

BAB II

LANDASAN TEORI

2.1 Konsep Dasar Sistem

2.1.1 Pengertian Sistem

Sistem merupakan kegiatan yang dibuat untuk diikuti dan dilaksanakan dalam sebuah perusahaan atau organisasi, sehingga kegiatan perusahaan atau organisasi tersebut dapat berjalan dengan benar.

Banyak para ahli mengemukakan pendapat yang berbeda-beda mengenai definisi. Sistem menurut Jeperson Hutahaean (2014:2) sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan kegiatan atau melakukan sasaran tertentu.

Sistem menurut Muhamad Muslihudin dan Oktafianto (2016:2) sistem adalah sekumpulan komponen atau jaringan kerja dari prosedur-prosedur yang saling berkaitan dan saling bekerja sama membentuk suatu jaringan kerja untuk mencapai sasaran atau tujuan tertentu.

Sistem menurut Elisabet Yunaeti Aggraeni dan Rita Irviani (2017:1) Sistem adalah kumpulan orang yang saling bekerja sama dengan ketentuan-ketentuan aturan yang sistematis dan terstruktur untuk membentuk satu kesatuan yang melaksanakan suatu fungsi untuk mencapai tujuan.

2.1.2 Karakteristik Sistem

Menurut Tata Sutabri (2012:20) sebuah sistem mempunyai beberapa masukan dan keluaran. Selain itu, sebuah sistem mempunyai karakteristik atau sifat-

sifat tertentu yang mencirikan bahwa hal tersebut bias dikatakan sebagai suatu sistem. Adapun karakteristik yang dimaksud adalah sebagai berikut:

a. Komponen Sistem

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, artinya saling bekerja sama membentuk satu kesatuan. Komponen-komponen sistem tersebut dapat berupa suatu subsistem. Setiap subsistem memiliki sifat dari sistem yang menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan. Suatu sistem dapat mempunyai sistem yang lebih besar atau sering disebut "supra sistem".

b. Batasan Sistem

Ruang lingkup sistem merupakan daerah yang membatasi antara sistem dengan sistem yang lain atau sistem lingkungan luarnya. Batasan sistem ini memungkinkan suatu sistem dipandang sebagai satu kesatuan yang tidak dapat dipisahkan.

c. Lingkungan Luar Sistem

Bentuk apapun yang ada diluar ruang lingkup atau batasan sistem yang mempengaruhi operasi sistem tersebut disebut lingkungan luar sistem. Lingkungan luar sistem ini dapat bersifat menguntungkan dan dapat juga bersifat merugikan sistem tersebut. Dengan demikian, lingkungan luar tersebut harus tetap dijaga dan dipelihara. Lingkungan luar merugikan harus dikendalikan. Kalau tidak, maka akan mengganggu kelangsungan hidup sistem tersebut.

d. Penghubung Sistem

Media yang menghubungkan sistem dengan subsistem lain disebut penghubung sistem atau interface. Penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem lain. Bentuk keluaran dari satu subsistem akan menjadi masukan untuk subsistem lain melalui penghubung tersebut. Dengan demikian, dapat terjadi suatu integrasi sistem yang membentuk satu kesatuan.

e. Masukan Sistem

Energy yang dimasukkan ke dalam sistem disebut masukan sistem, yang dapat berupa pemeliharaan (*maintenance input*) dan sinyal (*signal input*)> contoh, didalam suatu unit sistem computer, “ program” adalah *maintenance input* yang digunakan untuk mengoperasikan komputernya dan “data” adalah signal input yang digunakan untuk mengoperasikan komputernya dan “data” adalah signal input untuk diolah menjadi informasi.

f. Keluaran Sistem

Hasil energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna. Keluaran ini merupakan masukan bagi subsistem yang lain seperti sistem informasi. Keluaran yang dihasilkan adalah informasi. Informasi ini dapat digunakan sebagai masukan untuk pengambilan keputusan atau hal-hal lain yang menjadi input bagi subsistem lain

g. Pengolah Sistem

Suatu sistem dapat mempunyai suatu proses yang akan mengubah masukan menjadi keluaran, contohnya adalah sistem akuntansi. Sistem ini akan

mengolah data transaksi menjadi laporan-laporan yang dibutuhkan oleh pihak manajemen.

h. Sasaran Sistem

Suatu sistem memiliki tujuan dan sasaran yang pasti dan bersifat deterministic. Kalau suatu sistem tidak memiliki sasaran maka operasi sistem tidak ada gunanya. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuan yang telah direncanakan.

2.1.2 Klasifikasi Sistem

Menurut Jeperson Hutahaean (2014:6) sistem dapat diklasifikasikan dalam beberapa sudut pandang:

Klasifikasi sistem sebagai:

a. Sistem abstrak (*abstract system*)

Sistem abstrak adalah sistem yang berupa pemikiran-pemikiran atau ide-ide yang tidak tampak secara fisik

b. Sistem fisik (*physical system*)

Sistem fisik adalah sistem yang ada secara fisik

Sistem diklasifikasikan sebagai:

a. Sistem alamiah (*natural system*)

Sistem alamiah adalah sistem yang terjadi melalui proses alam, tidak dibuat oleh manusia. misalnya sistem perputaran bumi

b. Sistem buatan manusia (*human made system*)

Sistem buatan manusia adalah sistem yang dibuat oleh manusia yang melibatkan interaksi antara manusia dengan mesin (*human machine system*).

Sistem diklasifikasikan sebagai:

a. Sistem tertentu (*deterministic system*)

Sistem tertentu adalah sistem yang beroperasi dengan tingkah laku yang sudah data diprediksi, sebagai keluaran sistem yang dapat diramalkan

b. Sistem tak tentu (*probabilistic system*)

Sistem tak tentu adalah sistem yang kondisi masa depannya tak dapat diprediksi karena mengandung unsur probabilitas

Sistem diklasifikasikan sebagai:

a. Sistem tertutup (*close system*)

Sistem tertutup adalah sistem yang tidak terpengaruh dan tidak berhubungan dengan lingkungan luar, sistem bekerja otomatis tanpa ada turut campur lingkungan luar. Secara teoritis sistem tertutup ini ada, kenyataannya tidak ada sistem yang benar-benar tertutup, yang ada hanya *relatively closed system*.

b. Sistem terbuka (*open system*)

Sistem terbuka adalah sistem yang berhubungan dan terpengaruh dengan lingkungan luarnya. Sistem ini menerima input dan output dari lingkungan luarnya atau subsistem lainnya. Karena sistem terbuka terpengaruh lingkungan luar maka harus mempunyai pengendali yang baik

2.2 Pengertian Rancang bangun

Menurut Pressman yang dikutip oleh zulfiandri, dkk (2014) Kata “rancang” merupakan kata sifat dari “perancangan” yakni merupakan serangkaian prosedur untuk menerjemahkan hasil analisis dari sebuah sistem ke dalam bahasa pemrograman untuk mendeskripsikan dengan detail bagaimana komponen komponen sistem diimplementasikan.

Menurut Pressman yang dikutip oleh zulfiandri, dkk (2014) Kata “bangun” merupakan kata sifat dari “pembangunan” adalah kegiatan menciptakan sistem baru maupun mengganti atau memperbaiki sistem yang telah ada baik secara keseluruhan maupun sebagian.

Menurut Fajriyah, dkk (2017) rancang bangun adalah suatu istilah umum untuk membuat atau mendesain suatu objek dari awal pembuatan sampai akhir pembuatan.

Dengan demikian pengertian rancang bangun merupakan kegiatan menerjemahkan hasil analisa ke dalam bentuk paket perangkat lunak kemudian menciptakan sistem tersebut ataupun memperbaiki sistem yang sudah ada

2.3 File Sharing

Menurut Gede Wahyudi dan Trisna Hanggara, (2013:34). File sharing merupakan sebuah sistem di mana banyak pengguna dapat berbagi berkas dan sumber daya penyimpanan dengan konsep client-server yang dijalankan melalui jaringan. Sistem Client-Server adalah pembagian kerja antara server dan client yg mengakses server dalam suatu jaringan. Sebuah File Server memungkinkan

pengguna untuk berbagi informasi melalui jaringan tanpa harus secara fisik mengirim file dengan floppy disk atau beberapa perangkat penyimpanan eksternal lainnya. Adapun protokol maupun aplikasi yang sering digunakan untuk berbagi berkas file sharing adalah Network File System (NFS), dan Primary Domain Controller (PDC) Samba. NFS Network File System atau dalam Bahasa Indonesia: Sistem Berkas Jaringan adalah sebuah kumpulan protokol yang digunakan untuk mengakses beberapa sistem berkas melalui jaringan. Sedangkan Samba adalah program yang bersifat open source yang menyediakan layanan berbagi berkas file service dan berbagi alat pencetak (print service). Sebagai sebuah aplikasi file server.

2.4 Klasifikasi Sistem

Menurut Yakub (2012 : 4) pada buku Pengantar Sistem Informasi, Sistem dapat diklasifikasikan dari beberapa sudut pandang. Sistem yang digunakan dalam rancang bangun sistem inventaris pada bengkel Usaha Laris adalah sistem tertutup (closed system). Sistem tertutup (closed system) adalah sistem yang tidak berhubungan dan tidak terpengaruh dengan lingkungan luarnya. Sistem ini bekerja secara otomatis tanpa adanya turut campur tangan dari pihak diluarnya.

2.5 Pengertian Sistem Informasi

Menurut Krismiaji (2015:15), Sistem informasi adalah cara-cara yang diorganisasi untuk mengumpulkan, memasukkan, dan mengolah serta menyimpan data, dan cara-cara yang diorganisasi untuk menyimpan, mengelola, mengendalikan, dan melaporkan informasi sedemikian rupa sehingga sebuah organisasi dapat mencapai tujuan yang telah ditetapkan.

2.6 Android

Menurut Safaat (2011), Android adalah sebuah kumpulan perangkat lunak untuk perangkat mobile yang mencakup sistem operasi, middleware dan aplikasi utama mobile. Android memiliki 4 (empat) karakteristik sebagai berikut:

1. Terbuka

Android dibangun untuk benar-benar terbuka sehingga sebuah aplikasi dapat memanggil salah satu fungsi inti ponsel seperti membuat panggilan, mengirim pesan teks, menggunakan kamera, dan lain-lain. Android menggunakan sebuah mesin virtual yang dirancang khusus untuk mengoptimalkan sumber daya memori dan perangkat keras yang terdapat di dalam perangkat. Android merupakan open source, dapat secara bebas diperluas untuk memasukkan teknologi baru yang lebih maju pada saat teknologi tersebut muncul. Platform ini akan terus berkembang untuk membangun aplikasi mobile yang inovatif.

2. Semua Aplikasi dibuat sama

Android tidak memberikan perbedaan terhadap aplikasi utama dari telpon dan aplikasi pihak ketiga (third-party application). Semua aplikasi dapat dibangun untuk memiliki akses yang sama terhadap kemampuan sebuah telepon dalam menyediakan layanan dan aplikasi yang luas terhadap para pengguna.

3. Memecahkan hambatan pada aplikasi

Android memecah hambatan untuk membangun aplikasi yang baru dan inovatif. Misalnya, pengembang dapat menggabungkan informasi yang diperoleh dari web dengan data pada ponsel seseorang seperti kontak pengguna, kalender, atau lokasi geografis.

4. Pengembangan aplikasi yang cepat dan mudah

Android menyediakan akses yang sangat luas kepada pengguna untuk menggunakan library yang dipergunakan tools yang dapat digunakan untuk membangun aplikasi yang semakin baik. Android memiliki sekumpulan tools yang dapat digunakan sehingga membantu para pengembang dalam meningkatkan produktivitas pada saat membangun aplikasi yang dibuat.

Google Inc. Sepenuhnya membangun Android dan menjadikan bersifat terbuka (open source) sehingga para pengembang dapat menggunakan Android tanpa mengeluarkan biaya untuk lisensi dari Google dan dapat membangun Android tanpa adanya batasan-batasan. Android Software Development Kit (SDK) menyediakan alat dan Application Programming Interface (API) yang diperlukan untuk mulai mengembangkan aplikasi pada platform Android menggunakan bahasa pemrograman java.

2.7 Visual Studio Code

Visual Studio Code adalah Software yang sangat ringan, namun kuat editor kode sumbernya yang berjalan dari desktop. Muncul dengan built-in dukungan untuk JavaScript, naskah dan Node.js dan memiliki array beragam ekstensi yang tersedia untuk bahasa lain, termasuk C ++, C # , Python, dan PHP. Hal ini didasarkan sekitar Github ini Elektron, yang merupakan versi cross-platform dari Atom komponen kode-editing, berdasarkan JavaScript dan HTML5. Editor ini adalah fitur lengkap lingkungan pengembangan terpadu (IDE) dirancang untuk pengembang yang bekerja dengan teknologi cloud yang terbuka Microsoft. Visual Studio Code menggunakan open source NET perkakas untuk memberikan

Dukungan untuk ASP.NET C # kode, membangun alat pengembang Omnisharp NET dan compiler Roslyn. Antarmuka yang mudah untuk bekerja dengan, karena didasarkan pada gaya explorer umum, dengan panel di sebelah kiri, yang menunjukkan semua file dan folder Anda memiliki akses ke panel editor di sebelah kanan, yang menunjukkan isi dari file yang telah dibuka. Dalam hal ini, editor telah dikembangkan dengan baik, dan menyenangkan pada mata. Ia juga memiliki fungsi yang baik, dengan intellisense dan autocomplete bekerja dengan baik untuk JSON, CSS, HTML, {kurang}, dan Node.js. Visual Studio Code telah dirancang untuk bekerja dengan alat-alat yang ada, dan Microsoft menyediakan dokumentasi untuk membantu pengembang bersama, dengan bantuan untuk bekerja dengan ASP.NET 5, Node.js, dan Microsoft naskah, serta alat-alat yang dapat digunakan untuk membantu membangun dan mengelola aplikasi Node.js. Visual Studio Code benar-benar sedang ditargetkan pada pengembang JavaScript yang ingin alat pengembangannya lengkap untuk scripting server-side mereka dan yang mungkin ingin usaha dari Node.js untuk kerangka berbasis NET. Visual Studio Code, adalah belum solid, lintas platform kode Editor ringan, yang dapat digunakan oleh siapa saja untuk membangun aplikasi untuk Web.

2.8 Basis Data

Menurut Pratama (2014:17), menyatakan bahwa “Elemen basis data pada sistem informasi berfungsi sebagai media untuk menyimpan data dan informasi yang dimiliki oleh sistem informasi bersangkutan. Setiap aplikasi dan sistem yang memiliki data didalamnya (dengan disertai proses manipulasi data berupa insert, delete, edit/update), pasti memiliki sebuah basis data”.

Menurut Indrajani (2015:70), basis data adalah kumpulan data yang saling berhubungan secara logis dan didesain untuk mendapatkan data yang dibutuhkan oleh suatu organisasi.

Model basis data yang digunakan yaitu Entity Relationship Diagram (ERD) menurut Mata-Toledo dan Cushman (2007 : 139) merupakan representasi grafis dari logika database dengan menyertakan deskripsi detail mengenai seluruh entitas (entity), hubungan (relationship), dan batasan (constraint).

Jenis-Jenis atribut ERD :

- a. Identifier (Key) digunakan untuk menentukan suatu entity secara unik (primary Key).
- b. Descriptor (nonKey attribute) digunakan untuk menspesifikasikan karakteristik dari suatu entity yang tidak unik.

Metode/tahap untuk ERD :

- i. Menentukan Entitas
- ii. Menentukan Relasi
- iii. Menggambar ERD sementara
- iv. Mengisi Kardinalitas
- v. Menentukan Kunci Utama
- vi. Menggambar ERD berdasar Key
- vii. Menentukan Atribut
- viii. Atribut

2.9 SQL

Menurut Priyadi (2014:82) “SQL adalah suatu aplikasi komputer yang merujuk pada konsep *Relational Database Management Systems* (RDBMS), terdapat suatu struktur bahasa yang sudah standar untuk membangun basis datanya”.

Menurut Prasetio (2012:182) “SQL (dibaca “ess-que-el”) merupakan kependekkan dari *Structured Query Language*. SQL digunakan untuk berkomunikasi dengan sebuah *database*. Sesuai ANSI, SQL merupakan bahasa standar untuk sistem manajemen *database* relasional”.

Dapat disimpulkan bahwa SQL (*Structured Query Language*) merupakan struktur bahasa yang digunakan untuk membangun basis data dan digunakan untuk berkomunikasi dengan *database*.

2.10 XAMPP

Menurut Riyanto, (2011:45). Setelah menguji web server saatnya menguji database server yang ada dalam paket XAMPP yaitu MYSQL sebenarnya pada bahasan sebelumnya telah di jelaskan bagaimana mengaktifkan service MYSQL, yaitu melalui XAMPP control panel. Pada sub bab ini akan dibahas cara mengaktifkan server MYSQL secara manual, selanjutnya akan diuji melalui MYSQL. Seperti aplikasi basis data lainnya, MySQL juga memiliki database server yang harus dijalankan sebelum dapat melakukan hubungan ke basis data dan semua operasinya. Kelebihan-kelebihan MySQL adalah: a. Akses dari dan ke tabel database lebih cepat. b. Penggunaan ruang penyimpanan yang lebih efisien. c. Lebih

stabil jika dibandingkan dengan aplikasi database open source yang sekelas (seperti: PostgreSQL). d. Lebih mendukung fungsi standard ODBC.

2.11 Peralatan Pendukung (*Tools System*)

2.11.1 Unified Modelling Language (UML)

Menurut Miftakhul Huda (2010:137) Pemodelan (*Modelling*) adalah proses merancang perangkat lunak (*Software*) sebelum melakukan pengkodean (*Coding*). Dengan menggunakan model, diharapkan pengembangan perangkat lunak dapat memenuhi semua kebutuhan pengguna dengan lengkap dan tepat. *Unified Modeling Language* (UML) adalah sebuah “bahasa” yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem perangkat lunak.

Unified Modeling Language (UML) adalah sebuah bahasa yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. Denotasi yang lengkap untuk membuat visualisasi model suatu sistem. Sistem berisi informasi dan fungsi, tetapi yang secara normal digunakan untuk memodelkan sistem komputer (Sugiarti, 2013).

Tujuan dari *Unified Modeling Language* (UML) diantara lain 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.

Keuntungan menggunakan Unified Modeling Language (UML), adalah sebagai berikut:

1. Terdesain dan terdokumentasi secara professional sebelum dibuat.
2. Desain yang dibuat terlebih dahulu membuat reusable code dapat dikode dengan tingkat efisiensi yang tinggi.
3. UML menjanjikan akan menghasilkan hasil dengan biaya rendah, software lebih efisien, lebih dapat dipercaya, dan hubungan antar bagian yang terlibat menjadi lebih baik.

1. Use Case Diagram

Use case adalah deskripsi fungsi dari perspektif pengguna (Munawar, 2005:63). *Use case diagram* menggambarkan fungsionalitas yang diharapkan dari sebuah sistem yang ditekankan adalah “apa” yang diperbuat sistem bukan “bagaimana”. Sebuah *Use case* mempresentasikan sebuah interaksi antar aktor dengan sistem. Seorang aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem agar dapat melakukan sesuatu hal yang telah ditentukan. *Use case* dapat membantu mempresentasikan sebuah rancangan kepada klien.

2. Activity Diagram

Menurut Satzinger dalam Triandini dan Suardika (2012:37) menyatakan bahwa “activity diagram adalah sebuah diagram alur kerja yang menjelaskan berbagai kegiatan pengguna (atau sistem), orang yang melakukan masing-masing aktivitas, dan aliran sekuensial dari aktivitas-aktivitas tersebut”.

Menurut Nugroho (2010:62) mengemukakan bahwa “diagram aktivitas (activity diagram) sesungguhnya merupakan bentuk khusus dari state machine yang bertujuan memodelkan komputasi-komputasi dan aliran-aliran kerja yang terjadi dalam sistem/perangkat lunak yang sedang dikembangkan”. Dapat disimpulkan bahwa

3. Scenario

Menurut Kendall dan Kendall (2011:41-43) skenario adalah sebuah dokumentasi terhadap kebutuhan fungsional dari sebuah sistem. Form skenario merupakan penjelasan penulisan use case dari sudut pandang actor. skenario memiliki tiga area utama, yaitu:

1. Area *header*, area ini meliputi:
 - a. *Use Case Name* yang berisikan nama *use case*.
 - b. *UniqueID* yang berisikan nomor atau urutan *use case*.
 - c. Area yang berisikan area aplikasi atau sistem yang dimiliki oleh *use case*.
 - d. *Actor(s)* yang berisikan nama *actor* yang terdapat pada *use case*.
 - e. *Description* yang berisikan deskripsi singkat tentang *use case*.
 - f. *Triggering Event* yang berisikan kejadian yang menyebabkan atau memicu *use case* ini terjadi.
 - g. *Trigger Type* yang berisikan jenis pemicu. *Trigger type* terbagi menjadi dua jenis, yaitu *external* dan *temporal*. Pemicu *external* adalah pemicu peristiwa yang dimulai dari *actor*, baik seseorang atau sistem lain yang meminta informasi, seperti sistem reservasi penerbangan yang meminta

informasi penerbangan dari sistem penerbangan. Sedangkan pemicu *temporal* adalah pemicu peristiwa yang dimulai oleh waktu, seperti menghasilkan statistik pemerintah pada tanggal yang ditentukan.

2. *Area step performed*, area ini meliputi:
 - a. *Step performed* yang berisikan langkah-langkah atau tahapan yang dilakukan pada proses ini.
 - b. *Information for step* yang berisikan informasi yang diperlukan untuk masing-masing langkah atau tahapan.
3. *Area footer*, area ini meliputi:
 - a. *Pre-condition* yang berisikan kondisi sistem sebelum *use case* dilakukan.
 - b. *Post-condition* yang berisikan kondisi sistem setelah *use case* selesai.
 - c. *Assumptions* yang berisikan premis dasar yang diasumsikan oleh analis dipenuhi oleh aktor sebelumnya.
 - d. *Requirement Met* yang berisikan persyaratan yang harus dipenuhi sebelum penerapan *use case*.
 - e. *Outstanding Issues* yang berisikan masalah atau pertanyaan harus dijawab sebelum penerapan *use case*.
 - f. *Priority* yang berisikan indikasi *use case* mana yang harus dikembangkan terlebih dahulu dan yang mungkin tertunda.

Risk yang berisikan penilaian kasar apakah ada masalah atau kesulitan dalam mengembangkan *use case*.