

BAB II

TINJAUAN PUSTAKA

2.1. Tinjauan Pustaka

Penelitian mengenai pembuatan *website* profil yang dilakukan oleh Yulis Kristanto dengan judul “Membangun *Website* Profil Sekolah Dasar Negeri Laweyan Surakarta” membangun *website* profil sekolah berguna untuk memberikan informasi kepada masyarakat umum juga sebagai media promosi.

Mohammad Labibul melakukan penelitian yang sama dengan judul “*Website* Sekolah Menengah Pertama Negeri 1 Gatak” yang menampilkan informasi SMPN 1 Gatak, meliputi profil sekolah, visi-misi, fasilitas, prestasi, guru, dll. Peneliti menggunakan bahasa pemrograman *PHP* dan Adobe Dreamweaver sebagai *software* serta *XAMPP* sebagai web-servernya.

Widya Kusuma Wardany dalam penelitiannya yang berjudul “Pembuatan *Website* Profil Sekolah Menengah Pertama Negeri 1 Jaten Karanganyar” SMPN 1 Jaten Karanganyar merupakan salah satu Sekolah Menengah Pertama Negeri Berstandar Nasional (SMP SSN) yang berada di kabupaten Karanganyar yang masih sangat membutuhkan interaksi maupun sosialisasi agar sekolah tersebut dapat lebih dikenal masyarakat luas maupun siswa-siswanya sendiri, yaitu dengan cara memberikan informasi secara lengkap, detail dan jelas mengenai sekolah tersebut. Dan merupakan salah satu anjuran dari pemerintah bahwa sekolah yang telah berstandar nasional dianjurkan untuk memiliki sebuah media informasi salah satunya yaitu *website*.

Dalam perancangan *website* tersebut menggunakan bahasa pemrograman *PHP* dan menggunakan *database MySQL*. Dari hasil perancangan tersebut pengunjung dapat berinteraksi dengan pihak sekolah tanpa harus mendatangi SMPN 1 Jaten secara langsung.

Dari ketiga penelitian memiliki persamaan dengan penelitian penulis yaitu membuat *website* sebagai media penyampaian informasi yang berkaitan dengan profil sekolah dan penulis akan lebih mengembangkan *fitur-fitur* dari *website-website* sebelumnya seperti penambahan *role user* dimana nantinya *user* selain admin bisa menulis berita atau menyampaikan artikel sesuai hak akses masing-masing. Tinjauan pustaka ini bertujuan sebagai bahan referensi dan rujukan terhadap hasil penelitian sebelumnya yang berhubungan dengan penelitian yang akan dilakukan.

2.2. Landasan Teori

2.2.1. Pengertian Sistem

Sistem adalah sekumpulan elemen yang saling terkait atau terpadu yang digunakan untuk mencapai suatu tujuan (Abdul Kadir, 2003). Selain itu sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan dan berkumpul untuk melakukan suatu kegiatan atau menyelesaikan suatu sasaran tertentu (Jogiyanto, 1999). Elemen sistem antara lain tujuan, masukan, keluaran, proses, mekanisme pengendalian, dan umpan balik serta berinteraksi dengan lingkungan dan memiliki batas (Abdul Kadir, 2003). Data yang masuk melalui *input* (masukan) kemudian diproses dan diolah dan dikeluarkan melalui *output* (keluaran) sesuai permintaan. Jadi dapat disimpulkan bahwa sistem merupakan himpunan atau grup dari elemen atau komponen yang berhubungan atau saling bergantung satu sama lain untuk mencapai tujuan tertentu.

2.2.2. Pengertian Informasi

Informasi berarti penerangan, pemberitahuan, kabar atau berita tentang sesuatu (Kamus Besar Bahasa Indonesia 2001). Informasi sebagai data yang telah diproses sedemikian rupa sehingga meningkatkan pengetahuan seseorang yang menggunakan data tersebut (Abdul Kadir, 2005). Informasi adalah data yang telah diolah menjadi sebuah bentuk yang berarti bagi penerimanya dan bermanfaat dalam pengambilan keputusan saat ini atau saat mendatang (Davis, 1999).

Data merupakan bentuk yang masih mentah yang belum dapat bercerita banyak, sehingga perlu diolah lebih lanjut untuk menjadi sebuah informasi.

2.2.3. Pengertian Sistem Informasi

Sistem informasi merupakan suatu komponen yang terdiri dari manusia, teknologi informasi, dan prosedur kerja yang memproses, menyimpan, menganalisis, dan menyebarkan informasi untuk mencapai suatu tujuan (Agus Mulyanto, 2009). Sistem informasi adalah kumpulan perangkat keras dan perangkat lunak yang dirancang untuk mentransformasikan data ke dalam bentuk informasi yang berguna (Abdul Kadir, 1993). Sistem informasi adalah sebuah rangkaian prosedur formal dimana data dikelompokkan, diproses menjadi informasi, dan di distribusikan kepada pemakai (Abdul Kadir, 2003).

2.2.4. Website

Menurut Jovan (2007:1) “*Website* adalah media menyampaikan informasi di internet” berupa teks, data, gambar diam ataupun bergerak, data animasi, suara, video maupun gabungan dari semuanya, baik itu yang bersifat statis maupun yang dinamis, dimana membentuk satu rangkaian bangunan yang saling berkaitan dimana masing-masing dihubungkan dengan jaringan halaman atau *hyperlink*.

Website adalah kumpulan dari berbagai macam halaman situs yang terangkum di dalam sebuah domain atau juga sub domain yang lebih tempatnya berada di dalam WWW (*World Wide Web*) yang tentunya terdapat di dalam internet. Halaman *website* biasanya berupa dokumen yang ditulis dalam format *HTML (Hyper Text Markup Language)* yang dapat diakses melalui HTTP (protokol yang menyampaikan berbagai informasi dari *server website* untuk ditampilkan kepada para *user* atau pemakai melalui *browser*).

2.2.5. CMS (Comment Management System)

Comment Management System atau yang biasa disingkat menjadi *CMS* merupakan sebuah perangkat lunak yang menyediakan tools atau perangkat untuk pembuatan, penerbitan atau pengelolaan suatu *website* (Robertson, 2003 dalam Rancangan Bangun Portal E-commerce, 2012). Di dalam sebuah *website* terdapat konten atau isi yang berisi informasi atau pengetahuan yang disajikan untuk pengguna. Konten merupakan suatu informasi digital yang digunakan untuk mengisi sebuah *page* (TreeWorks, 2005 dalam Rancang Bangun Portal E-commerce, 2012). Konten yang dimaksud biasanya meliputi *file-file* elektronik, gambar-gambar, *file-file*, *audio*, *video* atau dokumen-dokumen elektronik.

2.2.6. CMS (Comment Management System) Wordpress

WordPress adalah sebuah aplikasi sumber terbuka (*open source*) yang sangat populer digunakan sebagai mesin blog (*blog engine*). *WordPress* dibangun dengan bahasa pemrograman *PHP* dan basis data (*database*) *MySQL*. *PHP* dan *MySQL*, keduanya merupakan perangkat lunak sumber terbuka (*open source software*). Selain sebagai blog, *WordPress* juga mulai digunakan sebagai sebuah *CMS (Content Management System)* karena kemampuannya untuk dimodifikasi dan disesuaikan dengan kebutuhan penggunanya.

2.2.7. XAMPP

Xampp sebagaimana yang terlihat pada gambar 2.1 adalah perangkat lunak *open source*, yang mendukung banyak sistem operasi, 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 (4 sistem operasi apapun), *Apache*, *Mysql*, *PHP* dan *Perl*. Program ini tersedia dalam *GNU (General Public License) open source* dan merupakan *web server* yang mudah digunakan untuk dapat melayani tampilan halaman *website* yang dinamis (Agam, 2006).

2.2.8. Web Browser

Web browser (Kasiman, 2006) adalah program untuk menampilkan halaman yang berbentuk kode *HTML*. Semua halaman *website* ditulis dengan bahasa *HTML* (*Hypertext Mark Up Language*). Walaupun beberapa *file* mempunyai ekstensi yang berbeda (contoh: *.HTML*, *.php*, *.php3*) *output file-file* tersebut tetap *HTML*. *HTML* adalah medium yang selalu dikirimkan ke *web browser* baik halaman itu berupa halaman *statis*, sebuah *script* (seperti *PHP*), ataupun yang dibuat oleh program *CGI* (*Common Gateway Interface*).

2.2.9. Apache web server

(Andi Wahyu Rahardjo Emanuel:2006) *Apache Web Server* merupakan program untuk menjalankan *website* dalam sebuah komputer. *Web service* ini akan melayani setiap permintaan dari *web browser* dan mengirimkan data dalam bentuk *HTML* yang dapat dibaca oleh *web browser* dari pengguna komputer.

Apache merupakan salah satu *server web* yang paling banyak digunakan di dunia, beberapa keunggulan *Apache* dibandingkan dengan *web server* yang lain seperti *IIS* (*Internet Information Service*) dari Microsoft adalah kemampuannya untuk mendukung berbagai bahasa *script* paling populer seperti *PHP* (*Personal Home Page*) dan *JSP* (*Java Server Pages*). Hal lain yang membuat *Apache* lebih diminati adalah sistem lisensinya yang gratis sehingga mengurangi biaya yang perlu dikeluarkan dalam membangun situs *website* dinamis.

2.2.10. PHP (Personal Home Page Hypertext Processor)

Personal Home Page Hypertext Processor atau yang sering di sebut *PHP* menurut (Kadir, 2008) merupakan bahasa *script* yang ditempatkan dalam *server* dan diproses di *server* hasilnya dikirimkan ke *client*, tempat pemakainya menggunakan *browser*. Seperti bahasa pemrograman yang lain, *PHP* memiliki beberapa kelebihan. Adapun kelebihan dari *PHP* antara lain :

- a. *PHP* merupakan suatu bahasa *script* yang tidak melakukan sebuah kompilasi dalam penggunaannya.

- b. *PHP* dapat berjalan pada *website server* yang dirilis oleh Microsoft, juga pada *Apache* yang bersifat *open source*.
- c. Karena sifatnya yang *open source*, maka perubahan dan perkembangan *interpreted* pada *PHP* lebih cepat dan mudah, karena banyak *developer* yang siap membantu pengembangannya.
- d. *PHP* memiliki referensi yang begitu banyak sehingga sangat mudah untuk dipahami.

PHP dapat berjalan pada 3 *operating system*, yaitu *linux*, *Unix* dan *Windows*, dan juga dapat dijalankan secara runtime pada suatu *console*.

2.2.11. CSS (Cascading Style Sheet)

Menurut Bunafit Nugroho (2013:1), *Cascading Style Sheet* adalah bahasa *style sheet* yang digunakan untuk mengatur tampilan suatu dokumen yang ditulis dalam bahasa markup. *CSS* bekerja sebagai pelengkap pada elemen *HTML* yang kesemuanya itu dapat dikendalikan dengan menggunakan dengan menggunakan sebuah bahasa *script CSS*. Penggunaan *CSS* dilakukan untuk memperluas kemampuan *HTML* dalam memformat dokumen *website* atau untuk memperindah tampilan *website*. Penulisan kode *CSS* disisipkan pada tag *HTML*. Kode *CSS* ditulis dengan tag `<style>` dan `</style>` dengan mendefinisikan suatu *style* baru yang kemudian dapat digunakan berulang kali.

2.2.12. MySQL

Mysql (Kadir, 2008) adalah salah satu jenis *database server* yang menggunakan *SQL* sebagai bahasa dasar untuk mengakses *databasenya*. Dengan menggunakan *script PHP* dan *PERL Software database* ini dapat berfungsi atau berjalan pada semua platform sistem operasi yang biasa digunakan (*Windows*, *Linux*, *OS/2*, berbagai varian *Unix*).

Mysql merupakan salah satu jenis program yang berfungsi untuk mengolah, menyimpan data dan manipulasi data di *server*. Di dalam suatu program terdapat suatu penggunaan *database*. *Mysql* merupakan sebuah *database* bahasa yang dikembangkan dari *Structured Query Language (SQL)* yang digunakan untuk metode

berkomunikasi antara *script* program dengan *database server* dalam memasukan atau mengambil data. *Mysql* termasuk dalam kelompok *RDBMS*(*Relational Database Management Sistem*).

2.3. Perancangan Sistem

2.3.1. UML (*Unified Modeling Language*)

Menurut Nugroho (2010:6) “*UML (Unified Modeling Language)* adalah bahasa pemodelan (*modeling*) untuk sistem atau perangkat lunak yang berorientasi objek. Pemodelan sesungguhnya digunakan untuk penyederhanaan permasalahan-permasalahan yang kompleks sedemikian rupa sehingga lebih mudah dipelajari dan dipahami”.

Berdasarkan pendapat yang dikemukakan di *Unified Modeling Language (UML)* adalah sebuah bahasa yang berdasarkan grafik atau gambar untuk memvisualisasikan, membangun dan pendokumentasian dari sebuah sistem pengembangan perangkat lunak berbasis OOP (*Object Oriented Programming*).

Ada 4 (empat) macam relationship dalam *UML*, yaitu:

a. Kebergantungan (*Dependention*)

Merupakan hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (*dependent*) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (*independent*).

b. Asosiasi (*Association*)

Merupakan apa yang menghubungkan antara objek satu dengan objek lainnya, bagaimana hubungan suatu objek dengan objek lainnya. Suatu bentuk asosiasi adalah agregasi yang menampilkan hubungan suatu objek dengan bagian-bagiannya.

c. Generalisasi (*Generalization*)

Merupakan hubungan dimana objek anak (*descendent*) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor*). Arah dari atas kebawah dari objek induk ke objek anak dinamakan spesialisasi, sedangkan arah berlawanan sebaliknya dari arah bawah keatas dinamakan generalisasi.

d. Realisasi (*Realizations*)

Merupakan operasi yang benar-benar dilakukan oleh suatu objek.

2.3.2. *Use case Diagram*

Diagram *Use case* adalah diagram yang menunjukkan fungsionalitas suatu sistem atau kelas dan bagaimana sistem tersebut berinteraksi dengan dunia luar dan menjelaskan sistem secara fungsional yang terlihat *user*. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem

Use case merupakan teknik menangkap kebutuhan-kebutuhan fungsional dari sistem baru atau sistem yang diubah. Setiap *use case* terdiri dari satu atau lebih skenario yang menerangkan bagaimana sistem berinteraksi dengan pengguna atau sistem yang lain untuk mencapai suatu sasaran bisnis tertentu. Dalam teknik ini tidak diterangkan cara kerja sistem secara *internal* maupun implementasinya. Yang ditunjukkan adalah langkah-langkah yang dilakukan pengguna dalam menggunakan perangkat lunak. Pada dasarnya ada dua jenis *use case* yaitu diagram *use case* dan naratif *use case* (Nyimas Artina:2006).

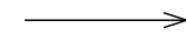
2.3.3. *Sequence Diagram*

Menurut Whitten (2007:394) *System Sequence Diagram* adalah diagram yang menggambarkan interaksi antara aktor dan sistem untuk skenario dalam *usecase*. Menurut Satzinger et al. (2010:252) *System sequence diagram* digunakan untuk menggambarkan aliran dari informasi yang masuk dan keluar dari sistem yang terotomatisasi. *System sequence diagram* merupakan tipe dari *interaction diagram* yaitu *communication diagram* atau *sequence diagram* yang menunjukkan interaksi diantara objek.

2.3.4. Class Diagram

Class diagram dibuat untuk menjelaskan hubungan antar kelas yang satu dengan kelas yang lainnya. Di dalam beberapa kelas terdapat atribut yang menjelaskan komponen apa saja yang terdapat di dalam kelas tersebut dan terdapat operation yang berfungsi sebagai penjelas kegiatan apa saja yang dapat dilakukan oleh suatu aktor ketika berada didalam kelas tersebut. Terdapat beberapa simbol dari bagian *Class diagram*, simbol-simbol tersebut antara lain terlihat pada Tabel 2.1.

Tabel 2. 1 Tabel Relasi *Class diagram*

Nama Relasi	Gambar	Keterangan
<i>Association</i>		Relasi antar kelas dengan makna umum. Biasanya disertai dengan <i>multiplicity</i> (keterangan banyak).
<i>Directed Association</i>		Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain.
<i>Agregation</i>		Relasi antar kelas dengan makna semua bagian (<i>whole-part</i>).
<i>Dependency</i>		Relasi antar kelas dengan makna kebergantungan antar kelas.
<i>Generalisation</i>		Relasi antar kelas dengan makna kebergantungan antar kelas.
<i>Competition</i>		Relasi antar kelas dengan makna ada beberapa kelas yang merupakan bagian dari kelas utama.

2.3.5. Activity Diagram

Activity diagram menggambarkan berbagai alir aktifitas dalam sebuah sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. *Activity diagram* tidak menggambarkan sifat internal dari sebuah sistem dan

interaksi antara beberapa sub sistem secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum.

Satzinger, J.W., Jackson, R.B., & Burd, S.D. (2010:142). Di bawah ini merupakan penjelasan simbol-simbol yang digunakan dalam *activity diagram*:

1. *Swimlane*(Merupakan area persegi pada *activity diagram* yang mewakili seluruh aktivitas di dalamnya.)
2. *Starting activity*(Merupakan awal dari aktivitas di dalam sistem.)
3. *Activity* (Merupakan aktivitas yang dilakukan di dalam sistem.)
4. *Decision activity* (Merupakan aktivitas yang harus dipilih.)
5. *Concurrent activities* (Merupakan aktivitas yang dilakukan secara persamaan atau paralel, biasanya diawali dengan *synchronization bar*.)
6. *Synchronization bar* (Merupakan simbol di dalam *activity diagram* yang digunakan untuk mengendalikan pemisahan atau penyatuan beberapa aktivitas.)
7. *Ending activity* (Merupakan akhir dari aktivitas di dalam sistem.)

2.3.6. Database (Basis Data)

Raghu Ramakrishnan dan Johannes Gehrke (2004:3) menyatakan bahwa *database* adalah kumpulan data, umumnya mendeskripsikan aktivitas satu organisasi yang berhubungan atau lebih. Setiap *database* dapat berisi atau memiliki sejumlah objek *database* seperti *field*, *table*, *indeks*, dan lain-lain

Menurut Connolly dan Begg (2002 : 14), pengertian basis data yaitu kumpulan koleksi data-data yang saling berhubungan secara logika yang isinya didesain untuk memenuhi kebutuhan informasi dari suatu perusahaan.

Ada beberapa istilah umum yang sering dipakai pada *database*, yaitu sebagai berikut :

- *Field*, yaitu sekumpulan kecil dari kata atau sebuah deretan angka-angka.
- *Record*, yaitu kumpulan dari *field* yang berelasi secara logis.
- *File*, yaitu kumpulan dari *record* yang berelasi secara logis.
- *Entity*, yaitu orang, tempat, benda, atau kejadian yang berkaitan dengan informasi yang disimpan.
- *Attribute*, yaitu setiap karakteristik yang menjelaskan suatu entity.
- *Primary key*, yaitu sebuah *field* yang nilainya unik yang tidak sama antara satu *record* dengan *record* yang lain.
- *Foreign key*, yaitu sebuah *field* yang nilainya berguna untuk menghubungkan *primary key* yang berada pada tabel yang berbeda.

2.3.7. ERD (*Entity Relationship Diagram*)

ERD merupakan suatu model untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi. *ERD* untuk memodelkan struktur data dan hubungan antar data, untuk menggambarannya digunakan beberapa notasi dan simbol.

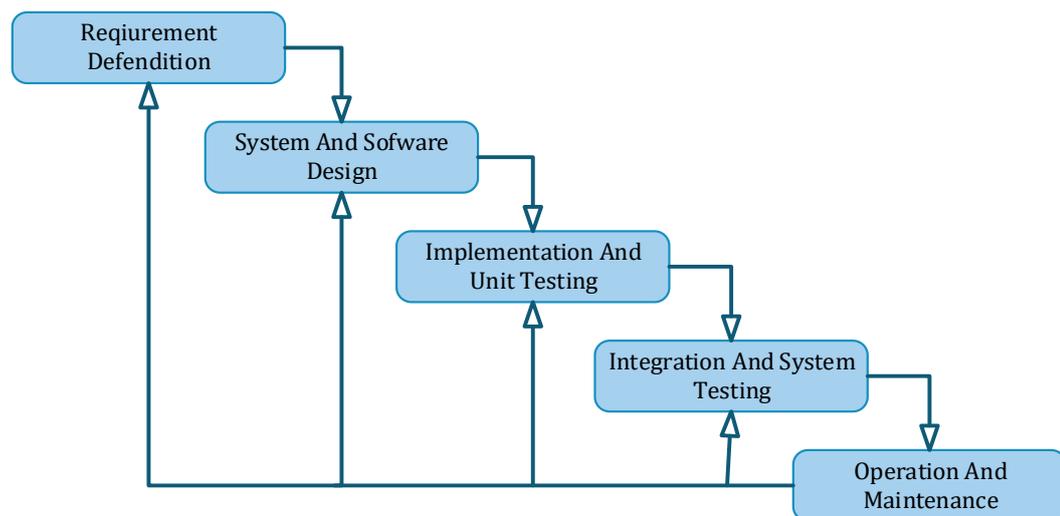
Menurut Brady dan Loonam (2010), *ERD* merupakan teknik yang digunakan untuk memodelkan kebutuhan data dari suatu organisasi, biasanya oleh sistem analis dalam tahap analisis persyaratan proyek pengembangan sistem. Sementara seolah-olah teknik diagram atau alat peraga memberikan dasar untuk desain *database* relasional yang mendasari sistem informasi yang dikembangkan. *ERD* bersama-sama dengan detail pendukung merupakan model data yang pada gilirannya digunakan sebagai spesifikasi untuk *database*.

2.4. Metode Pengembangan Sistem

2.4.1. Waterfall

Metode *Waterfall* pertama kali diperkenalkan oleh Royce pada tahun 1970 dengan 7 (tujuh) tahapan yang berurutan walaupun juga mempunyai *feedback loop* antar tahapan jika diperlukan.

Metode ini mengalami banyak perbaikan dan perubahan diantaranya adalah perubahan langkah dari 7 (tujuh) menjadi 5 (lima) tahapan (Pressman, 2010), (Sommerville, 2011) seperti terlihat pada Gambar 2.1.



Gambar 2. 1 Metode *Waterfall*

Metode *Waterfall* yang dibahas pada sub bab ini merupakan metode *Waterfall* versi Sommerville (2011). Pertimbangan menggunakan versi Sommerville karena metode ini merupakan versi terbaru dari metode *Waterfall*. Metode ini mempunyai tahapan-tahapan sebagai berikut:

a. *Requirements analysis and definition*

Layanan sistem, kendala, dan tujuan ditetapkan oleh hasil konsultasi dengan pengguna yang kemudian didefinisikan secara rinci dan berfungsi sebagai spesifikasi sistem.

b. *System and software design*

Tahapan perancangan sistem mengalokasikan kebutuhan-kebutuhan sistem baik perangkat keras maupun perangkat lunak dengan membentuk arsitektur sistem secara keseluruhan. Perancangan perangkat lunak melibatkan identifikasi dan penggambaran abstraksi sistem dasar perangkat lunak dan hubungannya.

c. *Implementation and unit testing*

Pada tahap ini, perancangan perangkat lunak direalisasikan sebagai serangkaian program atau unit program. Pengujian melibatkan verifikasi bahwa setiap unit memenuhi spesifikasinya.

d. *Integration and system testing*

Unit-unit individu program atau program digabung dan diuji sebagai sebuah sistem lengkap untuk memastikan apakah sesuai dengan kebutuhan perangkat lunak atau tidak. Setelah pengujian, perangkat lunak dapat dikirimkan ke *customer*.

e. *Operation and maintenance*

Biasanya tahapan ini merupakan tahapan yang paling panjang. Sistem dipasang dan digunakan secara nyata. *Maintenance* melibatkan pembetulan kesalahan yang tidak ditemukan pada tahapan-tahapan sebelumnya, meningkatkan implementasi dari unit sistem, dan meningkatkan layanan sistem sebagai kebutuhan baru.

Pada prinsipnya, setiap tahapan di metode *Waterfall* menghasilkan satu atau lebih dokumen yang sudah disetujui. Tahap berikutnya tidak dapat dimulai sebelum tahapan sebelumnya selesai. Dalam tataran praktis, tahapan-tahapan tersebut saling tumpang tindih dan memberikan informasi satu sama lain.

Proses pengembangan perangkat lunak bukan merupakan model linier yang sederhana karena juga melibatkan umpan balik (*feedback*) dari satu tahapan ke tahapan lainnya. Dokumen yang dihasilkan pada setiap tahapan ada kemungkinan harus diubah supaya sesuai dengan perubahan yang sudah dibuat (Sommerville, 2011).