalah “sebuah model sistem yang digambarkan dengan sebuah diagram – ER akan mengikuti pola atau aturan permodelan tertentu dalam kaitan dengan konvensi ke LRS” (Andika & Buani, 2017), sementara Simarmata dan Paryudi menerangkan bahwa *logical record structured* (LRS) representasi dari struktur *record-record* pada tabel-tabel yang terbentuk dari hasil relasi antar himpunan entitas” (Fridayanthie & Mahdiati, 2016).

Tabrani juga mengemukakan bahwa “*Logical Record Structure* (LRS) dibentuk dengan nomor dari tipe *record*” (Kuryanti & Sandra, 2016). Beberapa tipe *record* digambarkan oleh kotak persegi panjang dan dengan nama yang unik. Perbedaan LRS dengan E-R diagram adalah nama tipe *record* berada di luar kotak *field* tipe *record* ditempatkan (Kuryanti & Sandra, 2016).

Jadi LRS dapat diartikan juga bahwa LRS merupakan hasil transformasi dari diagram E-R dengan menggambarkan entitas dan atribut dalam sebuah kotak. Berikut contoh dari *logical record structure*:



Sumber : (Kuryanti & Sandra, 2016)

Gambar II.9.
Logical Record Structure

2.2.4. Basis Data

Fred R. McFadden mengemukakan basis data adalah “sebuah kumpulan terorganisasi dari data-data yang berhubungan secara logika”. Menurut (Sukanto & Shalahuddin, 2018) basis data adalah “sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan”. Definisi lain dari basis data atau *database* adalah “sekumpulan data yang sudah disusun sedemikian rupa dengan ketentuan atau aturan tertentu yang saling berelasi sehingga memudahkan pengguna dalam mengelola data dan juga mempermudah dalam memperoleh informasi” (Rozak, 2019).

Dengan demikian dapat disimpulkan bahwa basis data merupakan kumpulan data terorganisir berupa kumpulan *file* atau tabel yang berperan untuk menyimpan data yang dapat diolah sehingga menghasilkan informasi yang dibutuhkan.

2.2.6. Pengertian *Sequence Diagram*

Sequence Diagram adalah diagram yang menggambarkan kelakuan objek pada *Use Case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek (Sukamtoo, R. A., & Shalahudin, 2015).

Menurut (Fridayanthie, 2016) “Diagram komponen dibuat untuk menunjukan organisasi dari ketergantungan diantara kumpulan-kumpulan komponen dalam sebuah sistem”.

Sedangkan menurut (Sugiarti, Y., 2019) “Diagram sequence menggambarkan behavior objek pada use case dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek”.

Berdasarkan uraian diatas dapat disimpulkan bahwa *Sequence Diagram* merupakan diagram yang menggambarkan behavior objek dalam sebuah sistem.

2.2.5. Java

Bahasa pemrograman *Java* merupakan salah satu dari sekian banyak bahasa pemrograman yang dapat dijalankan di berbagai sistem operasi termasuk telepon genggam (Nofriadi, 2015). Kelebihan *Java* dari bahasa pemrograman yang lain adalah bisa dijalankan di berbagai jenis sistem operasi sehingga dikenal juga bahasa pemrograman *multiplatform*, bersifat pemrograman berorientasi obyek (PBO), memiliki *library* yang lengkap (Nofriadi, 2015).

Java awalnya dimiliki oleh Sun Microsystems sebelum akhirnya Sun diakuisisi oleh Oracle pada 2010. *Java* menjadi bahasa pemrograman ke-3 terpopuler setelah JavaScript dan Python pada tahun 2019 (Chan, 2019), hal ini dikarenakan *Java* dapat digunakan untuk kebutuhan berbagai macam kebutuhan, baik desktop, web, bahkan *embeded system*.

2.2.6. MySQL

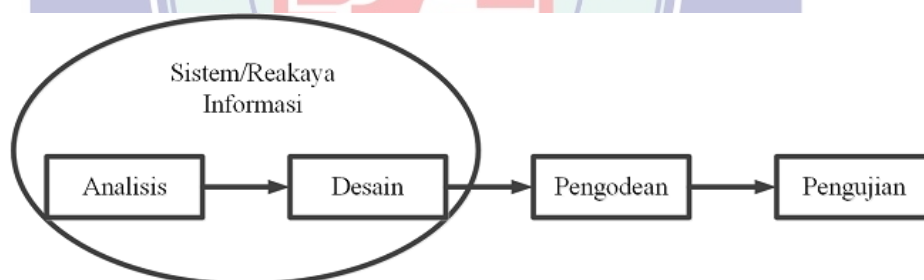
Arief memberikan definisi terhadap MySQL sebagai “salah satu jenis *database server* yang sangat terkenal dan banyak digunakan untuk membangun aplikasi web yang menggunakan *database* sebagai sumber dan pengelolaan datanya” (Fridayanthie & Mahdiati, 2016). Di sisi lain Puspitosari memberikan definisi

MySQL sebagai “salah satu *software* untuk *database* server yang banyak digunakan, *Mysql* bersifat *open source* dan menggunakan SQL” (Sintawati & Sari, 2017) .

Pada dasarnya MySQL merupakan proyek sumber terbuka, namun setelah diambil alih oleh Oracle, ada beberapa bagian dari MySQL yang tidak lagi merupakan sumber terbuka, sehingga lahir MariaDB yang dibuat oleh *developer* yang telah membuat MySQL (MariaDB Team, 2019).

2.2.7. Pengembangan Perangkat Lunak Metode *Waterfall*

Model SDLC air terjun (*waterfall*) sering juga disebut juga model sekuensial linier (*sequential linear*) atau alur hidup klasik (*classic life cycle*) (Sukamto & Shalahuddin, 2018). Model ini merupakan model yang paling sederhana dalam perancangan sistem. Berikut adalah gambar model air terjun :



Sumber : (Sukamto & Shalahuddin, 2018)

Gambar II.10.
Model *Waterfall*

Beberapa tahapan dalam perancangan sistem dengan metode waterfall antara lain (Sukamto & Shalahuddin, 2018):

1. Analisis Kebutuhan Perangkat Lunak

Proses pengumpulan kebutuhan dilakukan secara intensif untuk mespesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh *user*.

2. Desain

Desain perangkat lunak adalah proses multi langkah yang fokus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antar muka, dan prosedur pengodean.

3. Pembuatan Kode Program

Desain harus ditranslasikan ke dalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain.

4. Pengujian

Pengujian fokus pada perangkat lunak secara dari segi logik dan fungsional dan memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.

5. Pendukung (*Support*) Atau Pemeliharaan (*Maintenance*)

Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirimkan ke *user*. Perubahan bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan baru. Tahap pendukung atau pemeliharaan dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan perangkat lunak yang sudah ada, tapi tidak untuk membuat perangkat lunak baru.

2.2.8. *Blackbox Testing*

Untuk menjaga agar sistem dapat terjaga kualitasnya, perlu dilakukan pengujian terhadap semua fungsi dari sistem atau aplikasi yang telah dibuat. Tujuan dilakukannya pengujian terhadap perangkat lunak yang telah dibuat adalah untuk

memastikan bahwa perangkat lunak berjalan sesuai dengan yang diharapkan juga terjamin dari sisi keamanannya. Setidaknya ada 2 macam pengujian perangkat lunak yang lazim dilakukan, yaitu *blackbox testing* dan *whitebox testing*. Namun, pada perancangan sistem ini, hanya akan melakukan pengujian dengan metode *blackbox testing*.

Secara umum *blackbox testing* adalah menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program (Sukamto & Shalahuddin, 2018). Pengujian dimaksudkan untuk mengetahui apakah fungsi-fungsi, masukan, dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan. Pengujian kotak hitam dilakukan dengan membuat kasus uji yang bersifat mencoba semua fungsi dengan memakai perangkat lunak apakah sesuai dengan spesifikasi yang dibutuhkan. Kasus uji yang dibuat untuk melakukan pengujian kotak hitam harus dibuat dengan kasus benar dan kasus salah, misalkan untuk kasus proses *login* maka kasus uji yang dibuat adalah:

- a. Jika *user* memasukkan nama pemakai (*username*) dan kata sandi (*password*) yang benar.
- b. Jika *user* memasukkan nama pemakai (*username*) dan kata sandi (*password*) yang salah, misalnya nama pemakai benar tapi kata sandi salah, atau sebaliknya, atau keduanya salah.