**BAB II**

**LANDASAN TEORI**

**2.1. Tinjauan Pustaka**

**2.1.1. Konsep Dasar Sistem**

1. **Pengertian Sistem**

 Sistem menurut McLeod (2005:3) adalah “sekelompok elemen yang terintegrasi dengan maksud yang sama untuk mencapai suatu tujuan”

 Sistem menurut Lucas (1989) (2005:3) adalah “suatu komponen atau variabel yang terorganisir, saling berinteraksi, saling bergantung, satu sama lain dan terpadu”.

 Berdasarkan uraian di atas, dapat diambil kesimpulan bahwa sistem pada dasarnya merupakan sekelompok unsur yang saling berhubungan satu dengan yang lainnya dan berfungsi secara bersama-sama untuk mencapai tujuan tertentu. Menurut Jogiyanto (2005:4) suatu sistem mempunyai karakteristik atau sifat-sifat yang tertentu diantaranya :

1. Komponen Sistem

Suatu sistem terdiri dari jumlah komponen yang saling berinteraksi, yang artinya saling bekerja sama membentuk satu kesatuan. Komponen-komponen sistem atau elemen-elemen sistem dapat berupa suatu subsistem atau bagian-bagian dari sistem.

1. Batas Sistem (*Boundary*)

Merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya.

1. Lingkungan Luar Sistem (*Environment*)

Adalah apapun diluar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar ini dapat bersifat menguntungkan atau merugikan sistem tersebut.

1. Penghubungan Sistem (*Interface*)

Merupakan media penghubung antara satu subsistem dengan subsistem yang lainnya.

1. Masukkan Sistem (*Input*)

Adalah energi yang dimasukkan ke dalam sistem. Masukkan dapat berupa masukkan perawatan (*maintenance input*) dan masukkan sinyal (*signal input*). *Maintenance input* adalah energi yang dimasukkan supaya sistem dapat beroperasi. *Signal input* adalah energi yang diproses untuk didapatkan keluaran.

1. Keluaran Sistem (*Output*)

Adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan. Dapat berupa masukkan untuk subsistem lainnya.

1. Pengolah Sistem

Adalah merubah masukkan menjadi keluaran

1. Tujuan (*Goal*) atau Sasaran (*Objective*)

Suatu sistem dapat dikatakan berhasil bila mengenai sasaran atau tujuannya.

Suatu sistem dapat diklasifikasikan dari beberapa sudut pandang menurut Jogiyanto (2005:6) diantaranya adalah :

1. Sistem Abstrak (*Abstract System*)

Adalah sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik.

1. Sistem Alamiah (*Natural System*)

Adalah sistem yang terjadi melalui proses alam, tidak dibuat manusia.

1. Sistem Manusia (*Human Made System*)

Adalah sistem yang dirancang dan dibuat oleh manusia.

1. Sistem Tertentu (*Determinetic System*)

Adalah sistem yang beroperasi dengan tingkah laku yang sudah dapat diprediksi.

1. Sistem Tak Tentu (*Probabilistik System*)

Sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probabilistik.

1. Sistem Terbuka (*Open System*)

Adalah sistem yang berhubungan dan terpengaruh dengan lingkungan luarnya.

1. Sistem Tertutup (*Closed System*)

Adalah sistem yang berhubungan dan tidak terpengaruh oleh lingkungan luarnya.

**2. Rancang**

Menurut Rosyid (2012) rancang adalah setelah tahap analisis sistem selesai dilakukan, maka analis sistem telah mendapatkan gambaran dengan jelas apa yang harus dikerjakan. Tiba waktunya sekarang bagi analis sistem untuk memikirkan bagaimana bentuk sistem tersebut.

**2.1.2. Konsep Dasar Program**

1. **Program**

 Program menurut Jogiyanto (2005:582) “merupakan kegiatan menulis kode program yang akan dieksekusi oleh komputer. Kode program yang ditulis oleh pemprogram (*programmer*) harus berdasarkan dokumentasi yang disediakan oleh analis sistem hasil dari disain sistem secara rinci. Hasil program yang sesuai dengan disainnya akan menghasilkan program yang sesuai dengan yang dibutuhkan oleh pemakai sistem”.

1. **Web**

Suatu *web site* (*world wide web*) tidak lepas dari perkembangan *internet* yang berawal dari cara berkomunikasi para ilmuwan dan penelitian di Amerika, namun saat sekarang ini telah berkembang menjadi media komunikasi global yang dipakai semua orang di dunia. Pertumbuhan ini membawa penting mendasar, diantaranya kenyataan *internet* tidak diciptakan pada jaman *Graphical User Interface* (GUI) seperti saat sekarang ini. Internet dimulai pada masa dimana orang masih menggunakan alat-alat akses yang tidak *user-friendly* yaitu terminal berbasis *text* serta perintah-perintah (*command line*) yang panjang-panjang serta sulit diingat. Sangat berbeda dengan komputer dewasa ini yang mempunyai tampilan yang menarik disertai konsep multimedia dari mulai teks, gambar, animasi, suara, dan film, serta konsep hiperlink perhalaman. Bahkan terjadi manipulasi data antar komputer, penyimpanan data serta informasi yang sangat akurat, tercepat dan terbaru dari seluruh dunia.

Penyajian suatu *web* harus terprogram sedinamis mungkin, dengan pemanfaatan aplikasi *client-server* dalam hal ini *programmer* akan mengatasi kesulitan dalam proses pemeliharaannya karena pengunjung suatu *web* akan berinteraksi langsung dengan database jika nanti *web* tersebut melakukan pengolahan *query* di dalamnya.

a**.** Pengenalan Internet.

Saat ini semakin banyak orang menggunakan *internet* bahkan banyak orang yang sudah dapat membuat *website* sendiri, namun mereka buat itu adalah *website* statis. Pada dasarnya *web* itu merupakan kependekan dari *website* yaitu suatu layanan sajian informasi yang menggunakan konsep *hyperlink* atau menghubungkan satu halaman ke halaman lainnya. Informasi yang disajikan web menggunakan konsep multimedia dari multi *teks*, gambar, animasi, suara (*audio*), dan film (*video*). Sedangkan *internet* merupakan sumber informasi dan sumber daya komputer yang menjangkau seluruh dunia. *Internet* berasal dari kata *interconnection networking*, *inter* kependekan dari kata *international*, *connection* berarti hubungan, sedangkan *networking* adalah jaringan, sehingga internet dapat didefinisikan sebagai jaringan yang terdiri dari berbagai macam dan jenis komputer di seluruh dunia yang saling berkomunikasi dengan berbasiskan suatu *protocol* yang dinamakan TCP/IP (*Transmission Control Protocol/Internet Protocol*).

Istilah-istilah yang sering digunakan dalam menggunakan *internet* diantaranya adalah :

* + - 1. *World Wide Web* atau yang biasa disebut www

Adalah sistem pengiriman dokumen yang dapat menyebar keseluruh dunia. Pemakai internet yang menggunakan fasilitas ini dapat dengan mudah mencari informasi yang disajikan dalam halaman *web*. Dengan adanya fasilitas ini menjadikan www sebagai salah satu layanan *internet* yang kian berkembang dan paling banyak digunakan.

2. *Web Browser*

Untuk dapat mengakses dokumen *web,* diperlukan suatu program aplikasi yang dikenal dengan *web browser*. Terdapat beberapa *web browser* yang dapat diperoleh secara gratis (*freeware*) ataupun membayar (*shareware*) di internet, diantaranya adalah *internet explorer, netscape navigator, mozilla,* opera dan lain-lain.

1. ***Hypertext Preprocessor (PHP)***

 PHP merupakan bahasa pemrograman berbentuk script yang ditempatkan dalam server dan diproses di server. Hasil dari pengolahan akan dikirimkan ke klien, tempat pemakai menggunakan browser. Secara khusus, PHP dirancang untuk membentuk web dinamis. Artinya, ia dapat membentuk suatu tampilan berdasarkan permintaan terkini. Misalnya, kita bisa menampilkan isi database ke halaman web. Pada prinsipnya, PHP mempunyai fungsi yang sama dengan script seperti ASP (Actives Server Page), Cold Fusion, ataupun Perl (Kadir, 2003).

1. ***My Structure Query Language* (MySQL)**

Salah satu aplikasi basis data adalah MySQL yang merupakan sebuah program pembuat basis data yang bersifat *open source*, artinya siapa saja boleh menggunakannya dan tidak dicekal. (Nugroho, 2004). MySQL merupakan sistem manajemen basis data yang fungsinya untuk menambah, mengakses, dan memproses data yang disimpan dalam sebuah basis data komputer. Kelebihan yang dimiliki MySQL yaitu:

a. Ditulis dalam bahasa C dan C++.

b. Bekerja dalam berbagai platform (misalnya Windows, Mac OS X, Solaris, Unix, dan lain-lain).

c. Menyediakan mesin penyimpanan (*engine storage*) transaksi dan non transaksi.

d. *Server* tersedia sebagai program yang terpisah untuk digunakan pada lingkungan jaringan klien/*server*.

e. Mempunyai *library* yang dapat ditempelkan pada aplikasi yang berdiri sendiri (*standalone application*) sehingga sistem tersebut dapat digunakan pada komputer yang tidak mempunyai jaringan.

f. Mempunyai sistem *password* yang fleksibel dan aman.

g. Klien dapat terkoneksi ke MySQL *server* menggunakan soket TCP/IP pada *platform* mana pun.

h. *Server* dapat mengirim pesan kesalahan ke klien dalam berbagai bahasa.

MySQL termasuk jenis RDBMS (*Relational Database Management System*). Pada MySQL sebuah basis data terdiri atas tabel-tabel dan sebuah table terdiri atas baris dan kolom.

1. ***Database***

 Menurut Nugroho (2005 : 400) *Database* merupakan media yang digunakan untuk menyimpan data. Dalam membuat *website* perlu menyimpan artikel, berita, format halaman, nama *user*, *password*, dan lain-lain. Semua ini membutuhkan tempat penyimpanan yang biasanya dilakukan ke dalam database. Beberapa database yang sering digunakan diantaranya:

1. *Oracle*
2. *Microsoft SQL Server*
3. *Microsoft Access*
4. *DB2*
5. *My SQL*
6. *PostgreeSQL*

 Pada perancangan ini penulis akan mempergunakan *Microsoft SQL Server* 2000 sebagai database nya karena *Microsoft SQL Server 2*000 mempunyai banyak kelebihan, adapun kelebihan dari *Microsoft SQL Server*  2000antara lain:

1. Dari segi performa, *Microsoft SQL Server 2000* tidak bisa diragukan, pemrosesan databasenya sangat cepat.
2. Mudah untuk dipelajari.
3. Kompabilitas dengan berbagai sistem operasi dan *web server* yang ada.
	* 1. **Pengenalan UML (*Unified Modeling Language*)**

UML (*Unified Modelling Language*) merupakan alat komunikasi yang konsisten dalam mensupport para pengembang sistem (Herlawati, Widodo, 2011). Para pengembang sistem berorientasi objek menggunakan bahasa model untuk menggambarkan, membangun dan mendokumentasikan sistem yang sedang dirancang. UML (*Unified Modelling Language*) memungkinkan para anggota *team* untuk bekerja sama dengan model yang sama dalam merancang suatu sistem yang akan dibuat, dikarenakan dapat mempermudah bagi perancangan dalam merancangnya.

 UML (*Unified Modelling Language*) merupakan suatu alat yang telah menjadi standart dalam merancang suatu sistem visualisasi dan mendokumentasikan sistem piranti lunak.

 UML (*Unified Modelling Language*) menyediakan Sembilan jenis diagram yang dikelompokan berdasarkan sifatnya yaitu statis atau dinamis, jenis diagram itu antara lain :

**1. Diagram Kelas (*Class Diagram*)**

 Diagram kelas **(***Class Diagram***)** bersifat statis, diagram ini memperlihatkan himpunan kelas-kelas, antarmuka-antarmuka, kolaborasi-kolaborasi, serta relasi-relasi (Herlawati, Widodo, 2011). Diagram ini umum dijumpai pada pemodelan sistem berorientasi objek

**2. Diagram Paket (*Package Diagram*)**

 Diagram paket bersifat statis, diagram ini memperlihatkan kumpulan kelas-kelas, merupakan bagian dari diagram komponen (Herlawati, Widodo, 2011). Diagram paket juga dapat memvisualisasikan ketergantungan antar bagian dalam suatu sistem dan bermanfaat dalam mencari letak permasalahan.

 Suatu diagram paket adalah konstruksi utama UML (*Unified Modelling Language*) terdiri dari kumpulan model-model yang lain. Pengambaran diagram paket dapat dilakukan dengan menuliskan elemen paket di dalam suatu paket atau bila ingin lebih rinci dengan menghubungkan paket dengan elemenya lewat garis berujung lingkaran.

**3. Diagram *Use-Case***

 Diagram *use-case* bersifat statis. Diagram ini memperlihatkan himpunan *use-case* dan aktor-aktor (Herlawati, Widodo, 2011). Diagram ini terutama sangat penting untuk mengorganisasi dan memodelkan perilaku suatu sistem yang dibutuhkan serta diharapkan pengguna. Komponen diagram *use case* adalah :

 a. Aktor (*actor*), menggambarkan pihak-pihak yang berperan dalam sistem.

 b. *Use Case,* aktivitas atau saran yang disiakan oleh bisnis atau sistem.

 c. Hubungan (*Link*), aktor mana saja yang terlibat *use case.*

**4. Diagram Interaksi dan *Sequence* ( urutan )**

 Diagram interaksi dan *sequence* ( urutan ) bersifat dinamis, diagram urutan adalah diagram interaksi yang menekankan pada pengiriman pesan dalam suatu waktu tertentu (Herlawati, Widodo, 2011).

**5. Diagram Komunikasi (*communication diagram*)**

 Diagram Komunikasi (*communication diagram*) bersifat dinamis, diagram sebagai pengganti diagram kolaburasi yang menekankan organisasi struktural dari objek-objek yang menerima serta mengirim pesan (Herlawati, Widodo, 2011).

**6. Diagram Statechart ( *Statechart Diagram* )**

 Diagram Statechart ( *Statechart Diagram* ) bersifat dinamis, diagram status memperlihatkan keadaan-keadaan pada sistem, memuat status (*state*), transisi, kejadian serta aktifitas pesan (Herlawati, Widodo, 2011). Diagram ini terutama penting untuk memperlihatkan sifat dinamis antarmuka (*interface*), kelas, kolaburasi, dan terutama penting pada pemodelan sistem-sistem yang reaktif.

**7. Diagram Aktifitas (A*ctivity Diagram*)**

 diagram aktifitas (*activity diagram*) bersifat dinamis, diagram aktifitas adalah tipe khusus dari diagram status yang memperlihatkan aliran dari suatu aktifitas ke aktifitas lainya dalam suatu sistem (Herlawati, Widodo, 2011). Diagram ini penting terutama dalam permodelan fungsi-fungsi suatu sistem dan member tekanan pada aliran kendali suatu objek.

8. **Diagram Komponen (C*omponent Diagram*)**

 Diagram Komponen (C*omponent Diagram*) bersifat statis, diagram komponen ini memperlihatkan organisasi serta ketergantungan sitem atau perangkat lunak pada komponen-komponen yang telah ada sebelumnya. (Herlawati, Widodo, 2011). Diagram ini berhubungan dengan diagram kelas dimana komponen secara tipikal dipetakan kedalam satu atau lebih kelas-kelas, antarmuka-antarmuka serta kolaborasi-kolaborasi.

 Hubungan antara Komponen (*component*) dengan kelas adalah bahwa komponen merupakan subsistem berisi kelas-kelas yang bekerja sama merealisasikan antarmuka yang diinstruksikan.

**9. Diagram Deployment (*Deployment Diagram*)**

 Diagram deployment (d*eployment diagram*) bersifat statis, diagram ini memperlihatkan konfigurasi saat aplikasi di jalankan (*run-time*), memuat simpul-simpul beserrta komponen-komponen yang ada didalamnya (Herlawati, Widodo, 2011). Diagram *deployment* berhubungan erat dengan diagram komponen dimana diagram ini memuat satu atau lebih komponen-komponen. Diagram ini sangat berguna saat aplikasi kita berlaku sebagai aplikasi yang dijalankan pada banyak mesin (*distributed computing*).

**2.1.4 *Entity Relationship Diagram (ERD)***

1. **Diagram Hubungan Antara Entitas atau *Entity Relationship Diagram* (ERD).**

ERD menurut Al-Bahra (2004a:123) adalah “adalah notasi grafik dari sebuah model data atau sebuah model jaringan yang menjelaskan tentang data yang tersimpan (*storage* data) dalam sistem secara abstrak.” Diagram ini tidak menyatakan bagaimana memanfaatkan data, membuat data, mengubah data dan menghapus data. Elemen-elemen Diagram Hubungan Entitas antara lain :

a. *Entity*

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 dikelompakkan dalam empat jenis nama, yaitu orang, benda, lokasi, kejadian (terdapat *unsure* waktu didalamnya).

b. *Relationship*

Digambarkan dengan sebuah bentuk belah ketupat, *Relationship* adalah hubungan alamiah yang terjadi anatara entitas.

c. *Relationship Degree*

Atau Derajat *Relationship* adalah jumlah entitas yang berpartisipasi dalam satu *Relationship.*

1. *Unary Relationship*

Adalah model *Relationship* yang terjadi diantara *entity* yang berasal dari *entity set* yang sama. Disebut juga sebagai *Recursive Relationship* atau *Reflective Relationship*.

2. *Binary Relationship*

Adalah model *Relationship* antara *instance-instance* dari suatu tipe entitas (dua entitas yang berasal dari *entity* yang sama).

3. *Ternary Relationship*

Merupakan *relationship* antara *instance-instance* dari tiga tipe entitas secara sepihak.

d. *Atribut*

 Adalah sifat atau karakteristik dari tiap entitas maupun tiap relationship. Maksudnya adalah sesuatu yang menjelaskan apa sebenarnya yang dimaksud entitas maupaun relationship, sehingga sering dikatakan atribut adalah elemen dari setiap entitas dan relationship.

e. Kardinalitas (*Cardinality*)

Menunjukkan jumlah maksimum *tuple* yang dapat berelasi dengan *entitas* pada *entitas* yang lain.

Terdapat 3 (tiga) macam kardinalitas relasi, yaitu :

* + - 1. *One to One*

Tingkat hubungan satu ke satu, dinyatakan dengan satu kejadian pada *entitas* pertama, hanya mempunyai satu hubungan dengan satu kejadian pada *entitas* yang kedua dan sebaliknya.

* + - 1. *One to Many* atau *Many to One*

Tingkat hubungan satu kebanyak adalah sama banyak ke satu. Untuk satu kejadian pada *entitas* yang pertama dapat mempunyai banyak hubungan dengan kejadian pada entitas yang kedua dan sebaliknya.

* + - 1. *Many* to *Many*

Tingkat hubungan banyak ke banyak terjadi jika tiap kejadian pada sebuah entitas akan mempunyai banyak hubungan dengan kejadian pada *entitas* lainya. Baik dilihat dari sisi *entitas* yang pertama, maupun dilihat dari sisi *entitas* yang kedua.

**2.2 Penelitian Terkait**

 Qoyyimah, dkk (2012) telah melakukan penelitian yang berjudul *Rancang Bangun Helpdesk Ticketing System (Studi Kasus : PT. Primus Indojaya)*. Helpdesk ticketing system yang terkomputerisasi sangat mendesak diperlukan oleh PT. Primus Indojaya, dikarenakan dalam menjalankan proses usaha perusahaan seringkali ditemukan masalah-masalah pada peralatan penunjang yang ada dilapangan dan harus segera diselesaikan. Adapun masalah yang berkaitan adalah tentang hardware dan software. Namun penanganan masalah yang saat ini terjadi pada helpdesk ticketing system di PT. Primus Indojaya masih kurang baik, karena proses pencatatan helpdesk yang masih manual dan sederhana sehingga mempengaruhi kinerja perusahaan tidak cepat teratasi, dan prosesnya tidak terkontrol. Semua pencatatan laporan bulanan serta perhitungan masalah masih dilakukan manual. Hal ini dapat memberikan dampak negatif, yakni: terdapat masalah yang dicatat secara berulang, perhitungan masalah tidak akurat karena dilakukan manual dan perlunya manajer operasional menunggu untuk mendapatkan laporan yang diinginkan. Dan juga belum adanya sistem yang mampu mengukur tingkat masalah pada helpdesk dalam menangani permasalahan yang terjadi pada PT. Primus Indojaya. Oleh karena itu dibutuhkan sistem helpdesk ticketing system yang terintegrasi dengan baik dan cepat sehingga pengaksesan data pada helpdesk dapat dilakukan dengan mudah dan cepat guna pengukuran tingkat masalah pengaksesan laporan oleh manajer operasional, serta permasalahan dapat tertangani dengan baik dalam cakupan batasan masalah yang menghasilkan solusi tepat untuk mengatur sumberdaya yang ada.

 Selanjutnya penelitian yang dilakukan oleh. Ali, T (2008) dengan judul *Tinjauan umum tentang Helpdesk dan Framework terkait.* Langkah-langkah yang perlu dilakukan oleh instansi dalam implementasi tim helpdesknya antara lain, (1) membentuk tim khusus untuk menganalisis organisasinya terkait kebutuhan implementasi helpdesk tersebut, (2) membentuk organisasi helpdesk sesuai kebutuhan, (3) memilih workflow helpdesk yang disesuaikan dengan kondisi instansi, (4) memilih framework yang tepat untuk mendasari jalannya implementasi helpdesk di instansi terkait, dan (5) menuangkan aturan-aturan dasar (standard operation procedur-SOP) untuk menguatkan kiprah tim helpdesk di instansi terkait.Terbentuknya tim helpdesk diharapkan dapat mengatasi permasalahan dan memberikan layanan terhadap seluruh permasalahan bidang TIK baik yang bersifat internal (pengembang aplikasi) maupun seluruh permasalahan operasional aplikasi ataupun hal-hal yang berhubungan dengan sistem dan teknologi informasi dan komunikasi.