



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB II

LANDASAN TEORI

2.1 *Database*

Database dapat diartikan sebagai markas atau gudang, tempat bersarang atau berkumpul suatu data mentah (*raw*) yang belum di olah sama sekali. Prinsip utama *database* adalah pengaturan data dengan tujuan utama fleksibilitas dan kecepatan dalam pengambilan data kembali. Demikian didefinisikan *database* sebagai kumpulan bentuk *file* tabel yang saling terhubung yang akan disimpan pada informasi lain (Yakub, 2009).

2.2 *PHP*

PHP (dahulu dikenal sebagai *Personal Home Page*, sekarang *PHP: Hypertext Preprocessor*) yang merupakan salah satu jenis database server yang sangat terkenal dan banyak digunakan untuk membangun aplikasi *web* yang menggunakan database sebagai sumber dan pengolahan datanya. (M.Rudianto, 2011).

Menurut Anhar, *PHP* yaitu bahasa pemrograman *web server-side* yang bersifat *opensource*. *PHP* merupakan *script* yang terintegrasi dengan *HTML* dan berada pada *server* (*server side HTML embedded scripting*). *PHP* adalah *script* yang digunakan untuk membuat halaman *website* yang dinamis. Dinamis berarti

halaman yang ditampilkan dibuat saat halaman itu diminta oleh pengguna atau *client* (Anhar, 2010).

2.3 *HTML*

HTML adalah bahasa standar untuk membuat halaman *web* maupun situs. Namun, sebagai bahasa *markup* sederhana, *HTML* memiliki keterbatasan dalam fungsinya yang memiliki bagian-bagian seperti judul, paragraf, daftar, gambar, dan lain-lain. Sebagai *Web* yang telah berkembang, bahasa baru telah dikembangkan untuk menyediakan lebih banyak fitur dan fungsionalitas yang lebih besar ke situs Web (Chaffey, 2011).

2.4 *MySQL*

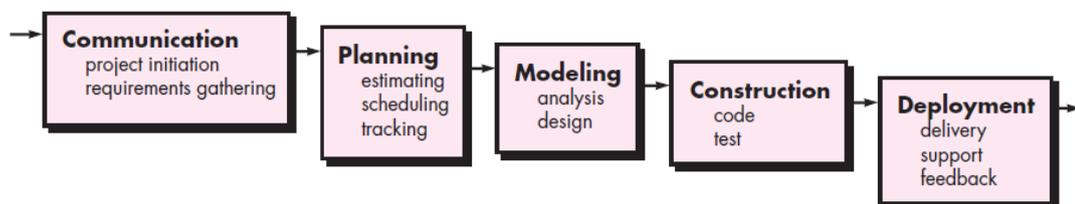
Menurut pendapat Anhar bahwa *MySQL* (*Structured Query Language*) dapat didefinisikan sebagai sebuah perangkat lunak sistem manajemen basis data atau *DBMSSQL* dari sekian banyak *DBMS* seperti; *Oracle*, *MSSQL*, *Postrage SQL*, dan lain-lain. *MySQL* merupakan *DBMS* yang *multithread*, *multi-user* yang bersifat gratis dibawah lisensi *GeneralPublicLicence* (GPL) (Anhar, 2010).

2.5 *Model Waterfall*

Terdapat dua pendapat tentang deskripsi dari model *waterfall* yang dipakai dalam pengembangan *software* yaitu model *waterfall* yang dikembangkan oleh Roger S. Pressman dan Ian Sommerville.

2.5.1 Model *Waterfall* (Pressman)

Menurut Pressman (Pressman, 2010), model *waterfall* adalah model klasik yang bersifat sistematis, berurutan dalam membangun *software*. Nama model ini sebenarnya adalah “*LinearSequentialModel*”. Model ini sering disebut juga dengan “*classiclifecycle*” atau metode *waterfall*. Model ini termasuk ke dalam model *generic* pada rekayasa perangkat lunak dan pertama kali diperkenalkan oleh Winston Royce sekitar tahun 1970 sehingga sering dianggap kuno, tetapi merupakan model yang paling banyak dipakai dalam *SoftwareEngineering* (SE) atau dalam perancangan *software*. Model ini melakukan pendekatan secara sistematis dan berurutan. Disebut dengan *waterfall* karena tahap demi tahap yang dilalui harus menunggu selesainya tahap sebelumnya dan berjalan berurutan. Fase-fase dalam model *Waterfall* yang ditampilkan pada gambar 2.1 (Pressman, 2010):



Gambar 2. 1 Tahapan pada metode *Waterfall* menurut Roger S. Pressman

Sumber: (Pressman, 2010)

1. *Communication (Project Initiation & Requirements Gathering)*, sebelum segala sesuatu dimulai, perlu adanya komunikasi antara *customer* dengan si pembuat *program* untuk memahami segala permintaan *customer* sehingga hasil *software* yang sudah selesai tersebut sesuai dengan keinginan *customer*. Hasil dari komunikasi tersebut akan dijadikan

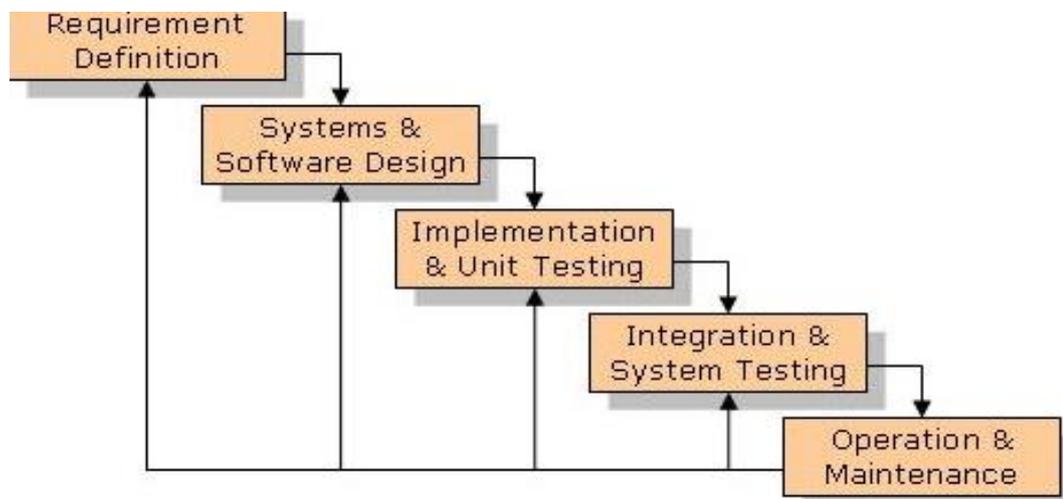
sebagai inisialisasi proyek, seperti menganalisa permasalahan yang akan dihadapi kedepannya, mengumpulkan data-data yang berhubungan dengan kepentingan pembangunan sistem, serta membantu mendefinisikan fitur-fitur dan fungsi-fungsi *software*.

2. *Planning (Estimating, Scheduling, Tracking)*, tahap berikutnya yaitu tahap perencanaan yang menjelaskan tentang estimasi dari hal-hal teknis yang akan dikerjakan, resiko yang dapat terjadi, sumber daya yang diperlukan dalam pembuatan sistem, menetapkan jadwal kerja, dan selalu mengikuti perkembangan proses pengerjaan sistem.
3. *Modeling (Analysis & Design)*, tahapan ini adalah tahap perancangan dan permodelan arsitektur sistem yang berfokus pada perancangan struktur data, arsitektur *software*, tampilan interface, dan algoritma program. Tujuannya untuk lebih memahami gambaran besar dari apa yang akan dikerjakan.
4. *Construction (Code & Test)*, pada tahap ini pekerjaan yang bersifat teknis sudah mulai dilakukan. Hasil desain yang sudah ada kemudian diteruskan menjadi bentuk kode atau bentuk bahasa pemrograman yang dapat dibaca oleh mesin atau komputer. Setelah proses pengkodean (*coding*) selesai, dilakukan kembali pengujian terhadap sistem dan juga kode yang telah dibuat dengan tujuan untuk mencari kesalahan maupun *bug* yang nantinya akan diperbaiki.
5. *Deployment (Delivery, Support, Feedback)*, tahap *Deployment* merupakan tahapan implementasi *software* ke *customer* atau kepada semua pengguna

sistem, pemeliharaan *software* secara berkala, perbaikan *software*, evaluasi *software*, dan pengembangan *software* berdasarkan umpan balik (*feedback*) yang diberikan agar sistem dapat tetap berjalan dan berkembang sesuai dengan fungsinya.

2.5.2 Model *Waterfall* (Sommerville)

Jika dibandingkan dengan model *waterfall* menurut (Sommerville, 2011), bahwa metode *sequential-linear* atau *waterfall* mengusulkan sebuah pendekatan kepada perkembangan perangkat lunak yang sistematis dan sekuensial yang mulai pada tingkat dan kemajuan sistem pada seluruh analisis, desain, kode, pengujian dan pemeliharaan. Pada gambar 2.2 ini adalah tahapan-tahapan dari metode *waterfall*:



Gambar 2.2 Tahapan pada metode *Waterfall* menurut Ian Sommerville

Sumber: (Sommerville, 2011)

1. Analisis dan definisi persyaratan (*Requirement Definition*), pelayanan,

batasan, dan tujuan sistem ditentukan melalui konsultasi langsung dengan pengguna sistem (*user*). Persyaratan ini kemudian didefinisikan secara rinci dan berfungsi sebagai spesifikasi sistem.

2. Perancangan sistem dan perangkat lunak (*System and Software Design*), Proses perancangan sistem membagi persyaratan dalam sistem perangkat keras atau perangkat lunak. Kegiatan ini menentukan arsitektur sistem secara keseluruhan. Perancangan perangkat lunak melibatkan identifikasi dan deskripsi abstraksi sistem perangkat lunak yang mendasar dan hubungannya.
3. Implementasi dan pengujian unit (*Implementation and Unit Testing*), Perancangan perangkat lunak direalisasikan sebagai serangkaian *program* atau *unit program* dan melakukan pengujian terhadap setiap fungsi-fungsi yang ada didalam *program*.
4. Integrasi dan pengujian sistem (*Integration and System Testing*), pada setiap *Unit program* yang ada diintegrasikan dan diuji sebagai sistem yang lengkap untuk menjamin bahwa persyaratan sistem telah dipenuhi. Setelah pengujian sistem, perangkat lunak dikirim ke *customer*.
5. Operasi dan pemeliharaan (*Operation and Maintenance*), Sistem yang telah selesai dibuat di-*install* dan dipakai. Pemeliharaan mencakup dari berbagai *error* atau kesalahan yang tidak ditemukan pada tahap-tahap terdahulu, yang akan ditemukan pada saat penggunaan sistem.

2.5.3 Keuntungan Metode *Waterfall*

Berikut ini adalah beberapa kelebihan yang ada dalam penggunaan metode *Waterfall* dalam pembangunan sistem yang membuat penelitian ini menggunakan metode tersebut dibandingkan dengan metode yang lain (Pressman, 2010), yaitu:

1. Kualitas dan sistem yang dihasilkan akan sesuai dengan keinginan *customer*. Ini dikarenakan oleh pelaksanaannya secara bertahap, sehingga tidak terfokus pada tahapan tertentu.
2. Walaupun metode ini tergolong kuno dan tradisional namun masih lebih baik digunakan dibandingkan dengan menggunakan pendekatan secara asal-asalan, selain itu juga metode ini lebih masuk akal jika kebutuhan-kebutuhan sudah diketahui dengan jelas.
3. Sesuai apabila digunakan untuk sistem yang kebutuhannya jelas dan dapat diperhitungkan diawal pembuatan, sehingga kesalahan dapat dihindari.
4. Metode ini mudah untuk diterapkan ataupun diaplikasikan pada proses pengembangan sistem.
5. Dokumen pengembangan sistem sangat terorganisir, karena setiap tahap harus terselesaikan dengan lengkap sebelum melangkah ke tahap berikutnya. Jadi pada masing-masing fase memiliki dokumennya tersendiri.

2.6 XAMPP

Menurut Betha Sidik, *XAMPP* atau (*X(windows/linux) Apache MySQL PHP and Perl*) merupakan paket *webservice PHP* dan *databaseMySQL* yang paling populer dikalangan pengembang web dengan menggunakan *PHP* dan *MySQL* sebagai *database*-nya (Sidik, 2017).

Menurut Purbadian (Purbadian, 2016) berpendapat bahwa *XAMPP* adalah suatu *software* yang bersifat *open source* yang merupakan pengembangan dari *LAMP (Linux, Apache, MySQL, PHP dan Perl)*.

Sedangkan menurut Kartini (Kartini, 2013) *XAMPP* merupakan tool yang menyediakan paket perangkat lunak ke dalam satu buah paket.

Berdasarkan pengertian diatas dapat disimpulkan bahwa *XAMPP* merupakan *tools* pembantu pengembangan paket perangkat lunak berbasis *open source* yang menggabungkan *Apache web server, MySQL, PHP* dan beberapa modul lainnya di dalam satu paket aplikasi.

2.7 Structured Analysis and Design (SAD)

Metode pendekatan secara terstruktur ini pertama kali diperkenalkan pada tahun 1970, yang merupakan hasil turunan dari pemrograman terstruktur tersebut. Metode ini terus menerus diperbaiki sampai akhirnya dapat digunakan secara baik dalam dunia nyata. Pendekatan terstruktur dilengkapi dengan alat-alat (*tools*) dan teknik-teknik (*techniques*) yang dibutuhkan dalam pengembangan sistem, sehingga hasil akhir dari sistem yang dikembangkan akan didapatkan sistem yang

strukturnya didefinisikan dengan baik dan jelas. Seperti pada pendekatan *engineering* yang dipakai dalam pemecahan masalah, pendekatan terstruktur memerlukan prosedur dan pendataan yang baku dan jelas atau paling tidak memerlukan metodologi yang akan dipakai dalam mengembangkan sistem informasi. Struktur dapat menentukan perintah (*order*) serta dapat meningkatkan kemampuan pemahaman terhadap sistem yang rumit. Oleh karena itu struktur merupakan ciri utama pada *design* dari pendekatan secara terstruktur. Ciri-ciri utama yang mendukung pendekatan terstruktur (Sherwood, 2009):

1. Merancang berdasar modul, modularisasi adalah proses yang membagi suatu sistem menjadi beberapa modul yang dapat beroperasi secara tersendiri atau *independent*.
2. Melakukan pendekatan secara visual seperti menggunakan *tools DFD* atau *ERD*, sehingga dalam tahap pembangunan dapat mudah dimengerti oleh pengguna atau *programmer*.
3. Kegiatan dilakukan secara paralel, maksudnya pengembangan subsistem-subsistem dapat dilakukan secara paralel yaitu secara sejajar dan tersusun, sehingga akan memperpendek waktu pengembangan sistem.
4. Bekerja dengan pendekatan *top-down*, pekerjaan yang dimulai dari level atas (secara *global*) kemudian diuraikan sampai ke tingkat modul (rinci).

Berikut ini adalah *tools* dari *SAD* yang akan digunakan dalam sebagai pendekatan pembangunan sistem pada CV Sumber Makmur Pompa, yaitu:

1. *ERD (Entity Relationship Diagram)*
2. *DFD (Data Flow Diagram)*

2.8 *ERD (Entity Relationship Diagram)*

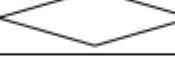
Entity Relationship diagram (ERD) adalah teknik yang digunakan untuk memodelkan kebutuhan data dari suatu organisasi, biasanya oleh Sistem Analisis dalam tahap analisis persyaratan proyek pengembangan sistem (Brady & Loonam, 2010).

Sedangkan menurut Shalahuddin dan Rosa, *ERD* adalah pemodelan awal basis data yang dikembangkan berdasarkan teori himpunan dalam bidang matematika. *ERD* digunakan untuk pemodelan basis data relasional (Shalahuddin & Rosa, 2018).

Komponen *Entity Relationship Diagram (ERD)* adalah sebagai berikut (Edhy Sutanta, 2011):

1. Entitas, yaitu suatu objek yang dapat dibedakan dari lainnya yang dapat diwujudkan dalam basis data. Objek dasar dapat berupa orang, benda, atau hal yang keterangannya perlu disimpan didalam basis data, dan entitas memiliki beberapa atribut, sebagai contoh yaitu atribut dari entitas mahasiswa yaitu NIM, nama, alamat, email, dan lain-lain.

2. Atribut, sebagai keterangan-keterangan yang terkait pada sebuah entitas yang perlu disimpan dalam basis data. Atribut digunakan untuk mendeskripsikan karakteristik dari entitas tersebut. Isi dari atribut mempunyai sesuatu yang dapat mengidentifikasi isi elemen satu dengan yang lain. Atribut digambarkan dengan simbol elips.
3. Relasi, merupakan hubungan antara sejumlah entitas yang berasal dari himpunan entitas yang berbeda. Relasi dinyatakan dengan simbol belah ketupat, dan sebisa mungkin menggunakan nama relasi yang mudah dipahami dan dapat menyatakan maknanya dengan jelas.
4. Dengan garis yang menandakan hubungan antara relasi dan entitas maupun entitas dengan atribut.

Komponen	Keterangan
	Persegi panjang mewakili entitas
	Elips mewakili atribut
	Belah ketupat mewakili relasi
	Garis menghubungkan relasi dan entitas atau relasi dan entitas dengan atribut

Gambar 2. 3 Entity Relationship Diagram (ERD) dan simbolnya

Sumber: (Edhy Sutanta, 2011)

UNIVERSITAS
MULTIMEDIA
NUSANTARA

2.9 DFD (Data Flow Diagram)

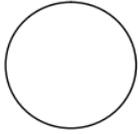
Menurut Kristanto, *Data Flow Diagram (DFD)* merupakan suatu model logika data atau proses yang dibuat untuk menggambarkan darimana asal data dan kemana tujuan data yang keluar dari sistem, dimana data disimpan, proses apa yang menghasilkan data tersebut dan interaksi antara data yang tersimpan dan proses yang dikenakan pada data tersebut (Kristanto, 2011).

Sedangkan menurut Sukamto dan Shalahuddin *Data Flow Diagram* atau dalam bahasa Indonesia menjadi Diagram Alir Data adalah representasi grafik yang menggambarkan aliran informasi dan transformasi informasi yang diaplikasikan sebagai data yang mengatur dari masukan (input) dan keluaran (*output*). *DFD* tidak sesuai untuk memodelkan sistem yang menggunakan pemrograman berorientasi objek (Sukamto & Shalahuddin, 2018).

Adapun simbol-simbol yang digunakan dalam *Data Flow Diagram* menurut Sukamto dan Shalahuddin (Sukamto & Shalahuddin, 2018), yaitu:

1. *Terminator*, mewakili entitas eksternal yang berkomunikasi dengan sistem yang sedang dikembangkan. Biasanya *terminator* dikenal dengan nama entitas (eksternal), sumber atau tujuan (*source and sink*). Dalam hal penamaan pada *terminator*, biasanya menggunakan kata benda. *Terminator* dapat berupa orang, sekelompok orang, organisasi, departemen di dalam organisasi, atau perusahaan yang sama tetapi di luar kendali sistem yang sedang dibuat modelnya

2. Proses, komponen proses menggambarkan bagian dari sistem yang mentransformasikan *input* ke *output*. Pemberian nama pada komponen proses menggunakan kata kerja yang membutuhkan subyek (transitif).
3. Alur Data, ini digunakan untuk menerangkan perpindahan data atau satu paket data/informasi dari satu bagian sistem ke bagian lainnya. Biasanya pemberian nama pada alur data dilakukan dengan menggunakan kata benda.
4. *Data Store*, digunakan untuk membuat model sekumpulan paket data. *Data Store* biasanya berkaitan dengan penyimpanan seperti *file* atau *database* yang berkaitan dengan penyimpanan, baik secara komputerisasi, misalnya *file hardisk*, maupun manual, misalnya nama dan alamat pada buku alamat. Nama yang diberikan pada *data store* biasanya menggunakan kata benda jamak.

No	Simbol	Keterangan
1.		<i>Proses</i> , Simbol ini digunakan untuk proses pengolahan atau transformasi data.
2.		<i>External Entity</i> , Simbol ini digunakan untuk menggambarkan asal atau tujuan data.
3.		<i>Data Flow</i> , simbol ini digunakan untuk menggambarkan aliran data yang berjalan.
4.		<i>Data Store</i> , Simbol ini digunakan untuk menggambarkan data flow yang sudah disimpan atau diarsipkan.

Gambar 2. 4 Data Flow Diagram (DFD) dan simbolnya

Sumber: (Sukamto & Shalahuddin, 2018)

2.10 Pembelian

Pengertian pembelian secara umum yaitu suatu peristiwa atau tindakan yang dilakukan oleh dua belah pihak dengan tujuan menukarkan barang atau jasa dengan menggunakan alat transaksi yang sah dan sama-sama memiliki kesepakatan dalam transaksinya, dalam pembelian terkadang akan terjadi tawar menawar antara pembeli dan penjual hingga mendapatkan kesepakatan harga yang kemudian akan melakukan transaksi penukaran barang atau jasa dengan alat tukar yang sah dan disepakati kedua belah pihak (Riyanto, 2010)

2.11 Penjualan

Penjualan ialah suatu kegiatan yang ditunjukkan untuk mencari pembeli, mempengaruhi dan memberikan petunjuk agar pembeli dapat menyesuaikan kebutuhannya dengan produk yang ditawarkan serta mengadakan perjanjian mengenai harga yang menguntungkan bagi kedua belah pihak (Ismanthono, 2010)

Penjualan adalah ilmu dan seni mempengaruhi pribadi yang dilakukan oleh penjual, untuk mengajak orang lain bersedia membeli barang atau jasa yang ditawarkan (Swasta, 2009).

2.12 Persediaan Barang (*Inventory*)

Persediaan barang (*inventory*) merupakan suatu aktiva yang meliputi barang-barang milik perusahaan dengan maksud untuk dijual dalam suatu periode usaha tertentu atau persediaan barang-barang yang masih dalam pengerjaan atau

proses produksi ataupun persediaan bahan baku yang menunggu penggunaannya dalam proses produksi (Alexandri, 2009).

Sartono Agus (Agus, 2010) mengatakan bahwa persediaan umumnya merupakan salah satu jenis aktiva lancar yang jumlahnya cukup besar dalam suatu perusahaan.

2.13 Penelitian Sebelumnya

2.13.1 Implementasi Model *Waterfall* Pada Pengembangan Sistem Informasi Alumni SMKN 1 Jenangan Ponorogo

SESINDO, Vol. XV, No. 2, Edisi November 2015

Sistem ini dibuat oleh **Eka Nur Ahmad Romadhoni, Triyanna Widiyaningtyas**, dan **Utomo Pujianto** yang bertujuan untuk merancang sistem informasi untuk Alumni SMKN 1 Janengan Ponogoro yang diharapkan dapat mengelolah informasi tentang total alumni yang bekerja setelah lulus dan melanjutkan studi, himpunan alumni sekolah dan mengolah data keterserapan alumni di dunia industri. Strategi yang digunakan untuk meningkatkan partisipasi alumni ialah dengan memberikan fitur informasi lowongan kerja, pelatihan, dan studi lanjut dengan sistem berbasis *web*. Dengan dibangunnya sistem informasi tentang informasi alumni SMKN 1 Janengan Ponogoro, dari pihak alumni, mereka dapat melakukan pencarian informasi tentang pekerjaan-pekerjaan yang dijalani sesama alumni sekolah tersebut, dan mereka dapat bertukar informasi tentang lowongan pekerjaan dan pelatihan secara *online*, untuk pihak sekolah dapat merekapitulasi total alumni yang bekerja setelah lulus dan total alumni yang

melanjutkan studi dan dapat menyimpan informasi tersebut untuk keperluan lainnya.

Proses pembuatan sistem informasi berbasis *web* ini menggunakan *Unified Modeling Language (UML)* pada perancangan sistem dan perangkat lunak. *UML* merupakan sekumpulan diagram yang telah memiliki standar untuk menggambarkan rancangan perangkat lunak berorientasi objek. Diagram *UML* yang digunakan ialah *diagram use case*, *diagram sequence*, dan diagram kelas. Proses perancangan *database* menggunakan *Entity Relationship Diagram (ERD)*.

Model pengembangan yang digunakan pada penelitian ini ialah model Waterfall. Model waterfall merupakan model pengembangan sistem informasi yang sistematis dan sekuensial. Alasan penggunaan model *waterfall* sebagai metode pengembangan sistem informasi alumni SMKN 1 Jenangan Ponorogo ialah kebutuhan pihak sekolah telah terdefinisi secara jelas dan tahap-tahap pada model *waterfall* terstruktur secara jelas.

Hasil penelitian tersebut menunjukkan bahwa dengan penggunaan sistem informasi berbasis *web* pada SMKN 1 Jenangan Ponorogo ini dapat memberikan informasi data yang ingin dibutuhkan oleh pengguna sistem, seperti informasi tentang alumni, sistem ini juga memberikan kemudahan untuk pengguna dalam hal pencarian data lowongan kerja, pekerjaan, himpunan alumni dalam lingkup SMKN 1 Jenangan Ponorogo, karena program yang dirancang berbasis *web* yang juga dilakukan secara *online* sehingga menghilangkan batasan waktu, jarak, dan tempat sebagai halangan bagi pengguna. Penggunaan model *waterfall* dapat membantu

pengembangan sistem informasi alumni di SMKN 1 Jenangan Ponorogo serta alasan pemilihan model waterfall sebagai metode pengembangan ialah kebutuhan sekolah telah terdefinisi secara jelas dan sistem informasi alumni di SMKN 1 Jenangan Ponorogo telah memenuhi definisi kebutuhan pengguna (*requirement and definition*) di SMKN 1 Jenangan Ponorogo, berdasarkan hasil uji coba fungsionalitas yang memperoleh persentase dengan nilai 100%

2.13.2 Rancang Bangun Sistem Informasi Rawat Jalan Pada Klinik Waluya Sejati Abadi

(SESINDO, Vol. XIV, No. 22, Edisi November 2014)

Sistem ini dibuat oleh **Lusi Fajarita, Andhis Susilo Bekti, Habib Ahsan Syakir, dan Ruli Brainada** yang bertujuan untuk untuk membuat sistem pencatatan obat (pembelian-penjualan) yang bersifat komputerisasi. Manfaatnya adalah memudahkan dalam pencatatan rekam medik, mencatat resep obat dan pembuatan laporan-laporan.

Metode yang digunakan dalam perancangan sistem informasi ini adalah metode *waterfall*, dimana metode ini menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengkodean, pengujian, dan tahap pendukung (*support*). Alasan menggunakan metode ini yaitu tahapantahapannya dilakukan secara berurutan sehingga mudah untuk dipahami dan dilakukan.

Berdasarkan dari analisa dan perancangan sistem informasi rawat jalan pada Klinik Waluya Sejati Abadi, maka dapat diambil kesimpulan sebagai berikut:

1. Sistem informasi rawat jalan ini mampu mengolah data pasien secara cepat dan akurat, seperti pada saat menyimpan data dan mencetak keluaran.
2. Sistem informasi rawat jalan ini mampu menyimpan data rekam medik pasien setiap kali pasien melakukan pemeriksaan.
3. Sistem informasi rawat jalan ini mampu menghasilkan surat keterangan sehat, sakit, dan rujukan sesuai dengan keperluan pasien.
4. Sistem informasi rawat jalan ini mampu menyimpan data resep obat dan mencetak bukti pembayaran secara rinci.
5. Sistem informasi rawat jalan ini mampu menghasilkan laporan-laporan berupa data kunjungan pasien dan pendapatan klinik yang dapat dicetak per periode (tanggal, bulan dan tahun).

Dari kedua penelitian ilmiah yang ada diatas, dapat disimpulkan bahwa penggunaan sistem informasi berbasis *web* yang digunakan oleh SMKN 1 Jenangan Ponorogo dan pada Klinik Waluya Sejati Abadi ini mempermudah dalam memberikan informasi penting yang berkaitan dengan *user*, serta dapat menunjang proses bisnis yang ada di dalamnya dengan *user*, serta dapat membantu dalam mengolah data transaksi. Penggunaan metode *waterfall* yang dipakai oleh kedua sistem diatas akan digunakan sebagai metode pembangunan sistem untuk

perusahaan CV Sumber Makmur Pompa dan fungsi yang dapat diadopsi dari kedua sistem tersebut yaitu dapat menghasilkan laporan-laporan yang dapat dicetak per periode (tanggal, bulan, dan tahun).

