

BAB II

LANDASAN TEORI

2.1. Konsep Dasar Sistem

2.1.1. Pengertian Sistem

Pengertian Sistem menurut McLeod yang dikutip oleh Asmara (2017) adalah sebagai berikut: *“A sistem is a group of elements that are integrated with the common porpose of achieving an objective”*. Sistem adalah sekelompok elemen yang terintegritasi dengan maksud yang sama untuk mencapai suatu tujuan.(Asmara, 2017)

Menurut (Utomo, Sakuroh, Yulinar, 2018: 108), “Sistem adalah suatu himpunan suatu benda nyata atau abstrak (*a set of thing*) yang berdiri dari bagian-bagian atau komponen-komponen yang saling berkaitan, berhubungan, berketergantungan,, saling mendukung yang secara keseluruhan bersatu dalam satu kesatuan (*unity*) untuk mencapai tujuan tertentu secara efisien dan efektif”

Menurut Nugroho dalam (Kaleb, 2019), sistem dapat didefinisikan sebagai :

“Sekelompok elemen yang terintegrasi dengan maksud yang sama untuk mencapai suatu tujuan. Hal pertama yang perlu diperhatikan dalam suatu sistem adalah elemen-elemennya. Tentunya setiap sistem memiliki elemen-elemennya sendiri, yang kombinasinya berbeda antara sistem yang satu dengan sistem yang lain. Namun demikian, susunan dasarnya tetap sama.”

Menurut James Havery yang di kutip dalam jurnal (Ismael, 2019) “Sistem adalah prosedur logis dan rasional untuk merancang suatu rangkaian komponen yang berhubungan satu dengan yang lainnya dengan maksud untuk berfungsi sebagai suatu kesatuan dalam usaha mencapai suatu tujuan yang telah ditentukan.”

Berdasarkan beberapa pendapat yang dikemukakan diatas dapat ditarik kesimpulan bahwa “Sistem merupakan kumpulan komponen-komponen atau subsistem yang saling berkaitan dan bekerja sama untuk mencapai suatu tujuan”.

2.1.2. Karakteristik Sistem

Menurut (Daulay, 2018) mengenai karakteristik sistem yang menyatakan bahwa : “Suatu sistem mempunyai karakteristik atau sifat-sifat tertentu, yaitu mempunyai komponen-komponen, batasan sistem, lingkungan luar sistem, penghubung, masukan, pengolahan, sasaran & tujuan, keluaran, dan umpan balik”.

Adapun penjelasan dari masing-masing karakteristik sistem menurut (Daulay, 2018) adalah sebagai berikut:

1. **Komponen Sistem (*Components System*)**

Suatu sistem tidak berada dilingkungan yang kosong, tetapi sebuah sistem berada dan berfungsi didalam lingkungan yang berisi sitem lainnya. Sistem terdiri dari sejumlah komponen yang saling berinteraksi, bekerja sama membentuk satu kesatuan. Apabila suatu sistem merupakan salah satu dari komponen sistem lain yang lebih besar, maka disebut dengan subsitem.

2. **Batasan Sistem (*Limitation System*)**

Batasan sistem merupakan pembatas atau pemisah antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya.

3. **Lingkungan Luar Sistem (*Environment*)**

Lingkungan luar adalah diluar batas sistem yang dapat mempengaruhi operasi sistem, baik pengaruh yang menguntungkan atau yang merugikan. Pengaruh yang menguntungkan ini tentunya harus dijaga sehingga dapat mendukung kelangsungan operasi sebuah sistem. Sedangkan lingkungan yang merugikan harus ditahan dan dikendalikan agar tidak mengganggu kelangsungan sistem.

4. Penghubung (*Interface*)

Interface merupakan media penghubung antara satu subsistem dengan subsistem lainnya. Penghubung inilah yang akan menjadi media yang digunakan data dari masukan (*input*) dan keluaran (*output*). Dengan adanya penghubung, suatu subsistem dapat berinteraksi dan berintegrasi dengan subsistem yang lainnya untuk membentuk satu kesatuan.

5. Masukan (*Input*)

Masukan (*input*) merupakan energi yang dimasukkan kedalam sistem. Masukan dapat berupa masukan perawatan (*maintenance input*), yaitu bahan yang dimasukkan agar sistem tersebut beroperasi dan masukan sinyal, yaitu masukan yang diproses untuk mendapatkan keluaran (*output*).

6. Pengolahan (*Processing*)

Pengolahan merupakan bagian yang melakukan perubahan dari masukan menjadi keluaran yang diinginkan.

7. Sasaran (*Objective*) dan Tujuan (*Goal*)

Suatu sistem pasti mempunyai sasaran atau tujuan. Apabila sistem tidak mempunyai sasaran atau tujuan, maka sistem operasi tidak akan ada gunanya. Tujuan inilah yang mengarahkan suatu sistem. Tanpa adanya tujuan, sistem tidak akan terarah dan terkendali.

8. Keluaran (*Output*)

Keluaran (*output*) merupakan hasil dari pemrosesan, keluaran dapat berupa informasi sebagai masukan pada sistem lain atau hanya sebagai sisa pembuangan.

9. Umpan Balik (*Feed Back*)

Umpan balik diperlukan oleh bagian kontrol sistem untuk mengecek atau meneliti terjadinya penyimpangan proses dalam sistem dan mengembalikannya kedalam kondisi normal.

2.1.3. Jenis-Jenis Sistem

1. Sistem Tertutup (*Close System*)

Sistem Tertutup (*Close System*) yaitu sistem yang secara total terisolasi dan lingkungannya. Tidak ada penghubung dengan eksternal, sehingga sistem ini tidak memiliki pengaruh terhadap dan dipengaruhi oleh lingkungan yang berada diluar batas sistem.

2. Sistem Relatif Tertutup (*Relatively Close System*)

Sistem Relatif Tertutup (*Relatively Close System*) yaitu sistem yang berinteraksi dengan lingkungan secara terkendali. Sistem semacam ini memiliki penghubung yang menghubungkan sistem dengan lingkungan dan mengendalikan pengaruh lingkungan terhadap proses yang dilakukan oleh sistem. Interaksinya berupa *input* jika *input* tersebut diperoleh dari lingkungan, dan *output* jika *output* tersebut ditujukan kepada pihak yang berada diluar batas sistem. Sistem yang dirancang dengan baik, akan membatasi, bukan mengeliminasi, pengaruh dari luar sistem.

3. Sistem Terbuka (*Open System*)

Sistem Terbuka (*Open System*) yaitu sistem yang berinteraksi dengan lingkungan secara terkendali. Disamping memperoleh *input* dari lingkungan, dan memberikan *output* bagi lingkungan, sistem terbuka juga memperoleh gangguan, atau *input* yang tidak terkendali yang akan mempengaruhi proses dalam sistem. Sistem yang dirancang dengan baik dapat meminimalkan

gangguan ini, dengan cara melakukan antisipasi terhadap kemungkinan munculnya gangguan dari lingkungan dan selanjutnya menciptakan proses dan cara-cara menanggulangi gangguan.

4. Sistem Umpan Balik (*Feedback Control*)

Sistem Umpan Balik (*Feedback Control*) yaitu sistem yang menggunakan sebagai *output* menjadi salah satu *input* untuk proses yang sama di masa berikutnya. Sebuah sistem dapat dirancang untuk memberikan umpan balik guna membantu sistem tersebut mencapai tujuan. Salah satu contoh sistem yang dirancang untuk tujuan melakukan pengendalian adalah sistem pelaporan pertanggung jawaban, dimana sistem ini menghasilkan laporan pelaksanaan kegiatan yang berisi perbandingan antara target dengan realisasi kegiatan. Atas dasar informasi dalam laporan tersebut, manajemen dapat menggunakannya sebagai umpan balik guna membuat rencana yang lebih baik di masa mendatang.

2.1.4. Pengertian Informasi

Menurut Ladjamudin dalam (Rudi Hermawan, Arief Hidayat, 2016) berpendapat “Informasi adalah data yang telah diolah menjadi bentuk yang lebih berarti dan berguna bagi penerimanya untuk mengambil keputusan masa kini maupun masa yang akan datang”.

Menurut Sutanta dalam (Apriyansyah, 2019) “Informasi merupakan hasil pengolahan data sehingga menjadi bentuk yang penting bagi penerimanya dan mempunyai kegunaan sebagai dasar dalam pengambilan keputusan yang dapat dirasakan akibatnya secara langsung saat itu juga atau secara tidak langsung pada saat mendatang.”

Menurut Sutabri dalam (Kaleb, 2019) “Informasi adalah data yang telah diklasifikasikan atau diolah atau diinterpretasikan untuk digunakan dalam proses pengambilan keputusan”

Menurut (Ramadhan, Sarkum & Purnama, 2019) mengemukakan bahwa informasi adalah “Data yang telah diproses sehingga memiliki arti dan nilai kepada penerima informasi tersebut. Selain itu juga disebutkan, informasi merupakan suatu pengetahuan yang diperoleh dari hasil pengolahan data-data sehingga dapat menjadi sebuah kesimpulan yang dapat membantu dalam proses pengambilan keputusan.

2.1.5. Karakteristik Informasi

Menurut Krismiaji (2015:15), “Suatu informasi semakin tinggi kualitas informasi yang tersedia bagi para pembuat keputusan, semakin baik keputusan yang dihasilkan”. Agar bermanfaat, informasi harus memiliki kualitas atau karakteristik sebagai berikut:

1. Relevan

Menambah pengetahuan atau nilai bagi para pembuat keputusan, dengan cara mengurangi ketidak pastian, menaikkan kemampuan untuk memprediksi, atau menegaskan/membenarkan ekspektasi semula.

2. Dapat Dipercaya

Bebas dari kesalahan atau bias dan secara akurat menggambarkan kejadian atau aktivitas organisasi.

3. Lengkap

Tidak menghilangkan data penting yang dibutuhkan oleh para pemakai.

4. Tepat Waktu

Disajikan pada saat yang tepat untuk mempengaruhi proses pembuatan keputusan.

5. Mudah dipahami

Disajikan dalam format yang mudah dimengerti.

6. Dapat Diuji Kebenarannya

Memungkinkan dua orang yang kompeten untuk menghasilkan informasi yang sama secara independen. Berdasarkan uraian diatas, penulis menyimpulkan bahwa informasi adalah data yang diolah menjadi bentuk yang lebih berguna bagi para pemakainya.

2.1.6. Pengertian Sistem Informasi

Menurut James O'Brien dalam (Agus Irawan & Hasna, 2016) "Sistem informasi adalah suatu kombinasi teratur apapun dari *people* (orang), *hardware* (perangkat keras), *software* (perangkat lunak), *computer networks and data communications* (jaringan komunikasi), dan *database* (basis data) yang mengumpulkan, mengubah dan menyebarkan informasi didalam suatu bentuk organisasi".

Menurut Jogiyanto dalam (Rudi Hermawan, Arief Hidayat, 2016) menjelaskan "Sistem informasi sebagai suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi, dan menyediakan pihak luar tertentu dengan laporan - laporan yang diperlukan".

Menurut Fauzi dalam (Anna et al., 2016) sistem informasi adalah :

"Suatu kombinasi dari orang-orang, fasilitas, teknologi, media, prosedur-prosedur dan pengendalian yang ditujukan untuk mendapatkan jalur komunikasi penting, memproses tipe transaksi rutin tertentu, memberi sinyal kepada manajemen dan yang lainnya terhadap kejadian-kejadian internal dan eksternal yang penting dan menyediakan sesuatu dasar untuk pengambilan keputusan".

Menurut Sutabri dalam (Kaleb, 2019) “Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan kepada pihak luar tertentu dengan laporan- laporan yang diperlukan”.

2.1.7. Komponen Sistem Informasi

Komponen sistem informasi, berikut ini adalah macam-macam komponen sistem informasi menurut (Farell, Saputra, & Novid, 2018) :

1. Komponen *input*/masukan

Input merupakan data yang masuk kedalam sistem informasi. Komponen ini merupakan bahan dasar dalam pengolahan informasi. Data untuk sistem informasi perlu ditangkap dan dicatat dalam dokumen dasar. Dokumen dasar merupakan formulir yang digunakan untuk menangkap (*capture*) dari data yang terjadi, yang selanjutnya data tersebut dimasukkan kedalam sistem informasi (*data entry*).

2. Komponen model

Informasi yang dihasilkan oleh sistem informasi berasal dari data yang diambil dari basis data yang diolah melalui model-model tertentu.

3. Komponen *output*/keluaran

Output adalah produk yang dihasilkan dari sistem informasi yang berguna bagi para pemakainya.

4. Komponen teknologi

Komponen teknologi merupakan komponen penting dalam sistem informasi. Tanpa ada teknologi yang mendukung, maka sistem informasi tidak akan dapat menghasilkan informasi yang tepat waktu.

5. Komponen basis data

Basis data (*database*) adalah kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan diperangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya.

2.1.8. Pengertian Gaji

Menurut Andrew F. Sikula (2015) “Gaji adalah balas jasa yang berbentuk uang yang diterima oleh karyawan sebagai konsekuensi dari statusnya sebagai karyawan yang memberikan kontribusi untuk mencapai tujuan perusahaan.”

Menurut Mulyadi (2017) “Gaji adalah pembayaran atas penyerahan jasa yang dilakukan oleh karyawan baik yang mempunyai jabatan maupun karyawan pelaksana”. Selanjutnya menurut Rokmulyati (2015) “Penggajian adalah suatu proses pemberian motivasi kepada karyawan yang dilakukan secara periodik”.

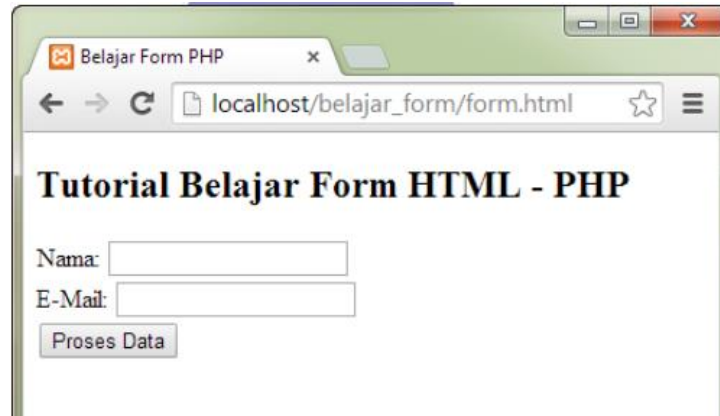
2.2. Peralatan Pendukung

2.2.1. Pengertian PHP

PHP (*Personal Home Page*) adalah teknologi yang diperkenalkan tahun 1994 oleh Rasmus Lerdorf. Menurut Purnomo dalam (Suharyanto, 2018), PHP merupakan :

“Bahasa skrip yang ditempatkan dalam *server* dan diproses di *server*. PHP sendiri mempunyai fungsi untuk menterjemahkan *script*/kode PHP yang kita buat. Dengan PHP ini dapat dibuat beragam aplikasi *web*, mulai dari halaman *web* yang sederhana sampai dengan aplikasi kompleks yang membutuhkan koneksi ke *database*. Sampai saat ini telah banyak *database* yang didukung oleh PHP dan kemungkinan akan terus bertambah. *Database* tersebut adalah Base, DBM, MySQL, SQL, ODBC, Oracle, Postgres,, Sybase, Velocois, HTML. “

Menurut Peranginangin dalam (Suharyanto, 2018) mengemukakan bahwa ”PHP adalah suatu program yang digunakan sebagai bahasa *Script server-side* dalam pengembangan *web* yang disisipkan pada dokumen HTML”.



Sumber: (Hidayat, 2017)

Gambar II.1

Hasil Penulisan PHP

Sistem kerja dari program ini adalah sebagai *interpreter* dan bukan sebagai *compiler*. Dikatakan sebagai bahasa *interpreter*, *script* mentahnya tidak harus diubah kedalam bentuk *source code*. Sehingga pada saat menjalankan program, kode dasar secara langsung akan dijalankan tanpa harus melalui pengubahan kedalam bentuk *source code*. Diantara kelebihan-kelebihan dari PHP :

1. Mudah dibuat dan kecepatan akses tinggi.
2. Dapat berjalan dengan *web server* yang berbeda dan dalam sistem operasi yang berbeda pula seperti *Unix*, *Windows NT* dan *Mancintosh*.
3. Diterbitkan secara gratis.
4. Dapat berjalan pada *web server* *Microsoft Personal Web Server* dan *Apache*.
5. Termasuk bahasa yang ditempel atau diletakan dalam *tag HTML (embedded)*.
6. Kemampuan untuk mengolah keluaran gambar (seperti *file* PDF, dan *Movie flash*).

2.2.2. Sublime Text

Sublime text salah satu kode editor yang biasa digunakan oleh para programmer untuk membuat suatu program.

Menurut Supono dan Putratama (2016:14) “*Sublime text* merupakan perangkat lunak *text editor* yang digunakan untuk membuat atau meng-*edit* suatu aplikasi. *Sublime text* mempunyai fitur *plugin* tambahan yang memudahkan programmer”.

Selain itu, menurut Faridi (2015:3) menjelaskan bahwa “*Sublime Text 3* adalah editor berbasis *python*, sebuah teks editor yang elegan, kaya akan fitur, *cross platform*, mudah dan simple yang cukup terkenal di kalangan *developer* (pengembang), penulis dan desainer”.

Jadi dapat disimpulkan bahwa *sublime text* ialah teks editor yang digunakan untuk membuat program aplikasi yang secara otomatis untuk mempermudah programmer dalam mengetikkan kode editor.

2.2.3. **Bootstrape**

Bootstrap sebuah *library framework CSS* yang dibuat khusus untuk bagian pengembang *front-end website*.

Menurut Alatas (2015:2) “*Bootstrap* merupakan *framework* ataupun *tools* untuk membuat aplikasi *web* ataupun situs *web responsive* secara tepat, mudah dan gratis”.

Sedangkan menurut Riyanto (2016:18) “*Bootstrap* adalah kumpulan alat gratis untuk membuat *website* dan aplikasi *web*”.

Dari penjelasan di atas dapat disimpulkan bahwa *bootstrap* ialah tampilan untuk membuat halaman *web* menjadi lebih bagus dan mempercepat seseorang dalam mendesain program.

2.2.4. **Pengertian MYSQL**

Menurut Raharjo (2015:16), “*MySQL* merupakan *software RDMS* atau *sever database* yang dapat mengelola *database* dengan sangat cepat, dapat menampung

data dalam jumlah sangat besar, dapat diakses oleh banyak *user* (*multi user*), dan dapat melakukan suatu proses secara sinkron atau berbarengan (*multi threaded*)”.

MySQL adalah sebuah *database server*, dapat juga berperan sebagai *client* sehingga sering disebut *database client/server*, yang *open source* dengan kemampuan dapat berjalan baik di OS (*operating system*) manapun, dengan *platform* maupun *linux*.

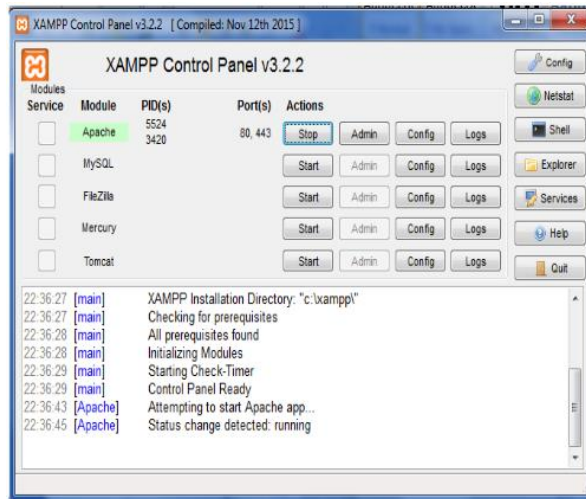
Sedangkan SQL adalah bahasa pemrogramannya, bahasa permintaan (*query*) dalam *database server* termasuk dalam *MySQL* itu sendiri. SQL juga dipakai dalam *software database server* lain, seperti *SQL Server*, *Oracle*, *PostgreSQL* dan lainnya.

2.2.5. Pengertian XAMPP

Menurut Madcoms (2016:148), “Xampp adalah sebuah paket kumpulan *software* yang terdiri dari *Apache*, *MySQL*, *PhpMyadmin*, *PHP*, *Perl*, *Filezilla*, dan lain-lain.

Menurut Riyanto (2015:3), “Xampp adalah sebuah *software* yang berfungsi untuk menjalankan *website* berbasis *PHP* dan menggunakan pengolah data *MySQL* yang dijalankan dikomputer secara lokal”.

XAMPP berperan sebagai *web server* pada komputer. XAMPP juga dapat disebut sebuah *CPanel server virtual*, yang dapat membantu anda melakukan *preview* sehingga dapat memodifikasi *website* tanpa harus *online* atau terakses dengan internet. *Software* XAMPP bersifat *open sources* yang dapat diperoleh secara gratis dari situs www.apachefriends.org. XAMPP adalah perangkat lunak yang mendukung banyak sistem operasi dan merupakan komplikasi dari beberapa program. Fungsinya adalah sebagai *server* yang berdiri sendiri dan terdiri atas *Apache*, *MySQL*, dan bahasa pemrograman *PHP*.



Sumber: (Labuhanbatu, 2016)

Gambar II.2

Xampp

2.2.6. Metode Pengembangan perangkat Lunak

Tahapan metode pengembangan perangkat lunak model *waterfall* menurut (Taufik et al., 2019):

1. Analisa Kebutuhan Perangkat Lunak

Proses pengumpulan kebutuhan dilakukan secara intensif dan spesifik terhadap kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh *user*. Spesifikasi kebutuhan perangkat lunak pada tahap ini perlu untuk didokumentasikan.

2. 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 pengodean. 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.

3. 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.

4. Pengujian

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

5. Pendukung (*support*) atau Pemeliharaan (*maintenance*)

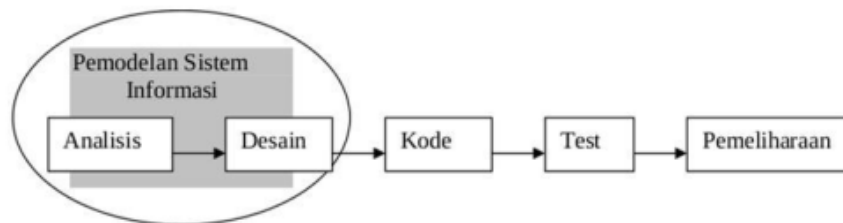
Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirimkan ke *user*. Perubahan bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan baru. Tahap pendukung atau pemeliharaan dapat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

Proses Pemeliharaan merupakan bagian paling akhir dari siklus pengembangan dan dilakukan setelah perangkat lunak dipergunakan.

Kegiatan yang dilakukan pada proses pemeliharaan antara lain :

- a. *Corrective Maintenance* : yaitu mengoreksi apabila terdapat kesalahan pada perangkat lunak, yang baru terdeteksi pada saat perangkat lunak dipergunakan.

- b. *Adaptive Maintenance* : yaitu dilakukannya penyesuaian/perubahan sesuai dengan lingkungan yang baru, misalnya *hardware*, *periperal*, sistem operasi baru, atau sebagai tuntutan atas perkembangan sistem komputer, misalnya penambahan *driver*.
- c. *Perfektive Maintenance* : Bila perangkat lunak sukses dipergunakan oleh pengguna. Pemeliharaan ditujukan untuk menambah kemampuannya seperti memberikan fungsi-fungsi tambahan, peningkatan kinerja dan sebagainya.



Sumber: (Taufik et al., 2019)

Gambar II.3
Metode Pengembangan Perangkat Lunak Model Waterfall

2.2.7. Metode Pengujian Sistem

1. Pengertian Pengujian Sistem

Menurut Al Fatta dalam (Hoendarto et al., 2017), pengujian sistem merupakan :

“Proses mengeksekusi sistem perangkat lunak untuk menentukan apakah sistem perangkat lunak tersebut cocok dengan spesifikasi sistem dan berjalan sesuai dengan lingkungan yang diinginkan. Pengujian sistem sering diasosiasikan dengan pencairan *bug*, ketidaksempurnaan program, kesalahan pada baris program yang menyebabkan kegagalan pada eksekusi sistem perangkat lunak.”

2. Metode Pengujian

Metode pengujian adalah cara atau teknik untuk menguji perangkat lunak. Metode pengujian berhubungan dengan perancangan data uji yang akan dieksekusi pada perangkat lunak yang dikembangkan. Metode pengujian diharapkan mempunyai mekanisme untuk menentukan data uji yang dapat menguji perangkat lunak secara lengkap (*completeness of test*) dan mempunyai kemungkinan tinggi untuk menemukan kesalahan (*high likelihood for uncovering error*) (Wibisono & Baskoro, 2017).

3. Metode *Black Box Testing*

Metode *Blackbox testing* adalah sebuah metode yang dipakai untuk menguji sebuah *software* tanpa harus memperhatikan detail *software*. Pengujian ini hanya memeriksa nilai keluaran berdasarkan nilai masukan masing-masing. Tidak ada upaya untuk mengetahui kode program apa yang *output* pakai. (Latif, 2015).

Metode *Blackbox testing* merupakan salah satu metode yang mudah digunakan karena hanya memerlukan batas bawah dan batas atas dari data yang diharapkan, Estimasi banyaknya *field* data entri yang akan diuji, aturan entri yang harus dipenuhi serta kasus batas atas dan batas bawah yang memenuhi, dan dengan metode ini dapat diketahui jika fungsionalitas masih dapat menerima masukan data yang tidak diharapkan maka menyebabkan data yang disimpan kurang (Cholifah et al., 2018).

Menurut Shihab dalam (Hoendarto et al., 2017), mengemukakan ciri-ciri *Black-Box Testing*, adalah sebagai berikut :

- a. *Blackbox testing* berfokus pada kebutuhan fungsional pada *software* berdasarkan pada spesifikasi kebutuhan dari *software*.
- b. *Blackbox testing* bukan teknik alternatif dari pada *white box testing*. Lebih dari pada itu, ia merupakan pendekatan pelengkap dalam mencakup *error* dengan kelas yang berbeda dari metode pengujian *white box testing*.
- c. *Blackbox testing* melakukan pengujian tanpa pengetahuan detail struktur internal dari sistem atau komponen yang dites juga disebut sebagai

behavioral testing, specification-based testing, input/output testing atau functional testing.

2.2.8. ERD (Enterprise Relationship Diagram)

Model Entity Relationship disebut secara umum dengan nama ERD (*Diagram Entity Relational*).

Menurut Fatta dalam (Romadhon & Desmulyati, 2019) “ERD (*Entity Relationship Diagram*) adalah gambar atau diagram yang menunjukkan informasi dibuat, disimpan, dan digunakan dalam sistem bisnis”.

Sedangkan menurut Rosa & Shalahuddin (2015:50) “ERD digunakan untuk pemodelan basis data relasional sehingga jika penyimpanan basis data menggunakan OODBMS maka perancangan basis data tidak perlu menggunakan ERD”. ERD memiliki beberapa aliran notasi seperti notasi Chen (dikembangkan oleh Peter Chen), Barker (dikembangkan oleh Richard Barker, Ian palmer, Harry Ellis), notasi Crow’s Foot, dan beberapa notasi lainnya. Namun yang banyak digunakan adalah notasi dari Chen.

ERD (*Entity Relationship Diagram*) adalah gambar atau diagram yang menunjukkan informasi dibuat, disimpan, dan digunakan dalam sistem bisnis. ERD ini digunakan untuk pemodelan basis data relasional.

Hubungan antara satu entitas dengan entitas yang lainnya dapat berupa :

1. Hubungan satu ke satu (*one to one relationship*)

Hubungan satu ke satu (*one to one*) terjadi apabila satu jenis entitas yang berhubungan dengan satu atribut pada entitas yang lain yang juga hanya memiliki satu hubungan dengan atribut pada entitas tersebut.



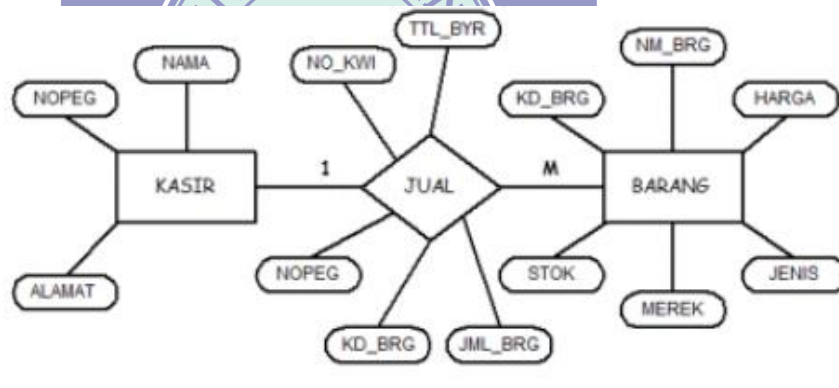
Sumber: (Suryadi, 2016)

Gambar II.4

Kardinalitas Satu ke Satu

2. Hubungan satu ke banyak (*one to many relationship*)

Hubungan satu ke banyak (*one to many*) terjadi apabila satu entitas memiliki satu atribut yang berhubungan dengan banyak atribut pada entitas yang lain, sedangkan atribut yang lain hanya memiliki hubungan satu atribut pada entitas tersebut.



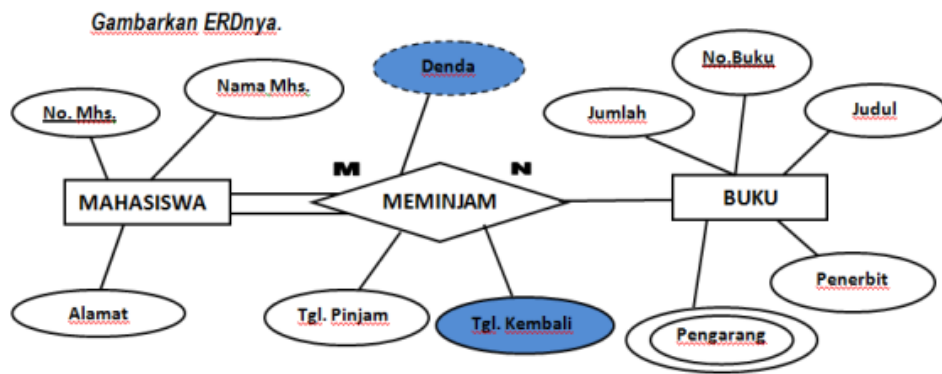
Sumber: (Suryadi, 2016)

Gambar II.5

Kardinalitas Satu ke Banyak

3. Hubungan banyak ke banyak (*many to many relationship*)

Hubungan banyak ke banyak (*many to many*) terjadi apabila satu entitas memiliki satu atribut yang berhubungan dengan banyak atribut pada entitas yang lain begitu juga sebaliknya dengan entitas yang lain memiliki satu atribut yang berhubungan banyak dengan atribut pada entitas tersebut.



Sumber: (Suryadi, 2016)

Gambar II.6

Kardinalitas Banyak ke Banyak

2.2.9. LRS (*Logical Record Structure*)

Menurut Frieyadie didalam (Firmansyah et al., 2020) menyatakan bahwa “LRS merupakan hasil dari ERD (*Entity Relationship Diagram*) berserta atributnya sehingga bisa terlihat hubungan-hubungan antara entitas”.

Sedangkan menurut Kusriani dalam (Septiani et al., 2019) “LRS (*Logical Record Structure*) adalah representasi dari struktur record-record pada tabel-tabel yang terbentuk dari hasil antar himpunan entitas”.

Menentukan kardinalitas, jumlah tabel dan *foreign key*. Model sistem yang digambarkan dengan sebuah diagram-ER akan mengikuti pola/ aturan pemodelan tertentu dalam kaitannya dengan konversi ke LRS, maka perubahan yang terjadi adalah mengikuti aturan -aturan berikut,

1. Setiap entitas akan diubah kebentuk kotak.
2. Sebuah atribut relasi disatukan dalam sebuah kotak bersama entitas jika hubungan yang terjadi pada diagram-ER 1:M (relasi bersatu dengan *cardinality* M) atau tingkat hubungan 1:1 (relasi bersatu dengan *cardinality* yang paling membutuhkan referensi).

3. Sebuah relasi dipisah dalam sebuah kotak tersendiri (menjadi entitas baru) jika tingkat hubungannya M:M (*many to many*) dan memiliki *foreign key* sebagai *primary key* yang diambil dari kedua entitas yang sebelumnya saling berhubungan.

2.2.10. UML (*Unified Modeling Language*)

Menurut Sukamto dan Shalahuddin dalam (Romadhon & Desmulyati, 2019) Mendefinisikan bahwa, “UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung”.

Menurut (Rohmayati & Purnama, 2017) “UML (*Unified Modeling Language*) adalah salah standar bahasa yang banyak digunakan didunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek”.

Menurut Shalahuddin dan Rosa dalam (Dalis, 2017) “UML (*Unified Modeling Language*) adalah salah satu standar bahasa yang banyak di gunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek”.

Sedangkan menurut Fowler dalam (Astuti, 2019) “UML (*Unifield Modelling Language*) adalah keluarga notasi grafis yang didukung oleh meta model tunggal, yang membantu pendeskripsian dan desain sistem perangkat lunak, khususnya sistem yang dibangun dengan menggunakan pemrograman berorientasi objek (OOP)”

Tujuan dari UML sebagai berikut:

1. Memodelkan suatu sistem (bukan hanya perangkat lunak) yang menggunakan konsep berorientasi objek.

2. Menciptakan suatu bahasa pemodelan yang dapat digunakan baik oleh manusia maupun mesin.
3. Memberikan bahasa yang bebas dari berbagai bahasa pemrograman.

UML terdiri dari banyak diagram antara lain :

1. *Use Case Diagram*

“*Use Case* mendeskripsikan sebuah interaksi antara satu atau lebih *actor* dengan sistem informasi yang akan dibuat serta menggambarkan fungsionalitas yang diterapkan dari sebuah sistem” (Shalahuddin & Rosa, 2018). *Use case* digunakan untuk mengetahui fungsi apa saja yang ada didalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu. Syarat penamaan pada *use case* adalah nama didefinisikan sederhana mungkin dan dapat dipahami. Ada dua hal utama pada *use case* yaitu pendefinisian apa yang disebut *actor* dan *use case*.

- a. *Actor* merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari *actor* adalah gambar orang, tapi *actor* belum tentu merupakan orang.
- b. *Use case* merupakan fungsionalitas yang disediakan sistem sebagai *unit-unit* yang saling bertukar pesan antar *unit* atau *actor*.

2. *Activity Diagram*

“Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja atau aktivitas dari sebuah sistem atau proses bisnis). Yang perlu diperhatikan disini adalah bahwa apa yang dilakukan *actor*, jadi aktivitas yang dapat dilakukan oleh sistem” (Shalahuddin & Rosa, 2018). Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal-hal berikut:

- a. Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan
- b. Urutan atau pengelompokan tampilan dari sistem dimana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan.
- c. Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya.

3. *Class Diagram*

Class diagram merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem.

Class diagram juga menunjukkan sifat-sifat dan operasi dari sebuah kelas dan kendala yang berlaku untuk cara objek yang terhubung. *Class* memiliki tiga area pokok yaitu :

- a. Nama, kelas harus mempunyai sebuah nama.
- b. Atribut, adalah kelengkapan yang melekat pada kelas. Nilai dari suatu kelas hanya bisa diproses sebatas atribut yang dimiliki.
- c. Operasi, adalah proses yang dapat dilakukan oleh sebuah kelas, baik pada kelas itu sendiri ataupun kepada kelas lainnya.

4. *Sequence Diagram*

Sequence diagram menggambarkan interaksi antar objek didalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* biasa digunakan untuk menggambarkan skenario atau langkah langkah yang dilakukan sebagai respon dari sebuah *event* untuk menghasilkan *output* tertentu.