

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Pada penulisan skripsi ini, peneliti menggali informasi dari penelitian-penelitian sebelumnya sebagai bahan perbandingan, baik mengenai metode yang digunakan dan kesimpulan yang ada. Selain itu, peneliti juga menggali dari buku-buku maupun skripsi dalam rangka mendapatkan suatu informasi yang ada sebelumnya tentang teori tentang judul yang digunakan untuk memperoleh landasan teori ilmiah.

1. Penelitian yang dilakukan oleh Nur Ariesanto Ramdhan, Dimas Wahyudi (2019) Sistem Informasi Penerimaan Peserta Didik Baru (PPDB) Berbasis Web di SMP Negeri 1 Wanasari Brebes. Pada penelitian yang dilakukan oleh Nur Ariesanto Ramdhan dan Dimas Wahyudi pada tahun 2019 ini menceritakan pelaksanaan PPDB di SMP Negeri 1 Wanasari Brebes ini menjadi lebih transparan, akuntabel, akomodatif serta dapat mengurangi bahkan menghilangkan kecurangan-kecurangan yang terjadi pada pelaksanaan proses PPDB yang dilakukan secara manual, berkaitan dengan hal tersebut maka dirancang dan dibangun sebuah Sistem Informasi PPDB berbasis Web untuk memberikan informasi kepada masyarakat serta mempermudah panitia dalam proses pengolahan data peserta dan penyampaian informasi dan pelaporan data peserta pada SMP Negeri 1 Wanasari Brebes.
2. Penelitian yang dilakukan oleh Alifiannisa Alyahasna Wighneswara, Rofita Siti Musdalifah (2020) Rancang Bangun Aplikasi SIStem Penerimaan Peserta Didik Baru (PPDB) SMP Kabupaten Ponorogo Berbasis Website. Pada penelitian yang dilakukan oleh Alifiannisa Alyahasna Wighneswara dan Rofita Siti Musdalifah membangun Aplikasi Sistem Penerimaan Peserta didik Baru berbasis web, studi kasus penelitian ini pada SMP Kabupaten Ponorogo dengan adanya aplikasi PPDB ini diharapkan dapat membantu Dinas Pendidikan dan sekolah-sekolah dalam penerimaan peserta didik baru, memudahkan orang tua

siswa dalam mendaftarkan putra-putrinya ke jenjang Pendidikan SMP serta Mengoptimasi sistem dari cara manual menjadi terkomputerisasi dan terpusat.

3. Penelitian yang dilakukan oleh Regi Witanto, Hanhan Hanafiah Solihim (2016) Perancangan sistem informasi penerimaan peserta didik baru berbasis web (studi kasus : SMP Plus Babussalam Bandung). Pada penelitian ini diceritakan bahwa proses penerimaan siswa baru masih menggunakan sistem konvensional, proses administrasi cenderung lambat dan masih menggunakan arsip dalam bentuk fisik sehingga rentan mengalami kerusakan bahkan hilang. Dengan adanya sistem informasi berbasis web dapat mengatasi pengolahan data dan administrasi calon siswa sehingga memberi kemudahan bagi panitia penerimaan siswa baru di SMP Plus Babussalam.

2.2 Penerimaan Peserta Didik Baru (PPDB)

Penerimaan peserta didik baru merupakan proses pendaftaran dan pelayanan kepada siswa yang baru masuk sekolah, setelah mereka memenuhi persyaratan yang ditetapkan oleh sekolah tersebut. Dalam penerimaan peserta didik baru ini kepala sekolah perlu membentuk panitia penerimaan peserta didik baru. Rekrutmen peserta didik di sebuah lembaga pendidikan pada hakikatnya merupakan proses pencarian, menarik peserta didik untuk sekolah di lembaga yang bersangkutan.

Penerimaan peserta didik baru bukan sekedar menerima peserta didik yang ingin memasuki suatu sekolah, melainkan juga menyeleksi apakah calon-calon peserta didik ini telah memenuhi syarat yang telah ditetapkan. Ada beberapa hal yang perlu diperhatikan dalam pengelolaan penerimaan peserta didik baru masalah panitia, persyaratan calon, pendaftaran, tes, seleksi, pengumuman hasil seleksi dan orientasi peserta didik baru. Tujuan penerimaan peserta didik baru ini adalah untuk menghasilkan yang kompeten sesuai dengan standar kompetensi lulusan, serta mampu bersaing dan mampu berperan aktif dalam menjaga kelangsungan hidup.

Menurut Ali Imron (2012) kebijakan operasional penerimaan peserta didik baru, memuat beberapa aturan mengenai jumlah peserta atau kuota penerimaan peserta didik baru yang akan diterima disuatu lembaga sekolah. Namun penentuan jumlah kuota peserta didik tersebut juga didasarkan pada kondisi atau kenyataan-

kenyataan yang ada disekolah seperti faktor-faktor kondisi sekolah. 5 Faktor kondisi sekolah tersebut

misalnya: (1) daya tampung kelas baru, (2) kriteria siswa yang dapat diterima, (3) anggaran yang tersedia, (4) sarana dan prasarana, (5) tenaga kependidikan yang tersedia, (6) jumlah peserta didik yang tinggal di kelas satu.Sekolah.

2.3 Sekolah

Menurut KBBI pengertian sekolah yaitu salah satu bangunan atau lembaga yang digunakan untuk melaksanakan kegiatan belajar dan mengajar dengan berbagai jenjang pendidikan. Jenjang pendidikan tersebut terdiri atas SD atau MI, SLTP atau Mts, dan SLTA atau MA.

Lembaga ini memberikan berbagai macam ilmu yang tentunya dapat bermanfaat bagi anak didiknya di masa sekarang maupun masa yang akan datang. Sebab itu, sekolah menjadi kebutuhan yang penting sebagai upaya untuk mencerdaskan generasi bangsa.

Menurut Yusran Pora sekolah tidak hanya sekedar tempat untuk menambah wawasan dan pengetahuan sebanyak-banyaknya. Akan tetapi hal terpenting adalah, sekolah menjadi tempat bagi para guru dan siswa untuk belajar bersama, mengamati sesuatu yang ada di sekeliling bersama-sama, dan sekolah juga menjadi tempat untuk para siswa membentuk jati dirinya. Sekolah juga dapat membentuk karakter peserta didik agar paham bagaimana cara bersikap terhadap sesama manusia maupun dengan lingkungan..

2.4 Aplikasi

Yuhefizar (2012) mengatakan bahwa aplikasi merupakan program yang dikembangkan untuk memenuhi kebutuhan pengguna dalam menjalankan pekerjaan tertentu. aplikasi merupakan suatu unit perangkat lunak yang dibuat untuk melayani kebutuhan akan beberapa aktifitas seperti perniagaan, game, pelayanan masyarakat, periklanan, atau semua proses yang hampir dilakukan oleh manusia. Berdasarkan dua definisi diatas, dapat diambil kesimpulan bahwa aplikasi merupakan kumpulan program yang dikembangkan sesuai dengan kebutuhan pengguna untuk melayani aktifitas yang dilakukan oleh pengguna.

2.5 Website

Menurut (Susanti, 2016), Website adalah suatu kumpulan yang kompleks dalam suatu jaringan komputer yang cukup besar maupun kecil yang dapat saling berkomunikasi dengan menggunakan jaringan yang terdapat di seluruh dunia. Manusia pada umumnya dapat aktif dalam berpartisipasi, sehingga website tersebut dapat memberikan informasi yang tentunya akan sangat berharga bagi para penggunanya. Secara umum, website dapat dipahami sebagai sekumpulan halaman yang didalamnya terdapat berbagai laman yang menyediakan informasi yang luas melalui computerized seperti gambar, teks, maupun animasi yang disediakan oleh masing-masing web, sehingga dapat dengan mudah diakses oleh pengguna diseluruh dunia yang telah memiliki koneksi jaringan internet.

2.6 HTML

Menurut (Wardana, 2016). *Hypertext markup language* (HTML) merupakan bahasa pemrograman dasar untuk mengelola *website*. Akan tetapi HTML hanya terbatas pada pembuatan *website* statis (*website* yang tidak dapat berinteraksi aktif dengan pengguna). Maka dari itu HTML biasa dikombinasikan dengan Bahasa pemrograman web lainnya.

2.7 PHP

Menurut (MADCOM, 2016) “PHP (*Hypertext Preprocessor*) adalah bahasa script yang dapat ditanamkan atau disisipkan ke dalam HTML. PHP banyak dipakai untuk membuat program situs web dinamis”. PHP dapat digunakan dengan gratis dan bersifat *Open Source*. PHP dirilis dalam lisensi *PHP license*. Untuk membuat program PHP kita diharuskan untuk menginstal *web server* terlebih dahulu. PHP mendukung komentar seperti pada bahasa ‘C’, ‘C++’, dan Unix shell-style. (Perl style).

2.8 CSS

CSS adalah suatu Bahasa pemrograman web yang digunakan untuk mengendalikan dan membangun berbagai komponen dalam web sehingga tampilan

web akan lebih rapi, terstruktur, dan seragam. CSS merupakan salah satu kode pemrograman yang bertujuan untuk menghias dan mengatur gaya tampilan atau layout halaman web agar lebih menarik. CSS adalah sebuah teknologi internet yang direkomendasikan oleh *World Wide Web Consortium* atau W3C pada tahun 1996 (Wahyudi, 2017).

2.9 XAMPP

Menurut (MADCOM, 2016) “Xampp adalah sebuah paket kumpulan software yang terdiri dari Apache, MySQL, PhpMyAdmin, PHP, Perl, Filezilla, dan lain.” Xampp berfungsi untuk memudahkan instalasi lingkungan PHP, di mana biasanya lingkungan pengembangan web memerlukan PHP, Apache, MySQL dan PhpMyAdmin.

2.10 JavaScript

JavaScript adalah bahasa script yang disisipkan pada kode HTML dan diproses di sisi klien yaitu sering disebut client side. Bahasa ini menjadikan dokumen HTML menjadi semakin luas (Agusriandi, 2018).

2.11 Bootstrap

Menurut Tectale, (2012) Bootstrap adalah sebuah library framework CSS yang di buat khusus untuk bagian pengembangan front-end website. bootstrap merupakan salah satu framework HTML, CSS dan javascript yang paling populer di kalangan web developer. pada saat ini hampir semua web developer telah menggunakan bootstrap untuk membuat tampilan front-end menjadi lebih mudah dan sangat cepat. karena anda hanya perlu menambahkan class-class tertentu untuk misalnya membuat tombol, grid, navigasi dan lainnya.

Bootstrap telah menyediakan kumpulan komponen class interface dasar yang telah di rancang sedemikian rupa untuk menciptakan tampilan yang menarik, bersih dan ringan. selain komponen class interface, bootstrap juga memiliki fitur grid yang berfungsi untuk mengatur layout pada halaman website yang bisa digunakan dengan sangat mudah dan cepat. dengan menggunakan bootstrap kita juga di beri keleluasaan dalam mengembangkan tampilan website yang menggunakan bootstrap

yaitu dengan cara mengubah tampilan bootstrap dengan menambahkan class dan CSS sendiri.

2.12 Data Flow Diagram (DFD)

Menurut Pahlevy (2010), *Data Flow Diagram* (DFD) adalah suatu diagram yang menggunakan notasi-notasi untuk menggambarkan arus dari data sistem, yang penggunaannya sangat membantu untuk memahami sistem secara logika, tersruktur dan jelas. Selanjutnya, menurut Wijaya (2007), *Data Flow Diagram* adalah gambaran grafis yang memperlihatkan aliran data dari sumbernya dalam obyek kemudian melewati suatu proses yang mentransformasikan ke tujuan yang lain, yang ada pada objek lain.

DFD ini merupakan alat perancangan sistem yang berorientasi pada alur data dengan konsep dekomposisi dapat digunakan untuk penggambaran analisa maupun rancangan sistem yang mudah dikomunikasikan oleh profesional sistem kepada pemakai maupun pembuat program (Pahlevy. 2010). Beberapa simbol yang terdapat pada Data Flow Diagram (DFD) menurut Al Bahra (2013) sebagai berikut:

a. Kesatuan Luar (External Entity)

Sesuatu yang berada di luar sistem, tetapi memberikan data ke dalam sistem atau memberikan data dari sistem . External Entity tidak termasuk bagian dari sistem. Bila sistem informasi dirancang untuk satu bagian atau departemen maka bagian lain yang masih terkait menjadi external entity. Dapat dilihat pada Gambar 2.1



Gambar 2.1 Simbol Entitas

b. Arus Data (Data Flow)

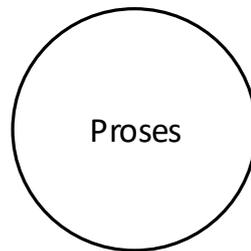
Arus data merupakan tempat mengalirnya informasi dan digambarkan dengan garis yang menghubungkan komponen dari sistem. Arus data ditunjukkan dengan arah panah dan garis diberi nama atas arus data yang mengalir. Arus data ini mengalir di antara proses, data store dan menunjukkan arus data dari data yang berupa masukan untuk sistem atau hasil proses sistem. Dapat dilihat pada Gambar 2.2



Gambar 2.2 Simbol Arus Data

c. Proses (Process)

Proses merupakan apa yang dikerjakan oleh sistem. Proses dapat mengolah data atau aliran data masuk menjadi aliran data ke luar. Proses berfungsi mengubah satu atau beberapa data masukan menjadi satu atau beberapa data keluaran sesuai dengan spesifikasi yang diinginkan. Setiap proses memiliki satu atau beberapa masukan serta menghasilkan satu atau beberapa keluaran. Dapat dilihat pada Gambar 2.3



Gambar 2.3 Simbol Proses

d. Simpanan Data (Data Store)

Simpanan data merupakan tempat penyimpanan data pengikat data yang ada dalam sistem. Data store disimbolkan dengan sepasang dua garis atau dua garis dengan salah satu sisi samping terbuka. Proses dapat mengambil data dari atau memberikan data ke *database*. Dapat dilihat pada Gambar 2.4



Gambar 2.4 Simbol Simpanan Data

2.13 Entity Relationship Diagram (ERD)

Menurut Al-Bahra bin Ladjamudin (2013:142) *Entity Relationship Diagram* (ERD) adalah suatu model jaringan yang menggunakan susunan data yang disimpan dalam sistem secara abstrak. Jadi, jelaslah bahwa ERD ini berbeda dengan DFD yang merupakan suatu model jaringan fungsi yang akan dilaksanakan oleh

sistem, sedangkan ERD merupakan model jaringan data yang menekankan pada struktur–struktur dan relationship data. Elemen–elemen dalam *Diagram Hubungan Entitas* meliputi:

a. *Entity*

Pada E-R *Diagram*, *Entity* dapat digambarkan dengan sebuah bentuk persegi panjang. *Entity* adalah sesuatu apa saja yang ada di dalam sistem, nyata maupun abstrak dimana data tersimpan atau dimana terdapat data. Entitas diberi nama dengan kata benda dan dapat dikelompokkan dalam empat jenis nama, yaitu orang, benda, lokasi, kejadian (terdapat unsur waktu di dalamnya).

b. *Relationship*

Pada E-R *Diagram*, *Relationship* dapat digambarkan dengan sebuah bentuk belah ketupat. *Relationship* adalah hubungan alamiah yang terjadi antara entitas. Pada umumnya penghubung (*Relationship*) diberi nama dengan kata kerja dasar, sehingga memudahkan untuk melakukan pembacaan relasinya (bisa dengan kalimat aktif atau pasif).

Penggambaran hubungan yang terjadi adalah sebuah bentuk belah ketupat dihubungkan dengan dua bentuk empat persegi panjang.

1. *Relationship Degree*

Relationship Degree atau Derajat *Relationship* adalah jumlah entitas yang berpartisipasi dalam suatu *Relationship*.

a. *Unary Degree* (Derajat Satu)



Gambar 2.5 *Unary Degree*

Sumber: Al-Bahra bin Ladjamudin (2013:142)

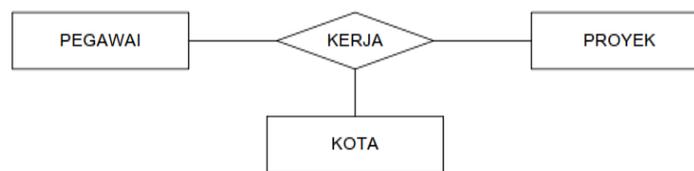
b. *Binary Degree* (Derajat Dua)



Gambar 2.6 *Binary Degree*

Sumber: Al-Bahra bin Ladjamudin (2013:142)

c. *Ternary Degree* (Derajat Tiga)



Gambar 2.7 *Ternary Degree*

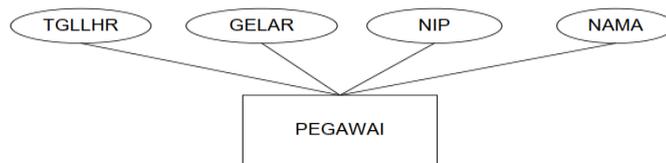
Sumber: Al-Bahra bin Ladjamudin (2013:142)

2. Atribut

Secara umum atribut adalah sifat atau karakteristik dari tiap entitas maupun tiap *Relationship*. Atribut adalah sesuatu yang menjelaskan apa sebenarnya yang dimaksud entitas maupun *Relationship*, sehingga sering dikatakan atribut adalah elemen dari setiap entitas dan *Relationship*. Atribut *Value* atau Atribut Nilai adalah suatu *occurrence* tertentu dari sebuah atribut di dalam suatu *entity* atau *Relationship*.

Ada empat jenis-jenis atribut:

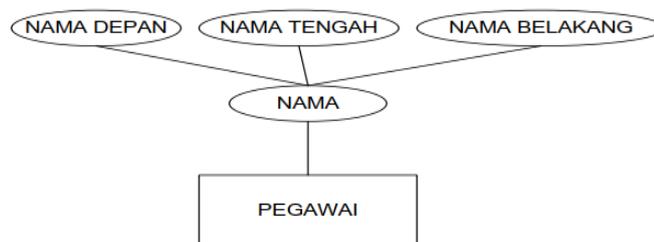
1. *Key* adalah atribut yang digunakan untuk menentukan suatu *entity* secara unik.
2. Atribut *Simple* adalah atribut yang bernilai tunggal.
3. Atribut *Multivalued* adalah atribut yang memiliki sekelompok nilai untuk setiap *instan entity*



Gambar 2.8 *Atribut Multivalue*

Sumber: Al-Bahra bin Ladjamudin (2013:142)

4. Atribut *Composite* adalah suatu atribut yang terdiri dari beberapa atribut yang lebih kecil yang mempunyai arti tertentu.



Gambar 2.9 *Atribut Composite*

Sumber: Al-Bahra bin Ladjamudin (2013:142)

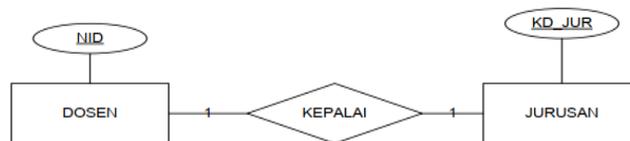
3. Kardinalitas

Kardinalitas Relasi menunjukkan jumlah maksimum tupel yang dapat berelasi dengan entitas pada entitas yang lain. Dari sejumlah kemungkinan banyaknya hubungan antra entitas tersebut, Kardinalitas Relasi merujuk kepada hubungan maksimum yang terjadi dari entitas yang satu ke entitas yang lain dan begitu juga sebaliknya. Terdapat tiga macam kardinalitas relasi, yaitu:

a. *One to One*

Tingkat hubungan satu ke satu, dinyatakan dengan satu kejadian pada entitas pertama, hanya mempunyai satu hubungan dengan satu kejadian pada entitas kedua dan sebaliknya. Yang berarti setiap tupel pada entitas A berhubungan dengan paling banyak satu tupel pada entitas B, dan begitu juga sebaliknya setiap tupel pada entitas B berhubungan dengan paling banyak satu tupel pada entitas A.

Contoh: Adanya relasi antara entitas Dosen dengan entitas Jurusan. Relasinya diberi nama 'Kepala'. Pada relasi ini, setiap dosen paling banyak mengepalai satu jurusan (walaupun memang tidak semua dosen yang menjadi ketua jurusan). Dan setiap jurusan dikepalai oleh paling banyak satu orang dosen.



Gambar 2.10 Diagram Kardinalitas *One to One*

Sumber: Al-Bahra bin Ladjamudin (2013:142)

b. *One to Many* atau *Many to One*

Tingkat hubungan satu ke banyak adalah sama dengan banyak ke satu. Tergantung dari arah mana hubungan tersebut dilihat. Untuk satu kejadian pada entitas yang pertama dapat mempunyai banyak hubungan dengan kejadian pada entitas yang kedua. Sebaliknya, satu kejadian pada entitas yang kedua hanya dapat mempunyai satu hubungan dengan satu kejadian pada entitas yang pertama

c. *One to Many*

Satu tupel pada entitas A dapat berhubungan dengan banyak tupel pada entitas B, tetapi tidak sebaliknya, di mana setiap tupel pada entitas B, berhubungan dengan paling banyak satu tupel pada entitas A.

Contoh: Adanya relasi antara entitas Dosen dengan entitas Kuliah. Relasinya diberi nama 'Ajar'. Setiap dosen dapat mengajar lebih dari satu mata kuliah, sedang setiap mata kuliah hanya oleh paling banyak satu orang dosen.



Gambar 2.11 Diagram Kardinalitas *One to Many*

Sumber: Al-Bahra bin Ladjamudin (2013:142)

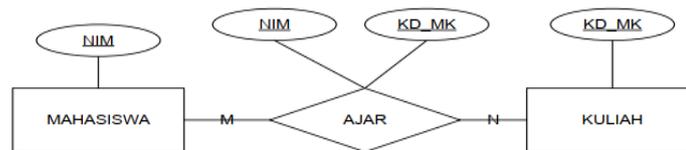
d. *Many to One*

Setiap tupel pada entitas A dapat berhubungan dengan paling banyak satu tupel pada entitas B, tetapi tidak sebaliknya, di mana setiap tupel pada entitas A berhubungan dengan paling banyak satu tupel pada entitas B.

e. *Many to Many*

Tingkat hubungan banyak ke banyak terjadi jika setiap kejadian pada sebuah entitas akan mempunyai banyak hubungan dengan kejadian pada entitas lainnya. Baik dilihat dari sisi entitas yang pertama, maupun dilihat dari sisi yang kedua. Yang berarti setiap tupel pada entitas A dapat berhubungan dengan banyak tupel pada entitas B, dan demikian juga sebaliknya, dimana setiap tupel pada entitas B dapat berhubungan dengan banyak tupel pada entitas A.

Contoh: Adanya relasi entitas Mahasiswa dengan entitas Kuliah. Relasinya diberi nama 'Belajar'. Setiap mahasiswa dapat mempelajari lebih dari satu mata kuliah. Demikian juga sebaliknya, setiap mata kuliah dapat dipelajari oleh lebih dari satu orang mahasiswa.

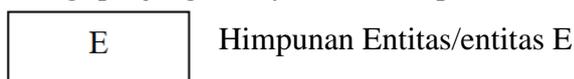


Gambar 2.12 Diagram Kardinalitas *Many to Many*

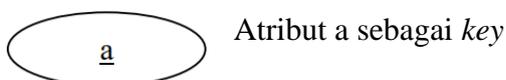
Sumber: Al-Bahra bin Ladjamudin (2013:142)

Notasi-notasi simbolik di dalam Diagram E-R:

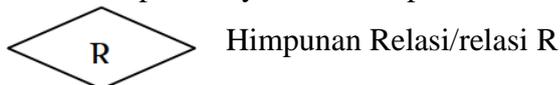
1. Persegi panjang, menyatakan Himpunan Entitas/entitas.



2. Lingkaran/Elip, menyatakan Atribut (Atribut yang berfungsi sebagai key digaris bawah).



3. Belah ketupat menyatakan Himpunan Relasi/relasi.



4. Garis, sebagai penghubung antara Himpunan Relasi dengan Himpunan Entitas dan Himpunan Entitas dengan Atributnya.



Langkah-langkah teknis untuk menghasilkan *Entity Relationship Diagram* (ERD):

- Melengkapi himpunan entitas dan himpunan relasi dengan atribut-atribut deskriptif (*non key*).
- Mengidentifikasi dan menetapkan seluruh entitas yang akan terlibat.
- Menentukan atribut-atribut key (*primary key*) dari masing-masing entitas.
- Mengidentifikasi dan menetapkan seluruh relasi diantara entitas-entitas yang ada beserta *foreign-key*-nya (jika terjadi kardinalitas relasi *One to Many* atau *Many to Many*)
- Menentukan derajat/kardinalitas relasi untuk setiap relasi.

2.14 Systems Development Life Cycle (SDLC)

Menurut Ladjamudin, Al-Bahra (2013:38) Daur hidup pengembangan sistem/SDLC berfungsi untuk menggambarkan tahapan-tahapan utama dan langkah-langkah dari setiap tahapan yang secara garis besar terbagi dalam tiga kegiatan utama, yaitu :

1. Analisis

Tahapan analisis digunakan oleh analis sistem untuk membuat keputusan. Apabila sistem saat ini mempunyai masalah atau sudah tidak berfungsi secara baik, dan hasil analisisnya digunakan sebagai dasar untuk memperbaiki sistem.

Kegiatan yang dilakukan dalam tahapan analisis ini adalah deteksi masalah, penelitian/investigasi awal, analisa kebutuhan sistem, mensortir kebutuhan sistem, dan memilih sistem yang baik.

2. Perancangan/Desain

Tahapan perancangan (Design) memiliki tujuan untuk mendesain sistem baru yang dapat menyelesaikan masalah-masalah yang dihadapi perusahaan yang

diperoleh dari pilihan alternatif sistem yang terbaik. Kegiatan yang dilakukan dalam tahapan perancangan ini meliputi masukan, keluaran dan file.

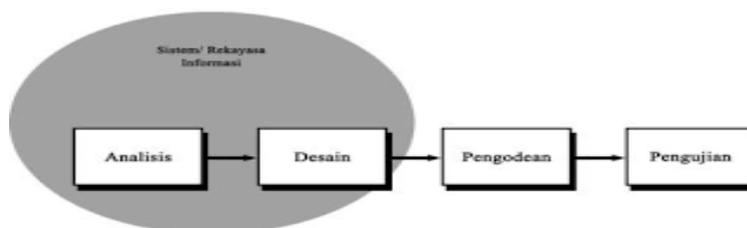
3. Implementasi

Tahap implementasi memiliki beberapa tujuan, yaitu untuk melakukan kegiatan spesifikasi rancangan logika ke dalam kegiatan yang sebenarnya dari sistem informasi yang akan dibangunnya atau dikembangkannya, lalu mengimplementasi sistem yang baru dapat berjalan secara optimal. Kegiatan yang dilakukan dalam tahap implementasi ini adalah pembuatan program dan test data, pelatihan, dan pergantian sistem.

2.15 Waterfall

Menurut Rosa dan Shalahuddin (2018:28), “Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain pengodean, pengujian dan tahap pendukung (support)”.

Fase-fase dalam *Waterfall Model* menurut referensi Rosa dan Shalahuddin (2018:29).



Gambar 2.13 Fase Waterfall

a. Analisis Kebutuhan Perangkat Lunak

Proses pengumpulan kebutuhan dilakukan secara insentif untuk menspesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh user. Spesifikasi kebutuhan perangkat lunak pada tahap ini perlu untuk didokumentasikan.

b. Desain

Desain perangkat lunak adalah proses multi langkah yang fokus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antarmuka, dan prosedur pengkodean. Tahap ini mentranslasi kebutuhan perangkat lunak dari tahap analisis kebutuhan ke representasi desain

agar dapat diimplementasikan menjadi program pada tahap selanjutnya. Desain perangkat lunak yang dihasilkan pada tahap ini juga perlu didokumentasikan.

c. Pembuatan Kode Program

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

d. Pengujian

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

e. Pendukung (support) atau Pemeliharaan (maintenance)

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

2.16 Pengujian Aplikasi

2.16.1 Pengujian Black Box

Black Box testing berfokus pada persyaratan fungsional perangkat lunak yang memungkinkan *engineers* untuk memperoleh set kondisi input yang sepenuhnya akan melaksanakan persyaratan fungsional untuk sebuah program (Pressman, 2010).

Black Box Testing berfokus pada spesifikasi fungsional dari perangkat lunak. Tester dapat mendefinisikan kumpulan kondisi input dan melakukan pengetesan pada spesifikasi fungsional program. *Black Box Testing* bukanlah solusi alternatif dari White Box Testing tapi lebih merupakan pelengkap untuk menguji hal-hal yang tidak dicakup oleh White Box Testing. Black Box Testing cenderung untuk menemukan hal-hal berikut (Mustaqbal dan Firdaus, 2015).

1. Fungsi yang tidak benar atau tidak ada

2. Kesalahan antarmuka (*interface errors*)
3. Kesalahan pada struktur data dan akses basis data
4. Kesalahan performansi (*performance errors*)
5. Kesalahan inisialisasi dan terminasi

2.16.2 Pengujian User Acceptance Test (UAT)

Menurut Perry, William E, User Acceptance Testing (UAT) merupakan pengujian yang dilakukan oleh end-user dimana user tersebut adalah staff/karyawan perusahaan yang langsung berinteraksi dengan sistem dan dilakukan verifikasi apakah fungsi yang ada telah berjalan sesuai dengan kebutuhan/fungsinya. Setelah dilakukan sistem testing, acceptance testing menyatakan bahwa sistem perangkat lunak memenuhi persyaratan.

Pengujian penerimaan pengguna (UAT) adalah fase terakhir dari proses pengujian perangkat lunak. Selama UAT, perangkat lunak perangkat lunak diuji untuk memastikan tugas-tugas apakah sudah sesuai dengan spesifikasinya. UAT adalah salah satu prosedur proyek perangkat lunak final dan paling penting yang harus terjadi sebelum perangkat lunak tersebut dikembangkan dan diluncurkan ke pasar. UAT juga dikenal sebagai pengujian beta, pengujian aplikasi atau pengujian pengguna akhir.

Menurut Black, acceptance testing biasanya berusaha menunjukkan bahwa sistem telah memenuhi persyaratan-persyaratan tertentu. Pada pengembangan software dan hardware komersial, acceptance test biasanya disebut juga "alpha tests" (yang dilakukan oleh pengguna in-house) dan "beta tests" (yang dilakukan oleh pengguna yang sedang menggunakan atau akan menggunakan sistem tersebut). Alpha dan beta test biasanya juga menunjukkan bahwa produk sudah siap untuk dijual atau dipasarkan. Acceptance testing mencakup data, environment dan skenario yang sama atau hampir sama pada saat live yang biasanya berfokus pada skenario penggunaan produk tertentu.

Alpha dan beta test biasanya juga menunjukkan bahwa produk sudah siap untuk dijual atau dipasarkan. Pengguna tes biasanya dilakukan oleh klien atau pengguna akhir, dan tidak fokus pada identifikasi masalah sederhana seperti kesalahan ejaan dan cacat showstopper atau crash perangkat lunak tapi juga masalah – masalah lainnya. Hasil tes ini memberikan kepercayaan kepada klien tentang bagaimana sistem akan siap di produksi. Pada pengembangan perangkat lunak, user acceptance testing (UAT) juga disebut pengujian beta (beta testing), pengujian aplikasi (application testing) dan pengujian pengguna akhir (end user testing) adalah tahapan pengembangan perangkat lunak ketika perangkat lunak diuji pada dunia nyata.

Proses UAT memastikan bahwa aplikasi sistem rekomendasi pelayanan jasa yang peneliti implementasikan tersebut akan memberi solusi, memenuhi harapan pengguna dan bekerja seperti yang diharapkan serta meyakinkan user atau pelanggan aplikasi tersebut apakah sistem bisa diterima dengan baik atau tidak. Dari definisi di atas dapat disimpulkan bahwa user acceptance testing adalah pengujian yang dilakukan oleh pengguna dari sebuah sistem untuk memastikan fungsi-fungsi yang ada pada sistem tersebut telah berjalan dengan baik dan sesuai dengan kebutuhan pengguna.