

## **BAB II**

### **LANDASAN TEORI**

#### **2.1. Teori – Teori Umum**

##### **2.1.1. Sistem**

###### **2.1.1.1. Pengertian Sistem**

Sistem merupakan hal penting yang sangat dibutuhkan oleh setiap perusahaan. Sistem sendiri dibuat untuk memudahkan perusahaan dalam melakukan aktivitas dan membuat kinerja perusahaan menjadi lebih efektif.

Menurut Satzinger, Jackson, dan Burd (2012, p5), sistem adalah sekumpulan kegiatan – kegiatan yang memungkinkan pengguna untuk mendefinisikan dan mendeskripsikan secara jelas di dalam memecahkan suatu permasalahan atas kebutuhan pengguna sistem tersebut. Sedangkan definisi sistem menurut O'Brien (2010, p26) sistem adalah kumpulan dari komponen yang saling terkait dengan ruang lingkup yang jelas saling bekerja sama untuk mencapai tujuan dengan cara menerima input dan menghasilkan *output* kedalam proses transformasi yang terorganisir.

Dari definisi diatas dapat disimpulkan bahwa sistem adalah kumpulan komponen yang saling terkait satu sama lain yang saling bekerja sama untuk menghasilkan tujuan yang terdapat dalam perusahaan.

###### **2.1.1.2. Pengertian Informasi**

Informasi mempunyai peran penting dalam perusahaan. Tanpa adanya informasi yang baik perusahaan tidak dapat bersaing dengan pesaing yang memiliki informasi yang lebih baik. Salah satu fungsi dari informasi adalah untuk mengambil suatu keputusan.

Pengertian informasi menurut Rainer dan Cegielski (2011, p10) Informasi adalah data yang sudah diolah

sehingga mempunyai arti dan nilai bagi perusahaan. Sedangkan menurut O'Brien (2010, p34) Informasi adalah data yang telah dikonfersi atau dirubah menjadi konteks yang bermakna dan berguna untuk pengguna yang spesifik.

Dari definisi diatas dapat disimpulkan bahwa informasi adalah suatu data yang telah dirubah sehingga memiliki nilai yang berguna.

#### **2.1.1.3. Pengertian Data**

Menurut O'Brien (2010, p34) Data adalah penelitian atau fakta mentah yang biasanya berkaitan dengan fenomena fisik atau transaksi bisnis. Lebih rincinya, data adalah pengukuran yang objektif dari atribut (karakteristik) dan entitas (manusia, tempat, barang dan kejadian). Sedangkan menurut Rainer dan Cegielski (2011, p10), Data adalah deskripsi elementer tentang suatu kejadian dan transaksi-transaksi yang di catat, diklarifikasi dan di simpan tetapi belum diolah sehingga memberikan suatu pesan yang spesifik. Dari definisi – definisi diatas dapat disimpulkan bahwa data adalah deskripsi fakta mengenai fenomena atau kejadian dan hasil pengukuran objektif dari atribut dan entitas.

#### **2.1.1.4. Pengertian Sistem Informasi**

Menurut Satzinger, Jackson, dan Burd (2012, p4), Sistem informasi adalah satu set komponen komputer yang mengumpulkan, memproses, menyimpan, dan menyediakan keluaran informasi yang diperlukan untuk menyelesaikan tugas - tugas bisnis. Sedangkan menurut O'Brien (2010, p4) Sistem Informasi adalah kombinasi dari orang, *hardware*, *software*, *computer network* dan *data communication* yang mengumpulkan, mengubah, dan menyebarkan informasi didalam suatu bentuk organisasi.

Dari definisi diatas dapat disimpulkan bahwa sistem informasi adalah kumpulan dari komponen seperti orang,

*hardware, software, data network, dan data communication* yang saling terkait satu sama lain yang bertugas untuk mengumpulkan, memproses dan menyediakan informasi didalam organisasi yang diperlukan untuk menyelesaikan tugas bisnis.

## **2.1.2. Internet**

### **2.1.2.1. Pengertian Internet**

Menurut Turban (2006, p50), *Internet* adalah sistem jaringan komputer dunia yang terdiri dari jaringan berupa jaringan publik, jaringan kerjasama dan jaringan tersendiri.

### **2.1.2.2. Pengertian World Wide Web**

Menurut Turban (2006, p50), *World Wide Web* adalah sebuah aplikasi yang digunakan memindahkan fungsi pada *internet* dan secara universal sudah memiliki standar dalam penyimpanan, penarikan, memformat dan menampilkan informasi via arsitektur *client/server*.

### **2.1.2.3. Pengertian Website**

Menurut Williams dan Sawyer (2011, p65) *website* adalah suatu lokasi komputer tertentu pada *web* yang memiliki alamat yang unik yang berguna untuk menyajikan berbagai informasi.

### **2.1.2.4. Pengertian Web-Based System**

Menurut Turban (2006, p50), *web-based system* adalah suatu aplikasi atau layanan yang berada pada *server* yang dapat diakses dengan menggunakan *web browser* dan oleh karena itu dapat diakses dimana saja melalui via *internet*.

### 2.1.3. Basis Data

#### 2.1.3.1. Pengertian Basis Data

Menurut Connolly dan Begg (2015, p.63), basis data adalah sekumpulan data yang saling berkaitan secara logika dan deskripsi dari data yang dirancang untuk memenuhi kebutuhan informasi suatu organisasi.

Sebagai satu kesatuan istilah, Basis Data (*Database*) sendiri dapat didefinisikan dalam sejumlah sudut pandang, seperti:

- Himpunan kelompok data (arsip) yang saling berhubungan yang diorganisasi sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah.
- Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (redundansi) yang tidak perlu, untuk memenuhi berbagai kebutuhan.
- Kumpulan *file* atau tabel atau arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronik.

#### 2.1.3.2. Pengertian *Database Management System*

Menurut Connolly dan Begg (2015, p.64), *database management systems* (DBMS) adalah suatu sistem perangkat lunak yang memungkinkan *user* untuk mendefinisikan, membuat, memelihara, dan mengontrol akses ke *database*. DBMS juga merupakan perangkat lunak yang berinteraksi dengan program aplikasi pengguna dan *database* itu sendiri. Secara khusus, suatu DBMS menyediakan fasilitas-fasilitas sebagai berikut:

##### 1. *Data Definition Language (DDL)*

Merupakan bahasa yang memungkinkan pengelola atau pengguna basis data untuk menjelaskan dan memberi nama sebuah entitas,

atribut, dan hubungan yang dibutuhkan oleh aplikasi. DDL juga memberikan batasan integritas dan keamanan (Connolly dan Begg, 2015, p90). Beberapa perintah dasar yang termasuk didalam DDL yaitu:

- **Create:** Digunakan untuk membuat *database* baru, tabel baru, *view* baru, dan kolom.
- **Alter:** Digunakan untuk mengubah struktur tabel yang telah dibuat yang mencakup pergantian nama tabel, menambah kolom, mengubah kolom, menghapus kolom, maupun memberikan atribut pada kolom.
- **Drop:** Digunakan untuk menghapus *database* dan tabel.

## 2. **Data Manipulation Language (DML)**

Adalah bahasa yang menyediakan sekumpulan operasi untuk mendukung manipulasi data dasar yang ada di dalam basis data (Connolly dan Begg, 2015, p90). DML dapat menyisipkan data baru, mengubah data yang ada, mengambil data, dan menghapus data yang ada di dalam basis data. Perintah SQL yang termasuk DML adalah:

- **Update:** Untuk memperbaharui data lama menjadi data yang baru, jika ada data yang salah maupun kurang maka dapat diubah isi data dengan menggunakan perintah UPDATE.
- **Insert:** Untuk menyisipkan data baru kedalam tabel setelah *database* dan tabel dibuat.
- **Select:** Untuk mengambil data atau menampilkan data dari suatu tabel atau beberapa tabel dalam relasi.

- **Delete:** Untuk menghapus data dari tabel. Biasa data yang dihapus adalah data yang tidak diperlukan lagi.

3. **Menyediakan *access control* pada basis data, sebagai contohnya:**

- **Security System**, yang mencegah akses illegal user ke database.
- **Integrity System**, yang menjaga konsistensi data yang disimpan.
- **Concurrency Control**, memungkinkan pembagian akses terhadap basis data.
- **Recovery Control System**, mengembalikan basis data pada keadaan sebelumnya yang konsisten apabila terjadi kegagalan pada perangkat keras atau perangkat lunak.
- **User Accessible Catalog**, berisi deskripsi data dalam basis data.

## 2.2. Teori – Teori Khusus

### 2.2.1. Metodologi *Scrum*

Menurut Kenneth S. Rubin (2013, p1 – 3) *Scrum* merupakan sebuah pendekatan dari *agile* yang dilakukan untuk mengembangkan produk serta layanan agar lebih inovatif.

Pada tahun 1986 Harvard Business Review mengangkat kembali *scrum* dalam sebuah artikel kemudian *scrum* dijuluki “*The new product Development Game*” (Takeuchi and Nonaka, 1986). *Scrum* menekankan pada penggunaan *scalable*, pentingnya penggunaan *scrum* dalam suatu tim yang diorganisir secara perorangan setelah itu diuraikan oleh masing – masing manajemen didalam setiap proses dalam tim tersebut.

Pola pengembangan *software* dengan menggunakan metode *scrum* meliputi:

## 1. *Product Backlog*

*Product backlog* merupakan kebutuhan proyek yang diutamakan atau fitur yang menyediakan nilai bisnis untuk pengguna. Fitur dapat ditambah ke *backlog* setiap saat yang disebut sebagai *Product Backlog Item*. *Product Backlog* memiliki *product owner* yang bertugas sebagai pengawas dalam *product backlog* untuk melakukan penambahan ataupun pengurangan yang dilakukan secara terus menerus dalam *product backlog*.

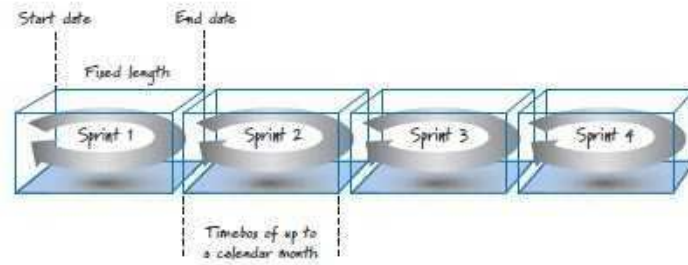


**Gambar 2.1** *Product Backlog*

Sumber: Kenneth S. Rubin (2013, p19)

## 2. *Sprints*

Dalam *scrum*, pekerjaan dilakukan secara berulang atau seperti siklus dalam suatu waktu disebut *sprints*. *Sprints* merupakan sebuah *timeboxed* yang memiliki tanggal mulai dan berakhirnya suatu pekerjaan. *Sprint* baru akan langsung mengikuti proses penyelesaian *sprint* sebelumnya.

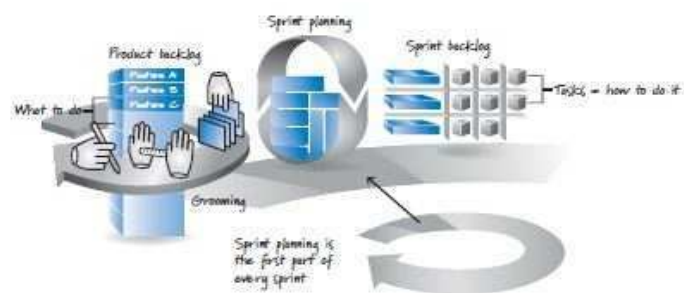


**Gambar 2.2** Tahapan *Sprint*

Sumber: Kenneth S. Rubin (2013, p21)

### 3. *Sprint Planning*

Didalam *sprint* terdapat *Sprint Planning* yang kegunaannya untuk menjamin pekerjaan dalam hitungan minggu ataupun bulan sesuai kesepakatan dalam satu tim yang bertujuan mendefinisikan secara jelas apa yang harus dicapai. Unit kerja yang dibutuhkan untuk mencapai kebutuhan yang telah ditetapkan dan harus sesuai dengan waktu yang telah ditetapkan. Batas waktu *sprints* bervariasi sesuai dengan *backlog* yang digunakan dalam *sprints*.



**Gambar 2.3** *Sprint Planning*

Sumber: Kenneth S. Rubin (2013, p21)

#### 2.2.2. Gantt Chart

Menurut Jack T. Marchewka (2014, p151) Gantt Chart adalah representasi visual yang membandingkan sebuah proyek kegiatan yang direncanakan dengan kemajuan aktual dari waktu ke waktu.



### 2.2.3. *E-Learning*

#### 2.2.3.1. Pengertian *E-Learning*

Menurut Horton (2012, p1) *E-learning* adalah "Penggunaan teknologi elektronik untuk menciptakan pengalaman belajar". Dari pengertian tersebut, definisi *e-learning* dapat dikembangkan pengertiannya menjadi lebih luas tentang bagaimana pengalaman tersebut diformulasi, diorganisir, dan diciptakan.

Menurut effendi dan Zhuang (2005, p6), *E-Learning* adalah semua kegiatan pelatihan yang menggunakan media elektronik atau teknologi informasi.

Menurut Mayerová dan Rosická (2014, p4) Program *E-Learning* atau *E-Education* dapat menawarkan konten yang lebih luas dari sebuah topik daripada pendidikan konvensional.

Dari definisi diatas dapat disimpulkan bahwa *E-learning* merupakan suatu kegiatan pembelajaran dengan memanfaatkan media elektronik berupa komputer sebagai media perantaranya.

#### 2.2.3.2. Tipe – Tipe *E-Learning*

Menurut effendi dan zhuang (2005, p7), *E-learning* terdapat dua tipe, yaitu sebagai berikut:

- ***Synchronous Learning***

*Synchronous* berarti “pada waktu yang sama”, *Synchronous Learning* adalah tipe pelatihan, dimana proses pembelajaran terjadi pada saat yang sama ketika pengajar sedang mengajar dan murid sedang belajar. Hal tersebut memungkinkan interaksi langsung antara guru dan murid, baik melalui internet maupun intranet. Pelatihan *E-Learning Synchronous* lebih banyak digunakan pada seminar yang pesertanya berasal

dari beberapa negara. penggunaan tersebut sering juga dinamakan *web conference* dan sering digunakan dikelas atau kuliah universitas *online white board*.

- ***Asynchronous Learning***

*Asynchronous Training* berarti “tindak pada waktu yang bersamaan”. Seseorang dapat mengambil pelatihan pada waktu yang berbeda dengan pengajar memberikan pelatihan. Pelatihan ini lebih populer didunia *E-Learning* karena memberikan keuntungan lebih bagi para peserta pelatihan, dapat mengakses pelatihan kapanpun dan dimanapun.

Pelatihan berupa paket pembelajaran yang dapat dijalankan dikomputer manapun dan tidak melibat interaksi dengan pengajar atau pelajar lain. Oleh karena itu, pelajar dapat memulai pelajaran dan menyelesaikannya setiap saat Paket pelajaran berbentuk bacaan dengan animasi, simulasi, permainan yang edukatif, maupun latihan atau tes dengan jawabannya.

### **2.2.3.3. Fungsi *E-Learning***

Menurut Chandrawati (2012, p4) Setidaknya ada 3 (tiga) fungsi pembelajaran elektronik terhadap kegiatan pembelajaran di dalam kelas (*classroom instruction*), yaitu sebagai suplemen yang sifatnya pilihan/opsional, pelengkap (komplemen), atau pengganti (substitusi) (Siahaan, 2002).

- **Suplemen (Tambahan)**, dikatakan berfungsi sebagai suplemen (tambahan), apabila peserta didik mempunyai kebebasan memilih, apakah akan memanfaatkan materi pembelajaran elektronik atau tidak. Dalam hal ini, tidak ada kewajiban/ keharusan bagi peserta didik untuk mengakses materi pembelajaran elektronik.

Sekalipun sifatnya opsional, peserta didik yang memanfaatkannya tentu akan memiliki tambahan pengetahuan atau wawasan.

- **Komplemen (Pelengkap)**, dikatakan berfungsi sebagai komplemen (pelengkap) apabila materi pembelajaran elektronik diprogramkan untuk melengkapi materi pembelajaran yang diterima siswa di dalam kelas (Lewis, 2002). Sebagai komplemen berarti materi pembelajaran elektronik diprogramkan untuk menjadi materi *reinforcement* (pengayaan) atau remedial bagi peserta didik di dalam mengikuti kegiatan pembelajaran konvensional. Materi pembelajaran elektronik dikatakan sebagai *enrichment*, apabila kepada peserta didik yang dapat dengan cepat menguasai/ memahami materi pelajaran yang disampaikan guru secara tatap muka (*fast learners*) diberikan kesempatan untuk mengakses materi pembelajaran elektronik yang memang secara khusus dikembangkan untuk mereka. Tujuannya agar semakin memantapkan tingkat penguasaan peserta didik terhadap materi pelajaran yang disajikan guru di dalam kelas. Dikatakan sebagai program remedial, apabila kepada peserta didik yang mengalami kesulitan memahami materi pelajaran yang disajikan guru secara tatap muka di kelas (*slow learners*) diberikan kesempatan untuk memanfaatkan materi pembelajaran elektronik yang memang secara khusus dirancang untuk mereka. Tujuannya agar peserta didik semakin lebih mudah memahami materi pelajaran yang disajikan guru di kelas.
- **Substitusi (Pengganti)**, ada 3 alternatif model

kegiatan pembelajaran yang dapat dipilih peserta didik, yaitu: (1) sepenuhnya secara tatap muka (konvensional), (2) sebagian secara tatap muka dan sebagian lagi melalui internet, atau bahkan (3) sepenuhnya melalui internet.

#### **2.2.3.4. Learning Management System**

Menurut Mahnegar (2012, p5) Learning Management System adalah sebuah perangkat lunak yang digunakan untuk menyampaikan, melacak, dan mengelola pelatihan/ pendidikan. Jangkauan Learning Management System memungkinkan pengguna untuk mengelola bahan ajar dari pelatihan/ pendidikan dengan menggunakan perangkat lunak untuk mendistribusikannya melalui internet dan menawarkan fitur kolaborasi secara *online*.

#### **2.2.3.5. Content Management System**

Menurut I Made Gunardi dan Lirva32 (2007, p3) Content Management System didefinisikan sebagai pengelolaan isi atau konten. Bila dikaitkan dengan *web*, maka Content Management System dapat didefinisikan sebagai *software* yang mampu mengelola isi atau konten dari sebuah *website* seperti melakukan publikasi, *edit* ataupun menghapus sebuah konten.

#### **2.2.4. Pendaftaran**

Menurut Kamus Besar Bahasa Indonesia (2008, p285) Pendaftaran adalah sebuah proses, cara, mendaftarkan, pencatatan nama dan alamat serta identitas diri ke dalam daftar. Menurut kamus pendaftaran itu sendiri adalah sebuah proses kegiatan yang melakukan pencatatan tentang identitas diri secara pribadi kedalam sebuah daftar nama atau dokumen yang disimpan dan diabadikan.

### 2.2.5. Pengertian *Dashboard*

Menurut Yigitbasioglu dan Velcu (2012, p4) *Dashboard* dapat didefinisikan sebagai "Manajemen kinerja visual dan interaktif" dengan menampilkan pada satu layar informasi yang berisi yang penting dan dibutuhkan untuk mengidentifikasi, mengeksplorasi, dan berkomunikasi masalah daerah yang membutuhkan tindakan korektif. *Dashboard* menggabungkan fitur visual dan fungsional, yang dalam kombinasi dapat membantu meningkatkan kognisi dan interpretasi.

### 2.2.6. Object Oriented Analysis and Design

Menurut Satzinger, Jackson & Burd (2012, p241), *Object-Oriented Analysis* (OOA) mendefinisikan benda – benda yang melakukan pekerjaan dan menentukan interaksi pengguna yang diperlukan untuk menyelesaikan tugas.

Menurut Satzinger, Jackson & Burd (2012, p241), *Object-Oriented Design* (OOD), mendefinisikan semua jenis benda tambahan yang diperlukan untuk berkomunikasi dengan orang dan perangkat dalam sistem, serta menunjukkan interaksi objek untuk menyelesaikan tugas, mendefinisikan setiap jenis objek sehingga dapat diimplementasikan dengan bahasa atau lingkungan tertentu.

### 2.2.7. UML (Unified Modeling Language)

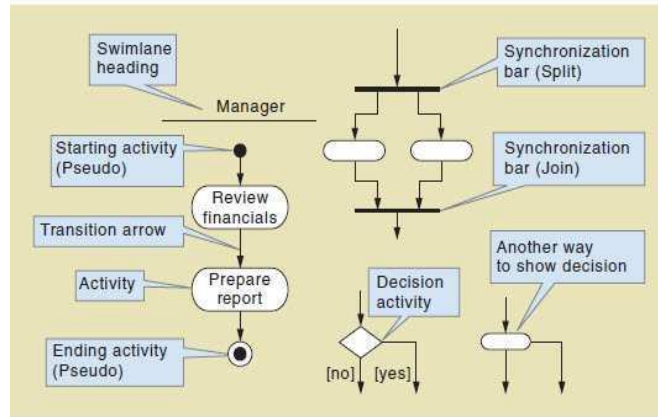
Menurut Satzinger, Jackson and Burd (2012, p46), UML merupakan sebuah kumpulan *standart model* dan notasi yang telah diakui oleh *Object Management Group* (OMG), sebuah organisasi yang memberikan standart untuk system development. Secara umum, UML digunakan untuk memvisualisasi, menspesifikasi, membangun serta mendokumentasikan sebuah sistem *software*. Dengan menggunakan UML, sebuah tim *developer* dapat mendeskripsikan dan mengerti diagram – diagram spesifik yang digunakan di dalam sebuah proyek.

#### 2.2.7.1. Activity Diagram

Menurut Satzinger, Jackson, dan Burd (2012, p57), *activity diagram* merupakan *diagram* yang menunjukkan alur kerja atau aktivitas user secara berurutan. Langkah

awal dalam membuat *activity diagram* adalah mengidentifikasi setiap *agent* untuk membuat *swimlane*. Adapun hal lain yang perlu diperhatikan dalam membuat *activity diagram* adalah penggunaan *decision symbol* untuk menggambarkan suatu keadaan satu alur atau alur lainnya tetapi bukan keduanya. Penggunaan *synchronization bar* untuk alur yang *parallel* – keadaan dimana kedua alur dilakukan. Berikut merupakan penjelasan notasi dan fungsi kegunaan *activity diagram* adalah sebagai berikut:

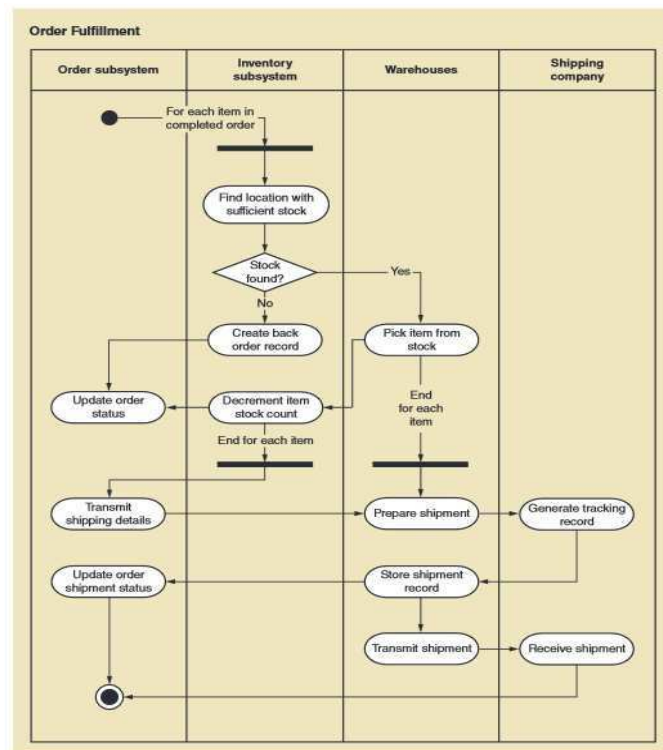
- ***Swimlane***, merupakan suatu bentuk persegi yang merepresentasikan aktivitas – aktivitas yang diselesaikan setiap agen.
- ***Starting Activity (Pseudo)***, merupakan notasi yang menunjukkan awal dimulainya suatu aktivitas.
- ***Transition Arrow***, merupakan notasi yang berupa anak panah yang mendeskripsikan arah perpindahan suatu aktivitas.
- ***Activity***, merupakan notasi yang mendeskripsikan aktivitas – aktivitas.
- ***Synchronization Bar***, merupakan notasi yang berfungsi memisahkan (*split*) atau menyatukan (*join*) urutan jalur aktivitas.
- ***Decision Activity***, merupakan notasi yang mendeskripsikan kondisi dari suatu aktivitas.
- ***Ending Activity (Pseudo)***, merupakan notasi yang menunjukkan diakhirinya suatu aktivitas.



**Gambar 2.4** Notasi *Activity Diagram*

Sumber: Satzinger, Jackson, dan Burd (2012, p58)

Berikut merupakan contoh penggambaran proses dengan menggunakan *activity diagram* adalah sebagai berikut:



**Gambar 2.5** Contoh *Activity Diagram*

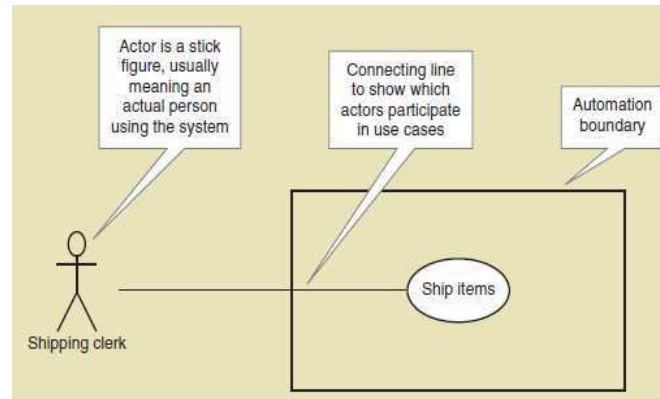
Sumber: Satzinger, Jackson, dan Burd (2012, p59)

### 2.2.7.2. Use Case Diagram

Menurut Satzinger, Jackson, dan Burd (2012, p78), *use case diagram* adalah aktivitas yang dilakukan oleh sistem berupa respon terhadap permintaan pengguna serta hubungan antara aktor – aktor pengguna tersebut di dalam sistem. Pengguna sistem dalam *use case diagram* disebut sebagai aktor. Aktor selalu berada di luar batas otomisasi sistem tetapi dapat menjadi bagian dari bagian pengguna sistem. Berikut merupakan penjelasan dari elemen – elemen *use case diagram* adalah sebagai berikut:

- **Actor** merupakan pelaku yang menggunakan sistem. Biasanya direpresentasikan dengan bentuk stick figure dengan label nama sesuai dengan *role actor* tersebut.
- **Use Case** merupakan aktivitas yang sistem lakukan dalam merespon permintaan dari pengguna. Umumnya, direpresentasikan dalam bentuk oval dengan nama *use case* itu sendiri tertulis di dalamnya.
- **Relationship** merupakan hubungan antara sistem dengan aktor yang direpresentasikan dengan sebuah garis antara *use cases* dengan aktor – aktor yang terlibat.
- **Boundary** merupakan batasan yang memisahkan antara aplikasi dengan para pengguna yang mengoperasikan aplikasi tersebut. Umumnya, direpresentasikan dengan garis berbentuk kotak yang melingkupi seluruh *use case*.

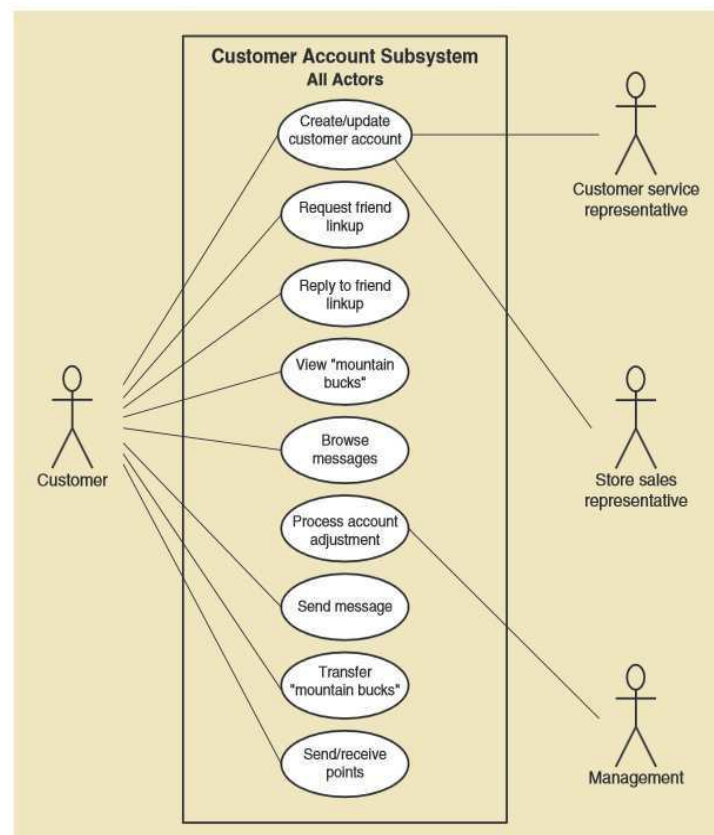




**Gambar 2.6** Elemen *Use Case Diagram*

Sumber: Satzinger, Jackson, dan Burd (2012, p81)

Berikut merupakan contoh penggambaran relasi antara *actor* dan *use case* dengan menggunakan *use case diagram* adalah sebagai berikut:

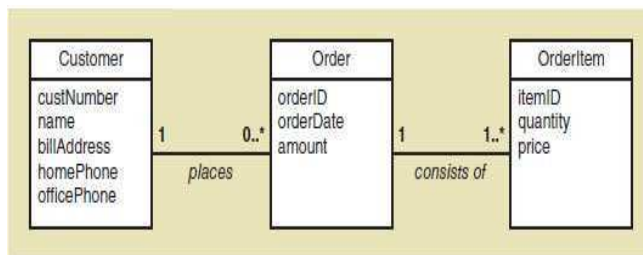


**Gambar 2.7** Contoh *Use Case Diagram*

Sumber: Satzinger, Jackson, dan Burd (2012, p82)

### 2.2.7.3. Class Diagram

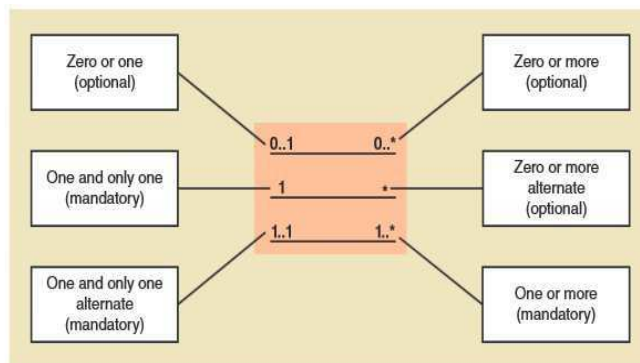
Menurut Satzinger, Jackson, dan Burd (2012, p101), *domain class diagram* adalah sebuah diagram UML yang merepresentasikan kelas – kelas *domain*, atribut, pekerjaan pengguna serta hubungan antar kelas tersebut. Pada *class diagram*, bentuk kotak menggambarkan *classes* dan garis menunjukkan hubungan antar *class* tersebut. *Domain class diagram* digunakan untuk memahami hubungan antar *class* yang terdiri dari beberapa objek di dalam pengembangan dan perancangan sistem nantinya.



**Gambar 2.8** Contoh *Domain Class Diagram*

Sumber: Satzinger, Jackson, dan Burd (2012, p102)

Berikut merupakan gambar yang menjelaskan tentang *multiplicity* pada *class diagram* adalah sebagai berikut:



**Gambar 2.9** *Multiplicity Class Diagram*

Sumber: Satzinger, Jackson, dan Burd (2012, p102)

### 2.2.7.4. Use Case Description

Menurut Satzinger, Jackson, dan Burd (2012, p121), *use case description* merupakan penjelasan terperinci

mengenai proses dari suatu *use case* atau bisa disebut juga sebagai daftar kasus penggunaan diagram *use case* yang memberikan gambaran dari semua penggunaan kasus untuk sistem. Menurut Satzinger, Jackson dan Burd (2012, p122), *fully developed use case description* adalah metode paling formal untuk mendokumentasikan *use case* karena menjelaskan secara rinci setiap proses dalam *use case diagram*. Dalam sebuah *fully developed use case description* terbagai atas 11 bagian kompartemen, yaitu:

1. ***Use Case Name*** merupakan nama dari *use case* yang akan diidentifikasi. Umumnya, *use case name* ini terletak pada kompartemen pertama dari *use case description*.
2. ***Scenario*** merupakan skenario tentang apa yang terjadi di dalam *use case* yang sedang diidentifikasi. Skenario ini terletak pada kompartemen kedua dari *use case description*.
3. ***Triggering Event*** menjelaskan tentang suatu *event* yang dapat memicu terjadinya *use case* tersebut. terletak pada kompartemen ketiga dari *use case description*.
4. ***Brief Description*** berisi tentang deskripsi singkat mengenai *use case* yang sedang dideskripsikan. *Brief description* ini terletak pada kompartemen keempat.
5. ***Actors*** merupakan pelaku dari *use case* tersebut. Umumnya, kompartemen aktor dapat diisi lebih dari 1 aktor.
6. ***Related Use Cases*** mengidentifikasi tentang *use case* lainnya yang berelasi dengan *use case* yang sedang dideskripsikan. Hubungan antara *setiap use case* ini dapat membantu dalam mendokumentasi semua aspek dari *user requirements*.

7. **Stakeholders** mengidentifikasi kelompok yang tertarik terhadap suatu hasil *use case* walaupun mereka tidak terlibat di dalam proses *use case* tersebut. Terletak pada kompartemen ketujuh.
8. **Preconditions** mengidentifikasi tentang kondisi apa saja dari sebuah sistem yang dapat membuat sebuah *use case* dapat berjalan. Hal ini meliputi objek apa saja yang harus ada, informasi apa saja yang tersedia sampai dengan kondisi dari aktor tersebut.
9. **Postconditions** mengidentifikasi apa saja hasil yang dapat dihasilkan dari penyelesaian sebuah *use case*. Sebuah *postcondition* dapat menghasilkan sebuah basis untuk suatu *test case* serta mengidentifikasi objek penting apa saja yang terlibat di dalam *use case* yang dibutuhkan untuk merancang sebuah sistem.
10. **Flow of activities** menjelaskan secara mendetail tentang aktivitas dari *use case* tersebut. Umumnya, kompartemen ini terbagi menjadi dua, dimana kompartemen A menggambarkan langkah – langkah aktivitas yang dijalankan oleh pengguna dan kompartemen B menggambarkan langkah – langkah atau respon dari sistem. Angka yang ada pada setiap baris mengidentifikasikan tentang langkah – langkah yang telah diambil.
11. **Exception Conditions** berisi tentang aktivitas alternatif/ pengecualian di dalam sebuah *flow* aktivitas.

<b>Use case name:</b>	Create customer account.	
<b>Scenario:</b>	Create online customer account.	
<b>Triggering event:</b>	New customer wants to set up account online.	
<b>Brief description:</b>	Online customer creates customer account by entering basic information and then following up with one or more addresses and a credit or debit card.	
<b>Actors:</b>	Customer.	
<b>Related use cases:</b>	Might be invoked by the <i>Check out shopping cart</i> use case.	
<b>Stakeholders:</b>	Accounting, Marketing, Sales.	
<b>Preconditions:</b>	Customer account subsystem must be available. Credit/debit authorization services must be available.	
<b>Postconditions:</b>	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.	
<b>Flow of activities:</b>	<b>Actor</b>	<b>System</b>
	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.
<b>Exception conditions:</b>	1.1 Basic customer data are incomplete. 2.1 The address isn't valid. 3.2 Credit/debit information isn't valid.	

**Gambar 2.10** Contoh *Fully Developed Use Case Description*

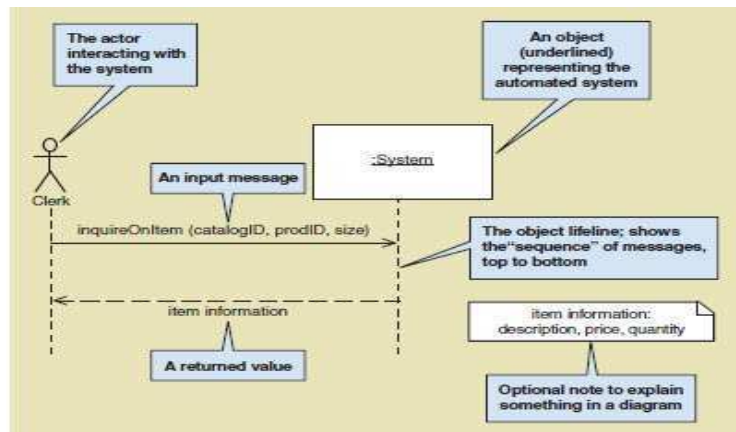
Sumber: Satzinger, Jackson, dan Burd (2012, p123)

#### 2.2.7.5. *System Sequence Diagram*

Menurut Satzinger, Jackson, dan Burd (2012, p126), *system sequence diagram* merupakan *diagram* yang menunjukkan urutan pesan antara aktor eksternal dan internal sistem di dalam *use case* atau *scenario* yang sudah dirancang sebelumnya. Berikut merupakan penjelasan dari komponen – komponen *system sequence diagram* adalah sebagai berikut:

1. **Actor** merupakan komponen yang berinteraksi dengan sistem. Umumnya digambarkan sebagai *stick figure*.
2. **:System** merupakan salah satu objek di dalam diagram yang merepresentasikan sebuah sistem.

3. ***Lifeline*** merupakan sebuah ekstensi dari objek, baik aktor maupun sebuah objek, di dalam suatu *use case*. *Lifeline* dibuat sebagai indikasi waktu suatu alur pesan yang dikirim dan diterima oleh aktor dan objek.
4. ***Input message*** merupakan data yang dimasukkan oleh aktor ke dalam suatu sistem. *Input message* dilambangkan dengan tanda panah dengan garis lurus yang menuju ke sistem.
5. ***Returned Value*** merupakan nilai atau data yang dikembalikan oleh sistem kepada objek atau aktor. Biasanya dilambangkan dengan tanda panah dengan garis putus-putus yang menuju kepada objek.
6. ***Frame*** merupakan sebuah kontainer yang melakukan kontrol terhadap *behaviour* dari *message* yang ada di dalam kontainer. Umumnya, kontainer ini berbentuk kotak dan memiliki jenis-jenis seperti *loop condition*, *true/false condition*, *alternative condition*, dan *optional condition*.
7. ***Activation Lifeline*** merupakan sebuah representasi waktu dimana *method* dari sebuah objek sedang aktif dan melakukan eksekusi.

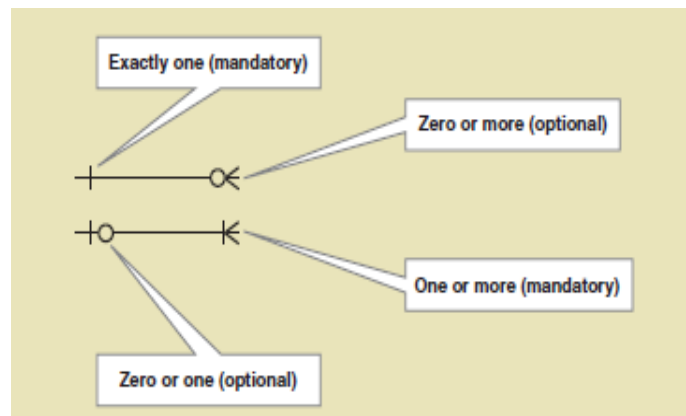


**Gambar 2.11** Notasi *System Sequence Diagram*

Sumber: Satzinger, Jackson, dan Burd (2012, p102)

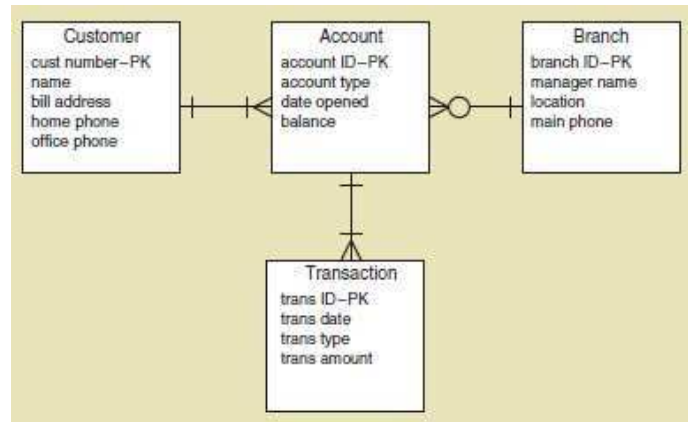
#### 2.2.7.6. *Entity Relationship Diagram*

Menurut Satzinger, Jackson dan Burd (2012, p98), *Entity relationship diagram* (ERD) merupakan sebuah *diagram* yang terdiri dari data entitas. Sebuah kotak di dalam ERD merupakan sebuah representasi dari data entitas, sedangkan garis yang menghubungkan kotak tersebut merupakan representasi dari hubungan entitas data. ERD memiliki 4 hubungan cardinality di antara setiap entitas, yaitu *exactly one*, *zero or more*, *zero or one*, dan *one or more*.



**Gambar 2.12** *Cardinality ERD*

Sumber: Satzinger, Jackson, dan Burd (2012, p99)



**Gambar 2.13** Contoh ERD

Sumber: Satzinger, Jackson, dan Burd (2012, p100)

## 2.2.8. Software and Programming Tools

### 2.2.8.1. XAMPP

Menurut Bunafit Nugroho (2008, p2), XAMPP merupakan paket PHP berbasis *open source* yang dikembangkan oleh sebuah komunitas *open source* dengan menggunakan XAMPP, dan tidak membingungkan untuk melakukan penginstalan program lain, karena semua kebutuhan telah disediakan oleh XAMPP.

XAMPP adalah perangkat lunak gratis yang mendukung banyak sistem operasi yang merupakan kompilasi dari beberapa perangkat lunak. XAMPP dikembangkan oleh sebuah tim proyek bernama *Apache Friends*.

Fungsinya adalah sebagai server yang berdiri sendiri (*localhost*), yang terdiri atas program *Apache HTTP Server*, *MySQL database*, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl. Nama XAMPP merupakan singkatan dari X berarti mendukung 4 sistem operasi, *Apache*, *MySQL*, *PHP*, dan *Perl*.



### 2.2.8.2. PHP

Menurut Raharjo dan Heryanto (2012, p4), PHP adalah salah satu bahasa pemrograman *script* 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*).

Menurut Loka Dwiartara (2010, p4) yang menjadikan PHP berbeda dengan HTML adalah proses dari PHP itu sendiri. HTML merupakan bahasa statis yang apabila kita ingin merubah konten/ isinya maka yang harus dilakukan pertama kali adalah membuka *filenya* terlebih dahulu, kemudian menambahkan isi kedalam *file* tersebut. Beda halnya dengan PHP, di bahasa PHP seorang pengguna tidak berhubungan langsung dengan *scriptnya*, sehingga seorang pemula sekalipun dapat menggunakan aplikasi seperti itu. Keunggulan PHP antara lain:

1. Gratis

PHP berkembang pesat sehingga jutaan domain menggunakan PHP karena menggunakan Bahasa ini tidak dipungut biaya.

2. *Cross Platform*

Artinya dapat digunakan diberbagai sistem operasi, mulai dari *linux*, *windows*, *mac os*, dan *os* yang lainnya.

3. Mendukung banyak basis data

PHP telah banyak mendukung basis data, oleh karena itu banyak *developer web* yang menggunakan PHP.

#### 4. *On the fly*

PHP telah mendukung *on the fly*, artinya dengan PHP pengguna dapat membuat dokumen berupa teks, *word*, *excel*, PDF, menciptakan *image* dan *flash*, serta menciptakan *file – file* seperti zip, XML, dan lainnya.

#### 2.2.8.2.1 Codeigniter

Menurut Ibnu Daqiqil (2011, p4) CodeIgniter adalah sebuah *web application framework* yang bersifat *open source* digunakan untuk membangun aplikasi php dinamis. Tujuan utama pengembangan codeigniter adalah untuk membantu *developer* untuk mengerjakan aplikasi lebih cepat daripada menulis semua code dari awal. Codeigniter menyediakan berbagai macam *library* yang dapat mempermudah dalam pengembangan. CodeIgniter sendiri dibangun menggunakan konsep *Model-View-Controller development* pattern. CodeIgniter sendiri merupakan salah satu *framework* tercepat dibandingkan dengan *framework* lainnya.

#### 2.2.8.3. Konektivitas PHP dan MySQL

Menurut Loka Dwiartara (2010: 6), sebelum *script* PHP melakukan pengaksesan data ke *server database* MySQL, *script* tersebut perlu dihubungkan dengan *web server* kemudian mengkoneksikannya pada *server* MySQL. Berikut adalah *syntax* umum untuk mengkoneksikan PHP dan MySQL:

```
mysql_connect(“host”, “user_mysql”, “password_mysql”);
```

*Syntax* di atas dapat dijelaskan sebagai berikut:

1. *Host*: merupakan alamat *server* komputer yang menyediakan basis data MySQL
2. *User\_MySQL*: merupakan pengguna MySQL yang dimiliki dalam *server* MySQL
3. *Password\_MySQL*: merupakan *password* pengguna MySQL, yang dimiliki

#### 2.2.8.4. Structure Query Language

Menurut Connolly dan Begg (2015, p.192), SQL merupakan standar bahasa *query* yang secara formal maupun *de facto* digunakan untuk mengakses sistem manajemen basis data relasional. Pada umumnya, sebuah DBMS menyediakan tiga bahasa, yaitu *Data Definition Language* (DDL), *Data Manipulation Language* (DML) dan *Data Control Language* (DCL).

#### 2.2.8.5. HTML

Menurut Fathansyah (2004, p327), *Hypertext Markup Language* (HTML) merupakan bahasa tampilan *hypertext* untuk kebutuhan umum. HTML menyediakan fitur – fitur untuk mendukung pemasukan data. Contohnya, sebuah dokumen HTML dapat mengatur bagaimana sebuah *form* harus ditampilkan. Program tampilan HTML dari *browser* memungkinkan pengguna untuk mengisi data ke dalam formulir tersebut. Menu dan fasilitas grafis untuk memasukan yang lainnya juga tersedia, seperti pemilihan *list*, penggunaan radio button menggerakkan *slide bar* untuk memilih sebuah nilai dalam batas nilai tertentu, penunjukan dan pemilihan lokasi dalam gambar dan sebagainya. Ketika pengguna telah selesai melakukan pemasukan data, dia dapat mengklik tombol ‘Ok’ untuk mengirimkan datanya ke *server Web*.

Tampilan layar aktual, formulir spesifik yang bisa diisi dan menu yang bisa dipilih, kesemuanya dikendalikan oleh dokumen HTML. Akan tetapi, tidak ada koneksi terus

menerus antara *client* dan *server*. *Server* tidak memiliki histori atas interaksinya dengan *client*, dan semua status informasi harus disimpan di *client* dan dikirim ke *server* setiap kali keduanya berkomunikasi. Hal ini merupakan salah satu kelemahan HTML.

#### 2.2.8.6. Javascript

Menurut Woychowsky (2007, p6-7), Javascript adalah bahasa pemrograman yang ringan, dapat dipahami dan berbasis obyek yang telah menjadi standar bahasa *scripting*. Javascript diadaptasikan dari bahasa pemrograman C. Javascript dapat digunakan untuk *event trapping*, validasi dan kebanyakan logika pemrograman *web code* yang ditanamkan di sisi klien menggunakan Javascript. Pengaplikasian Javascript dapat diakses dan digunakan menggunakan seluruh *browser* yang saat ini tersedia.

Menurut Flanagan (2011, p1), Javascript merupakan salah satu teknologi yang perlu dipelajari oleh *web developer*. HTML merupakan pengatur spesifikasi konten *web page*, CSS untuk mengatur spesifikasi tampilan *web page*, sedangkan Javascript digunakan untuk mengatur spesifikasi *behavior web page*. Javascript memperoleh *syntax* dari Java, namun selain dari kemiripan *syntax* yang digunakan sebenarnya berbeda dengan bahasa pemrograman Java.

Javascript telah berkembang dari hanya sebuah akar bahasa *scripting* menjadi sebuah bahasa pemrograman *general-purpose* yang kuat dan efisien. Javascript merupakan bahasa *scripting* yang biasanya disisipkan pada halaman HTML yang berisikan kode yang dijalankan pada komputer (*web browser*), Javascript merupakan bahasa interpreter yang maksudnya adalah *script* dieksekusi tanpa proses kompilasi.

### 2.2.8.7. CSS

Menurut Madcoms (2009, P89), CSS merupakan kependekan dari Cascading Style Sheet, yang digunakan untuk membantu pengguna mendesain isi halaman *web*. Bentuk penulisan CSS terdiri dari 3 bagian, yaitu *Selector*, Properti dan Nilai. Penulisan Properti dan nilai dipisahkan oleh tanda titik dua (;), jika nilai terdiri lebih dari satu kata, penulisan nilai diapit dengan tanda petik dua.

Untuk menggunakan CSS pada halaman web ada 3 pilihan, yaitu *External Style Sheet* (file CSS disimpan sendiri), *Internal Style Sheet* (menuliskan kode CSS dalam *tag Head*) dan *Inline Style Sheet* (langsung menuliskan kode CSS jadi satu dengan tag HTML atau XHTML).

#### 2.2.8.7.1 Bootstrap

Menurut Jake Spurslock (2013, p9) Bootstrap adalah *front-end framework* untuk membangun *website* yang responsif. Baik itu *application frameworks*, *blog*, atau aplikasi CMS lainnya, Bootstrap dapat menjadi pilihan terbaik yang Anda inginkan. Kombinasi dari HTML, CSS, dan Javascript membuat mudah untuk membangun *website* yang kuat tanpa menambahkan banyak kode. Dengan sistem *default grid*, *layout* menyatu dengan mudah, dan *styling* tombol, *navs*, dan tabel membuat markup dasar tampak memudahkan Anda menambahkan elemen interaktif ke *website* Anda.

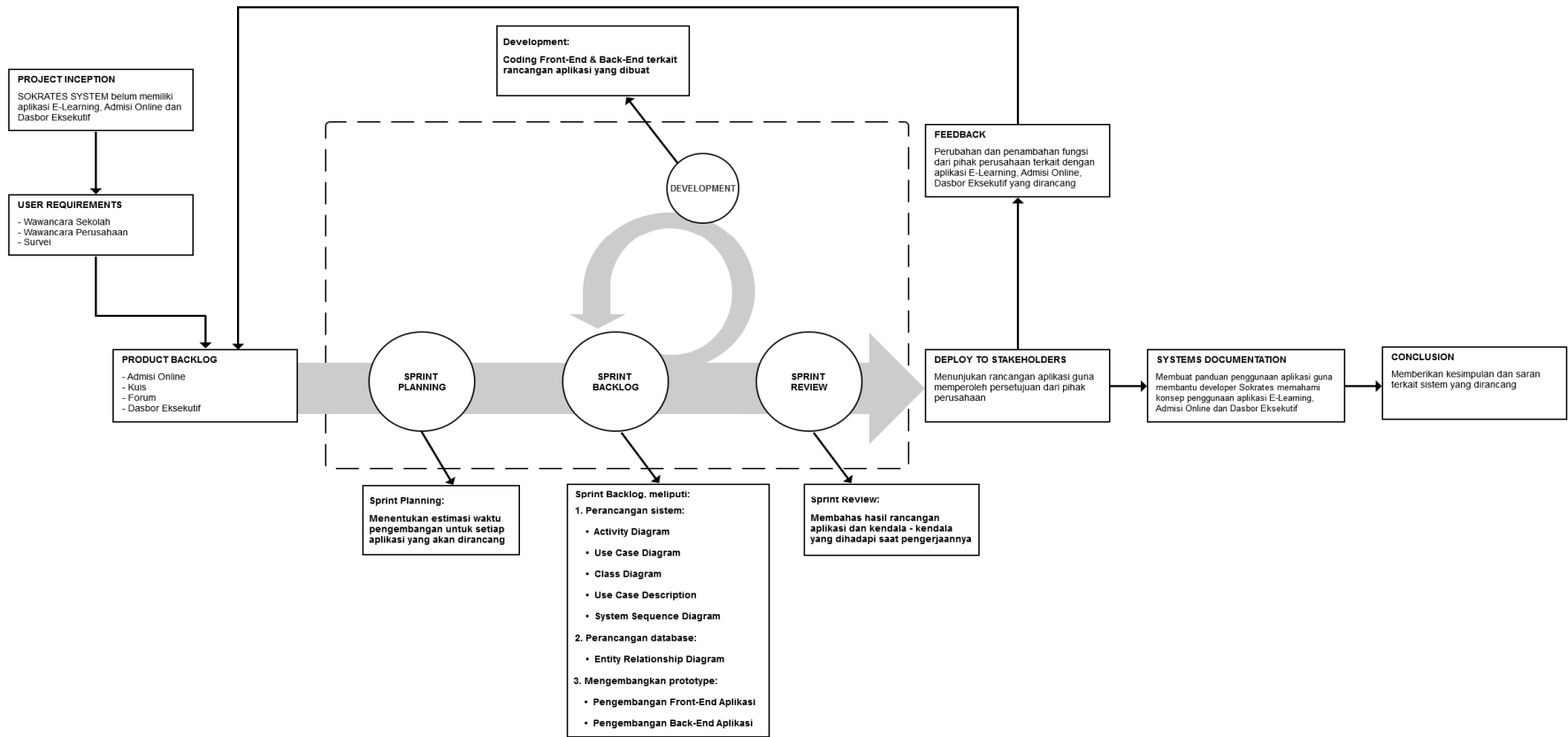
### 2.2.8.8. DBMS Tools

#### 2.2.8.8.1 MySQL

Menurut Loka Dwiartara (2010, p6) MySQL adalah *database*. *Database* sendiri merupakan suatu jalan untuk menyimpan berbagai informasi dengan membaginya

berdasarkan kategori-kategori tertentu. Dimana informasi –informasi tersebut saling berkaitan, satu dengan yang lainnya.

### 2.3. Kerangka Berpikir



Gambar 2.14 Kerangka Berpikir

