

BAB II

LANDASAN TEORI

II.1. Definisi Sistem

Terdapat dua kelompok pendekatan di dalam pendefinisian sistem, yaitu sekelompok yang menekankan pada prosedur dan kelompok yang menekankan pada elemen atau komponennya. Pendekatan yang menekankan pada prosedur dan mendefinisikan sistem sebagai suatu jaringan kerja prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu. Sedangkan pendekatan sistem yang lebih menekankan pada elemen atau komponen mendefinisikan sistem sebagai kumpulan elemen yang berinteraksi untuk mencapai suatu tujuan tertentu. kedua kelompok definisi ini adalah benar dan tidak bertentangan. Yang berbeda adalah cara pendekatannya. (Tata Sutabri : 2012 ; 2).

Teori sistem melahirkan konsep-konsep futuristik, antara lain yang terkenal adalah konsep sibernetika (*cyberbetics*). Konsep atau bidang kajian ilmiah ini terutama berkaitan dengan upaya-upaya untuk menerapkan berbagai disiplin ilmu, yaitu ilmu perilaku, fisikan, biologi, dan teknik. Oleh karena itu sibernetika biasanya berkaitan dengan usaha-usaha otomasi tugas-tugas yang dilakukan oleh manusia, sehingga melahirkan studi-studi tentang robotika, kecerdasan buatan (*artificial intelegence*) dan lain sebagainya. Unsur-unsur yang mewakili suatu sistem secara umum adalah masukan (*input*), pengolahan (*processing*) dan keluaran (*output*). Di samping itu suatu sistem senantiasa tidak

terlepas dari lingkungan sekitarnya. Maka umpan balik (*feed-back*) selain dapat berasal dari output, juga dapat berasal dari lingkungan sistem tersebut. Organisasi dipandang sebagai suatu sistem yang juga memiliki semua unsur. (Tata Sutabri : 2012 ; 3).

II.1.1. Informasi

Informasi adalah data yang telah diklasifikasikan atau diolah atau diinterpretasikan untuk digunakan dalam proses pengambilan keputusan. Sistem pengolahan informasi akan mengolah data menjadi informasi atau mengolah data dari bentuk tak berguna menjadi berguna bagi yang menerimanya. Nilai informasi berhubungan dengan keputusan. Bila tidak ada pilihan atau keputusan maka informasi tidak diperlukan. Keputusan dapat berkisar dari keputusan berulang sederhana sampai keputusan strategis jangka panjang. Nilai informasi dilukiskan paling berarti dalam konteks pengambilan keputusan (Tata Sutabri ; 2012 : 21-22).

II.1.2. Konsep Dasar Sistem Informasi

Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan laporan-laporan yang diperlukan oleh pihak luar tertentu.

Tipe sistem informasi adalah sebagai berikut:

1. Sistem informasi Akuntansi
2. Sistem informasi Pemasaran
3. Sistem informasi Manajemen Persediaan
4. Sistem informasi Personalia
5. Sistem informasi Distribusi
6. Sistem informasi Pembelian
7. Sistem informasi Kekayaan
8. Sistem informasi Analisis Kredit
9. Sistem informasi Penelitian dan Pengembangan
10. Sistem informasi Teknik

Semua sistem informasi tersebut dimaksudkan untuk memberikan informasi kepada semua tingkat manajemen, mulai manajemen tingkat bawah, manajemen tingkat menengah, hingga manajemen tingkat atas (Tata Sutabri ; 2012: 38-41).

II.2. Sistem Pendukung Keputusan

Konsep Sistem Pendukung Keputusan pertama kali dikemukakan oleh Michael S.Scott Morton pada tahun 1970 dengan istilah *Management Decision System*. Konsep tersebut ditandai dengan dibuatnya sebuah sistem komputer yang interaktif dalam mengumpulkan data dan menggunakan model untuk mendukung keputusan penyelesaian masalah-masalah yang tidak terstruktur. Secara umum SPK dapat diartikan sebagai sistem yang dimaksudkan untuk mendukung para pengambil keputusan manajerial dalam situasi keputusan semi terstruktur. SPK

dimaksudkan untuk menjadi alat bantu bagi para pengambil keputusan untuk memperluas kapabilitas mereka, namun tidak untuk menggantikan penilaian mereka. Sulpan Henry Siregar menyatakan SPK ditujukan untuk keputusan-keputusan yang memerlukan penilaian atau pada keputusan-keputusan yang sama sekali tidak dapat didukung oleh algoritma.

Beberapa definisi lain dari para ahli tentang SPK antara lain menurut Mat dan Watson SPK merupakan suatu sistem interaktif yang membantu pengambilan keputusan melalui penggunaan data dan model-model keputusan untuk memecahkan masalah-masalah yang sifatnya semi terstruktur dan tidak terstruktur. Sedangkan menurut Alter SPK adalah sistem informasi interaktif yang menyediakan informasi, pemodelan, dan pemanipulasian data. Sistem ini digunakan untuk membantu pengambilan keputusan dalam situasi yang semi terstruktur dan situasi yang tidak terstruktur, dimana tidak seorang pun tahu secara pasti bagaimana keputusan seharusnya dibuat (Made Astradanta, et al., 2016 : 3).

Aplikasi sistem pendukung keputusan menggunakan data, memberikan antar muka pengguna yang mudah, dan dapat menggabungkan pemikiran pengambilan keputusan. Sistem pendukung keputusan tidak dimaksudkan untuk mengotomatisasikan pengambilan keputusan, tetapi memberikan perangkat interaktif yang memungkinkan pengambilan keputusan untuk melakukan berbagai analisis menggunakan model-model yang tersedia. Tujuan dari DSS adalah (Sylvia Hartati Saragih, 2013 : 83) :

1. Membantu manajer dalam pengambilan keputusan atas masalah semistuktur.

2. Memberikan dukungan atas pertimbangan manajer dan bukannya dimaksudkan untuk menggantikan fungsi manajer.
3. Meningkatkan efektifitas keputusan yang diambil lebih daripada perbaikan efisiensinya.
4. Kecepatan komputasi. Komputer memungkinkan para pengambil keputusan untuk melakukan banyak komputansi secara cepat dengan biaya rendah.
5. Peningkatan produktivitas.
6. Dukungan kualitas.
7. Berdaya saing.
8. Mengatasi keterbatasan kognitif dalam pemrosesan dan penyimpanan.

II.3. Metode TOPSIS

TOPSIS adalah salah satu metode pengambilan keputusan multikriteria yang pertama kali diperkenalkan oleh Yoon dan Hwang pada tahun 1981. Topsis menggunakan prinsip bahwa alternatif harus mempunyai jarak terdekat dari solusi ideal positif dan jarak terpanjang (terjauh) dari solusi ideal negatif dari sudut pandang geometris dengan menggunakan jarak *Euclidean* (jarak antara dua titik) untuk menentukan kedekatan relatif dari suatu alternatif.

Solusi ideal positif didefinisikan sebagai jumlah dari seluruh nilai terbaik yang dapat dicapai untuk setiap atribut, sedangkan solusi negatif ideal terdiri dari seluruh nilai terburuk yang dicapai untuk setiap atribut. TOPSIS mempertimbangkan keduanya, jarak terhadap solusi ideal positif dan jarak

terhadap solusi ideal negatif dengan mengambil kedekatan relatif terhadap solusi ideal positif.

Metode TOPSIS banyak digunakan untuk menyelesaikan pengambilan keputusan secara praktis. Hal ini disebabkan konsepnya sederhana dan mudah dipahami, komputasinya efisien, dan memiliki kemampuan mengukur kinerja relative dari alternatif-alternatif keputusan (Muhammad Husein, 2016 : 148-149).

Adapun tahapan dari metode topsis yaitu :

1. Membuat matriks keputusan yang ternormalisasi

Untuk mendapatkan matriks normalisasi r, setiap normalisasi dari nilai rij dapat dilakukan dengan perhitungan membuat matriks keputusan ternormalisasi r yang elemen-elemennya ditentukan dengan rumus berikut :

$$r_{ij} = \frac{x_{ij}}{\sum_{i=1}^m x_j^2} \dots \dots \dots (II.1)$$

2. Membuat matriks keputusan yang ternormalisasi terbobot.
3. Menentukan matriks solusi ideal positif dan matriks solusi ideal negatif.

Rumus:

$$A^+ = \max(y1+, y2+, \dots, yn+) \text{ dan } A^- = \max(y1-, y2-, \dots, yn-) \dots (II.2)$$

4. Menentukan jarak antara nilai setiap alternatif dengan matriks solusi ideal positif dan negatif.

$$D_i^+ = \sqrt{\sum_{i=1}^n (v_{ij} - v_{ij}^+)^2}, \text{ Untuk } i = 1, 2, 3, \dots, m \dots \dots \dots (II.3)$$

$$D_i^- = \sqrt{\sum_{j=1}^n (v_{ij} - v_{ij}^-)^2}, \text{ Untuk } i = 1, 2, 3, \dots, m \dots \dots \dots (11.4)$$

5. Menentukan nilai preferensi untuk setiap alternatif atau kedekatan relatif dengan solusi ideal.

$$V_i = \frac{D_i^-}{D_i^- + D_i^+} \dots \dots \dots (11.5)$$

Nilai V_i yang lebih besar menunjukkan bahwa alternatif i lebih dipilih.

II.4. Metode AHP

Analytical Hierarchy Process (AHP) merupakan metode pendukung pengambilan keputusan yang dikembangkan oleh Thomas L., Saaty pada tahun 1980. AHP merupakan alat pengambil keputusan yang menguraikan suatu permasalahan kompleks dalam struktur hierarki dengan banyak tingkatan yang terdiri dari tujuan, kriteria, dan alternatif. Hierarki didefinisikan sebagai suatu representasi dari sebuah permasalahan yang kompleks dalam suatu struktur Multilevel dimana level pertama adalah tujuan, yang diikuti level faktor, kriteria, subkriteria, dan seterusnya ke bawah hingga level terakhir dari alternatif. Dengan hierarki, suatu masalah yang kompleks dapat diuraikan ke dalam kelompok-kelompoknya yang kemudian diatur menjadi suatu bentuk hierarki sehingga permasalahan akan tampak lebih terstruktur dan sistematis.

Dalam menggunakan metode AHP terdapat beberapa langkah atau tahap yang harus dilakukan. Adapun tahapan-tahapan itu adalah :

1. Mendefinisikan masalah dan menentukan solusi yang diinginkan.
2. Membuat struktur hierarki yang diawali dengan tujuan utama.

3. Membuat matriks perbandingan berpasangan
4. Mendefinisikan perbandingan berpasangan
5. Menghitung nilai eigen matriks
6. Menghitung vektor eigen dari setiap matriks
7. Memeriksa konsistensi hierarki.

Yang diukur dalam AHP adalah rasio konsistensi dengan melihat indeks konsistensi. Konsistensi yang diharapkan adalah yang mendekati sempurna agar menghasilkan keputusan yang mendekati valid. Walaupun sulit untuk mencapai yang sempurna, rasio konsistensi diharapkan kurang dari atau sama dengan 10 %. Berikut adalah cara menghitung rasio konsistensi.

Menghitung konsistensi indeks (CI) dengan rumus :

$$CI = \frac{(\max - n)}{n} \dots\dots\dots (II.6)$$

max adalah jumlah hasil kali perkalian jumlah kolom dengan eigen faktor utama dan n adalah jumlah kriteria.

Menghitung rasio konsistensi CR :

$$CR = CI/IR \dots\dots\dots (II.7)$$

Persamaan Konsistensi Rasio dengan IR adalah nilai random indeks sesuai dengan ordo matriks.

Memeriksa konsistensi hierarki :

Jika nilainya lebih dari 10%, maka penilaian data judgment harus diperbaiki, namun jika rasio konsistensi CI/IR kurang atau sama dengan 0,1, maka hasil perhitungan bisa dinyatakan benar (Made Astradanta, et al., 2016 : 3-4).

II.5. Definisi Database

Menurut Edhy Sutanta (2011), *Database* adalah kumpulan data yang saling berelasi. Data sendiri merupakan fakta mengenai obyek, orang, dan lain-lain. Data dinyatakan dengan nilai (angka, deretan karakter, atau simbol) (Deppi Linda, 2016 : 63).

Menurut Abdul Kadir, (2009:14) *Database* didefinisikan sebagai kumpulan data yang terkait secara teknis, yang berada dalam sebuah *database* adalah sekumpulan tabel atau objek lain (*indeks, view, dan lain-lain*). *Database* adalah tempat media penyimpanan data kita dalam membuat sebuah program yang berisikan tabel, *field* dan *record*, yang diselimuti namanya DBMS (*Database Management System*). Berdasarkan uraian tersebut maka dapat disimpulkan *database* adalah dimana dapat menyimpan suatu kumpulan data yang terkait secara teknis dan dapat membuat sebuah program yang berisikan tabel, *field* dan *record* (Ekkal Prasetyo, 2015 : 22).

II.6. Normalisasi

Normalisasi diartikan sebagai suatu teknik yang menstrukturkan mendekomposisi data dalam cara-cara tertentu untuk mencegah timbulnya permasalahan pengolahan data dalam basis data. Permasalahan yang dimaksud adalah berkaitan dengan penyimpangan-penyimpangan (*anomallies*) yang terjadi akibat adanya kerangkapan data dalam relasi dan in-efisiensi pengolahan (Martin, 1975) : (Edy Sutanta ; 2011 : 174-175).

Secara berturut-turut masing-masing level normal tersebut dibahas berikut ini, dimulai dari bentuk tidak normal. (Edy Sutanta ; 2011 : 176-179)

1. Relasi bentuk tidak normal (*Un Normalized Form* / UNF)

Relasi-relasi yang dirancang tanpa mengindahkan batasan dalam defisi basis data dan karakteristik *Relational Database Management System* (RDBM) menghasilkan relasi *Un Normalized Form* (UNF). Bentuk ini harus di hindari dalam perancangan relasi dalam basis data. Relasi *Un Normalized Form* (UNF) mempunyai kriteria sebagai berikut :

- a. Jika relasi mempunyai bentuk *non flat file* (dapat terjadi akibat data disimpan sesuai dengan kedatangannya, tidak memiliki struktur tertentu, terjadi duplikasi atau tidak lengkap)
- b. Jika relasi membuat *set atribut* berulang (*non single values*)
- c. Jika relasi membuat *atribut non atomic value*

2. Relasi bentuk normal pertama (*First Norm Form* / 1NF)

Relasi disebut juga *First Norm Form* (1NF) jika memenuhi kriteria sebagai berikut :

- a. Jika seluruh atribut dalam relasi bernilai *atomic* (*atomic value*)
- b. Jika seluruh atribut dalam relasi bernilai tunggal (*single value*)
- c. Jika relasi tidak memuat set atribut berulang
- d. Jika semua record mempunyai sejumlah atribut yang sama.

Permasalahan dalam *First Norm Form* (1NF) adalah sebagai berikut.

- a. Tidak dapat menyisipkan informasi parsial

b. Terhapusnya informasi ketika menghapus sebuah *record*

3. Bentuk normal kedua (*Second Normal Form / 2NF*)

Relasi disebut sebagai *Second Normal Form (2NF)* jika memenuhi kriteria sebagai berikut :

a. Jika memenuhi kriteria *First Norm Form (1NF)*

b. Jika semua atribut nonkunci *Functional Dependence (FD)* pada *Primary Key (PK)*

Permasalahan dalam *Second Normal Form / 2NF* adalah sebagai berikut :

a. Kerangkapan data (*data redundancy*)

b. Pembaharuan yang tidak benar dapat menimbulkan inkonsistensi data (*data inconsistency*)

c. Proses pembaharuan data tidak efisien

4. Bentuk normal ketiga (*Third Norm Form / 3NF*)

Suatu relasi disebut sebagai *Third Norm Form* jika memenuhi kriteria sebagai berikut.

a. Jika memenuhi kriteria *Second Normal Form (2NF)*

b. Jika setiap atribut nonkunci tidak (*TDF*) (*Non Transitive Dependency*) terhadap *Primary Key (PK)*

Permasalahan dalam *Third Norm Form (3NF)* adalah keberadaan penentu yang tidak merupakan bagian dari *Primary Key (PK)* menghasilkan duplikasi rinci data pada atribut yang berfungsi sebagai *Foreign Key (FK)* (duplikasi berbeda dengan keterangan data).

Mengubah relasi *Second Normal Form* (2NF) menjadi bentuk *Third Normal Form* (3NF) dapat dilakukan dengan mengubah struktur relasi dengan cara :

- a. Identifikasi TDF relasi *Second Normal Form* (2NF)
- b. Berdasarkan informasi tersebut, dekomposisi relasi *Second Normal Form* (2NF) menjadi relasi-relasi baru sesuai TDF-nya.

II.7. *Unified Modeling Language* (UML)

Menurut Windu Gata, Grace (2013:4), *Unified Modeling Language* (UML) adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak. UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem (Ade Hendini, 2016 : 108).

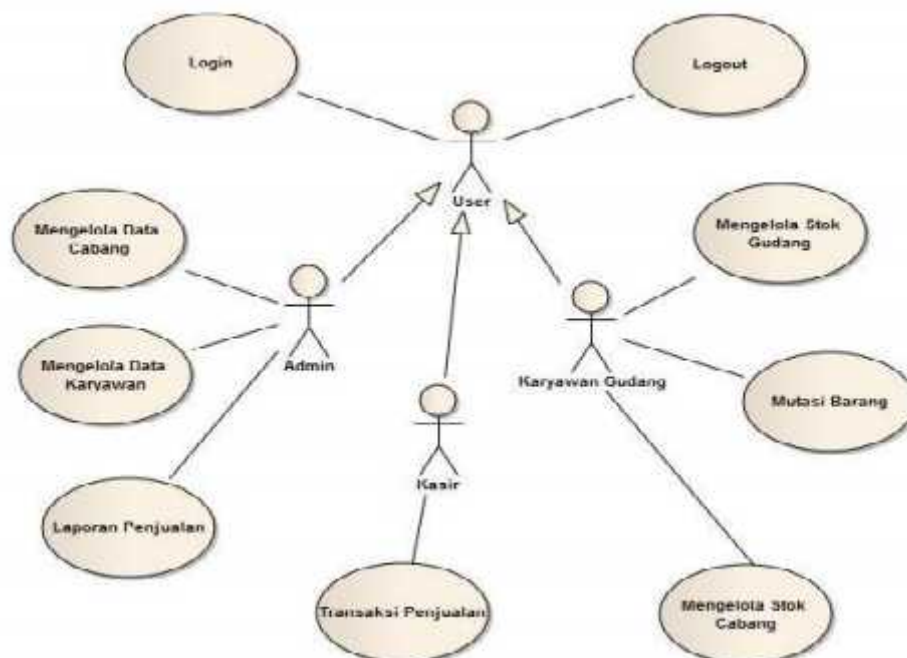
Unified Modelling Language (UML) adalah sebuah bahasa yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak sebuah sistem. UML lebih mengedepankan penggunaan diagram untuk menggambarkan aspek dari sistem, karena tergolong bahasa visual yang lebih mudah dan lebih cepat dipahami dibandingkan dengan bahasa pemrograman.

Pemodelan penggunaan UML merupakan metode pemodelan berorientasi objek dan berbasis visual. Karenanya pemodelan objek yang fokus pada pendefinisian struktur statis dan model sistem informasi yang dinamis daripada

mendefinisikan data dan model proses yang tujuannya adalah pengembangan tradisional. UML menawarkan diagram yang dikelompokkan menjadi lima perspektif berbeda untuk memodelkan suatu sistem. Seperti satu set *blue print* yang digunakan untuk membangun sebuah rumah (Saipul Anwar, et al., 2016 : 75-76).

II.7.1. Use Case Diagram

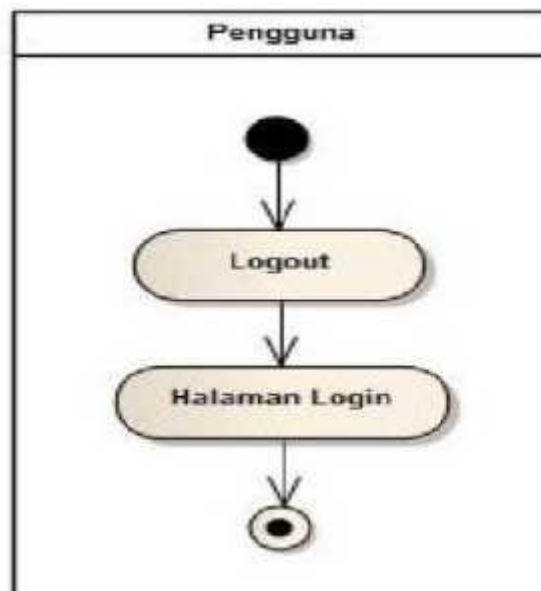
Use case diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut (Ade Hendini, 2016 : 108).



Gambar II.1. Use Case Diagram
(Sumber : Ade Hendini, 2016 : 112)

II.7.2. Activity Diagram

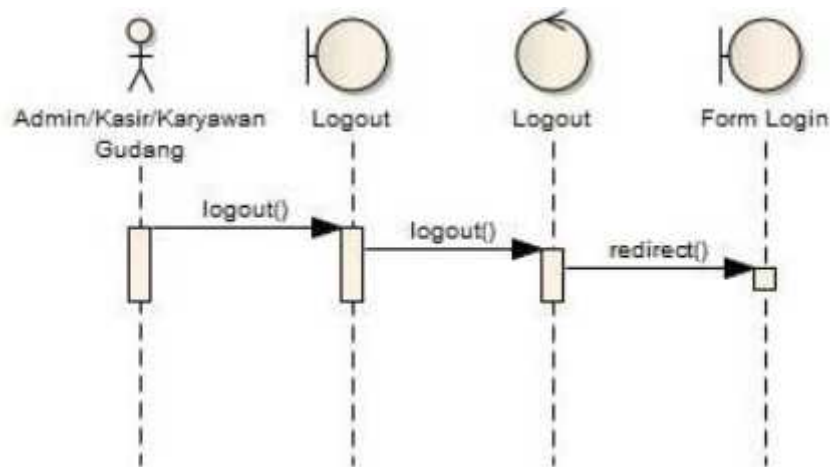
Activity diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis (Ade Hendini, 2016 : 109).



Gambar II.2. Activity Diagram
(Sumber : Ade Hendini, 2016 : 112)

II.7.3. Sequence Diagram

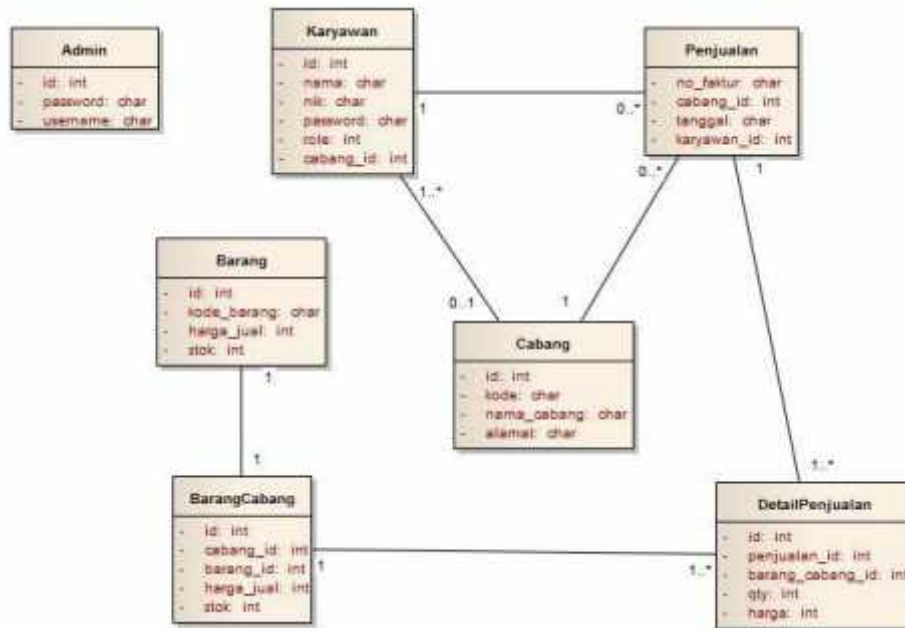
Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek (Ade Hendini, 2016 : 110).



Gambar II.3. Sequence Diagram
(Sumber : Ade Hendini, 2016 : 114)

II.7.4. Class Diagram

Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem. *Class* diagram juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan. *Class* diagram secara khas meliputi : Kelas (*Class*), Relasi *Associations*, *Generalitiation* dan *Aggregation*, atribut (*Attributes*), operasi (*operation/method*) dan *visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan *Multiplicity* atau *Cardinality* (Ade Hendini, 2016 : 111).



Gambar II.4. Class Diagram
 (Sumber : Ade Hendini, 2016 : 115)

II.8. PHP

Menurut Budi Raharjo, (2012:41-42) PHP adalah salah satu bahasa pemrograman skrip yang dirancang untuk membangun aplikasi web. Ketika dipanggil dari *web browser*, program yang ditulis dengan PHP akan di-parsing di dalam *web server* oleh interpreter PHP dan diterjemahkan ke dalam dokumen HTML, yang selanjutnya akan ditampilkan kembali ke *web browser*. Karena pemrosesan program PHP dilakukan di lingkungan web server, PHP dikatakan sebagai bahasa sisi *server (server-side)*. Oleh sebab itu, seperti yang telah dikemukakan sebelumnya, kode PHP tidak akan terlihat pada saat *user* memilih perintah “View Source” pada *web browser* yang mereka gunakan.

Berdasarkan uraian diatas dapat disimpulkan bahwa PHP adalah bahasa pemrograman script untuk membangun aplikasi web dimana membutuhkan *website server* dalam menjalankan aksinya (Ekkal Prasetyo, 2015 : 22).

PHP memiliki beberapa kelebihan yang tidak dimiliki oleh bahasa-bahasa sejenisnya, yaitu (Utami Dewi Widiyanti, 2012 : 58) :

1. Bahasa pemrograman PHP adalah sebuah bahasa *script* yang tidak melakukan sebuah kompilasi dalam penggunaannya.
2. *Web Server* yang mendukung PHP dapat ditemukan dimana - mana dari mulai IIS sampai dengan *apache*, dengan konfigurasi yang relatif mudah.
3. Dalam sisi pengembangan lebih mudah, karena banyaknya milis-milis dan *developer* yang siap membantu dalam pengembangan.
4. Dalam sisi pemahaman, PHP adalah bahasa *scripting* yang paling mudah karena referensi yang banyak.
5. PHP adalah bahasa *open source* yang dapat digunakan di berbagai mesin (*linux, unix, windows*) dan dapat dijalankan secara *runtime* melalui *console* serta juga dapat menjalankan perintah-perintah sistem.

II.9. MySQL

Menurut Budi, (2012:216) MySQL merupakan sistem *database* yang banyak digunakan untuk pengembangan aplikasi web. Alasannya mungkin karena

grafis, pengelolaan datanya sederhana, memiliki tingkat keamanan yang bagus, mudah diperoleh, dan lain-lain.

Menurut Abdul Kadir (2009:15) MySQL merupakan *software* yang tergolong *database server* dan bersifat *Open Source* menyatakan bahwa *software* ini dilengkapi dengan *source code* (kode yang dipakai untuk membuat MySQL), selain itu tentu saja bentuk *executable*-nya atau kode yang dapat dijalankan secara langsung dalam system operasi, dan bisa diperoleh dengan cara mengunduh di internet secara gratis. Hal menarik lainnya adalah MySQL juga bersifat *multiplatform*. MySQL dapat dijalankan pada berbagai sistem operasi.

MySQL bukan termasuk bahasa pemrograman. MySQL merupakan salah satu *database* populer dan mendunia. MySQL bekerja menggunakan *SQL Language (Structure Query Language)*. Itu dapat diartikan bahwa MySQL merupakan standar penggunaan *database* di dunia untuk pengolahan data

Kesimpulan pengertian MySQL merupakan sistem *database* yang banyak digunakan untuk pengembangan aplikasi web dan MySQL merupakan *software* yang tergolong *database server* dan bersifat *Open Source* menyatakan bahwa *software* ini dilengkapi dengan *source code*, tetapi MySQL bukan termasuk bahasa pemrograman.

Beberapa kelebihan yang dimiliki oleh MySQL (Ekkal Prasetyo, 2015: 22-23) :

1. Bersifat *Open Source*, yang dimiliki kemampuan untuk dapat dikembangkan lagi.

2. Menggunakan bahasa SQL (*Structure Query Language*), yang merupakan standar bahasa dunia dalam pengolahan data.
3. Super *performance* dan *reliable*, tidak bisa digunakan, pemrosesan *database*-nya sangat cepat dan stabil.
4. Sangat mudah dipelajari (*easy ofuse*).
5. Memiliki dukungan *support (group)* pengguna MySQL.
6. Mampu lintas *Platform*, dapat berjalan di berbagai sistem operasi.
7. *Multiuser*, di mana MySQL dapat digunakan oleh beberapa *user* dalam waktu yang bersamaan tanpa mengalami konflik.