

BAB II

LANDASAN TEORI

2.1. Tinjauan Pustaka

2.1.1. Pengertian Sistem

Menurut Sutabri (2012:6) “Sistem pada dasarnya adalah sekelompok unsur yang erat hubungannya satu dengan yang lain, yang berfungsi bersama-sama untuk mencapai tujuan tertentu.”

Sistem mempunyai beberapa karakteristik atau sifat-sifat tertentu, antara lain:

1. Komponen Sistem (*Component*)

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

2. Batasan Sistem (*Boundary*)

Ruang lingkup sistem merupakan daerah yang membatasi antara sistem dengan sistem lainnya atau sistem dengan lingkungan luarnya. Batasan sistem ini memungkinkan suatu sistem dipandang sebagai satu kesatuan yang tidak dapat dipisah-pisahkan.

3. Lingkungan Luar Sistem (*Environment*)

Bentuk apapun yang ada di luar ruang lingkup atau batasan sistem yang mempengaruhi operasi sistem tersebut disebut dengan lingkungan luar sistem. Lingkungan luar sistem ini dapat menguntungkan dan dapat juga merugikan sistem tersebut. Lingkungan luar yang menguntungkan merupakan energi bagi sistem tersebut, yang dengan demikian lingkungan luar tersebut harus selalu dijaga dan dipelihara. Sedangkan lingkungan luar yang merugikan harus dikendalikan. Kalau tidak maka akan mengganggu kelangsungan hidup sistem tersebut.

4. Penghubung Sistem (*Interface*)

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

5. Masukan Sistem (*Input*)

Energi yang dimasukkan kedalam sistem disebut masukan sistem, yang dapat berupa pemeliharaan (*maintenance input*) dan sinyal (*signal input*). Sebagai contoh, di dalam suatu unit sistem komputer, “program” adalah *maintenance input* yang digunakan untuk mengoperasikan komputer. Sementara “data” adalah *signal input* yang akan diolah menjadi informasi.

6. Keluaran Sistem (*Output*)

Hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna. Keluaran ini merupakan masukan bagi subsistem yang lain. Seperti contoh sistem informasi, keluaran yang dihasilkan adalah informasi, dimana informasi ini dapat digunakan sebagai masukan untuk pengambilan keputusan atau hal-hal lain yang merupakan input bagi subsistem lainnya.

7. Pengolah Sistem (*Process*)

Suatu sistem dapat mempunyai suatu proses yang akan mengubah masukan menjadi keluaran. Sebagai contoh, sistem akuntansi. Sistem ini akan mengolah data transaksi menjadi laporan-laporan yang dibutuhkan oleh pihak manajemen.

8. Sasaran Sistem (*Objective*)

Suatu sistem memiliki tujuan dan sasaran yang pasti dan bersifat deterministik. 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. Model Pengembangan Sistem

Model pengembangan sistem informasi yang penulis gunakan dalam sistem informasi Taman Kanak-Kanak Kepompong Alam Jakarta, yaitu model *waterfall*. Menurut Sukanto dan Shalahuddin (2013:28) “Model SDLC air terjun (*waterfall*) sering juga disebut model sekuensial linier (*sequential linear*) atau alur hidup

klasik (*classic life cycle*)". Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengodean, pengujian, dan tahap pendukung (*support*).

1. Analisis Kebutuhan Perangkat Lunak

Proses pengumpulan kebutuhan dilakukan secara intensif untuk memspezifikasikan 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 dari segi logik 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 analisis spesifikasi untuk perubahan perangkat lunak yang sudah ada, tapi tidak untuk membuat perangkat lunak baru.

2.1.3. Web Sekolah

Menurut Hariyanto (2015:2) “Salah satu tuntutan era globalisasi dalam dunia pendidikan adalah adanya sebuah pengelolaan sekolah yang terintegrasi dengan sebuah sistem informasi dan teknologi berbasis komputer”. Oleh karena itu, sudah semestinya perkembangan teknologi informasi dan komputer yang kebanyakan orang mengkaitkan dengan teknologi internet itu direspon oleh sekolah dalam rangka meningkatkan pelayanan dan kualitas pendidikan kepada siswa.

Website merupakan salah satu media penyampaian informasi dan publikasi yang mudah diakses dari mana saja, kapan saja tanpa terbatas oleh wilayah geografis yang dapat dimanfaatkan oleh sekolah.

Website dapat diartikan sebagai kumpulan halaman yang menampilkan informasi data teks, data gambar, data animasi, suara, video, dan atau gabungan

dari semuanya, baik yang bersifat statis maupun dinamis yang membentuk suatu rangkaian bangunan yang saling terkait, dimana masing-masing dihubungkan dengan jaringan-jaringan halaman.

Website sekolah sekarang ini sedang menjadi tren di dunia pendidikan dan *website* sekolah banyak mendapatkan perhatian, tanggapan yang sangat bagus dari berbagai pihak, baik dari tenaga pendidik, siswa dan juga masyarakat.

Berikut beberapa kegunaan dari *website* sekolah, diantaranya:

1. Memperkenalkan profil sekolah
2. Membantu berinteraksi semua civitas akademik, baik dari internal sekolah maupun eksternal sekolah.
3. Media publikasi informasi remi ke masyarakat.
4. Menampilkan karya siswa.
5. Penerimaan peserta didik baru secara *online*.

2.1.4. Konsep Dasar Pemrograman

a. HTML

Menurut Hidayatullah dan Kawistara (2014:13) “*Hypertext Markup Language* (HTML) adalah bahasa *standard* yang digunakan untuk menampilkan halaman *web*”. Yang bisa dilakukan dengan HTML yaitu:

1. Mengatur tampilan dari halaman *web* dan isinya.
2. Membuat tabel dalam halaman *web*.
3. Mempublikasikan halaman *web* secara *online*.

4. Membuat form yang bisa digunakan untuk menangani registrasi dan transaksi via *web*.
5. Menambahkan objek-objek seperti citra, audio, video, animasi, java applet dalam halaman *web*.
6. Menampilkan area gambar (*canvas*) di *browser*.

b. PHP

Menurut Hidayatullah dan Kawistara (2015:231) “PHP *Hypertext Preprocessor* atau disingkat dengan PHP ini adalah suatu bahasa *scripting* khususnya digunakan untuk *web development*. Karena sifatnya yang *server side scripting*, maka untuk menjalankan PHP harus menggunakan web server”.

Pada awalnya PHP merupakan kependekan dari *Personal Home Page* (Situs Personal). PHP pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. Pada waktu itu PHP masih bernama *Form Interpreted* (FI), yang wujudnya berupa sekumpulan skrip yang digunakan untuk mengolah data formulir dari *web*.

Salah satu fitur yang paling signifikan dalam PHP adalah dukungan penggunaan untuk berbagai *database*. PHP juga memiliki dukungan untuk berkomunikasi dengan layanan lain menggunakan protokol seperti LDAP, IMAP, SNMP, NNTP, POP3, HTTP, COM (pada windows) dan lain-lain. Tak hanya itu, PHP juga memiliki fitur pengolahan teks yang sangat baik, dari *POSIX Extended* atau *Perl regular expression* untuk memarsing dokumen dari XML

c. Database MySQL

Menurut Hidayatullah dan Kawistara (2015:180) “MySQL adalah salah satu aplikasi *Database Management System* (DBMS) yang sudah sangat banyak digunakan oleh para pemrograman aplikasi *web*”.

Kelebihan dari MySQL adalah gratis, handal, selalu di-*update* dan banyak forum yang memfasilitasi para pengguna jika memiliki kendala. MySQL juga menjadi DBMS yang sering di *bundling* dengan *web server* sehingga proses instalasinya menjadi lebih mudah.

d. Web Server

Menurut Hidayatullah dan Kawistara (2015:125) “*Web server* adalah tempat dimana menyimpan aplikasi *web* kemudian mengaksesnya melalui internet”. Salah satu *web server* yang terkenal adalah Apache. Apache sebetulnya merupakan salah satu projek dari sebuah yayasan yang memiliki banyak projek software gratis untuk digunakan oleh banyak orang. Yayasan itu bernama The Apache Software Foundation.

Aplikasi *web server* Apache bisa digunakan bukan hanya di windows, tetapi juga di Linux dan Mac. Apache *web server* sudah sangat populer di internet sejak 1996. Kelebihan lainnya adalah The Apache Software Foundation sangat tinggi komitmennya untuk terus mengembangkan *web server* Apache sehingga keterjaminannya untuk senantiasa kompatibel dengan teknologi *web* terkini sangat tinggi.

Dan yang terakhir, begitu banyak forum-forum untuk bertanya jika kita memiliki masalah dalam penggunaan *web server* Apache.

2.1.5. Peralatan Pendukung

A. *UML*

Menurut Widodo dan Herlawati (2011:6) “UML singkatan dari *Unified Modeling Language* yang berarti bahasa pemodelan standar. Ketika kita membuat model menggunakan konsep *UML* ada aturan-aturan yang harus diikuti. Bagaimana pada model-model yang kita buat berhubungan satu dengan lainnya harus mengikuti standar yang ada”.

UML diaplikasikan untuk maksud tertentu, biasanya antara lain untuk:

1. Merancang perangkat lunak.
2. Sarana komunikasi antara perangkat lunak dengan proses bisnis.
3. Menjabarkan sistem secara rinci untuk analisa dan mencari apa yang diperlukan sistem.
4. Mendokumentasikan sistem yang ada, proses-proses dan organisasinya.

Blok pembangun utama *UML* adalah diagram. Beberapa diagram ada yang rinci (jenis *timing* diagram) dan lainnya ada yang bersifat umum (misalnya diagram kelas).

UML mendefinisikan diagram-diagram sebagai berikut:

1. *Use Case Diagram*

UML menyediakan serangkaian gambar dan diagram yang sangat baik.

Beberapa diagram menfokuskan diri pada ketangguhan teori object-

oriented dan sebagian lagi memfokuskan pada detail rancangan dan konstruksi.

Komponen pembentuk diagram use case adalah:

- a. Aktor (*actor*), menggambarkan pihak-pihak yang berperan dalam sistem.
- b. *Use Case*, aktivitas/sarana yang disiapkan oleh bisnis/sistem.
- c. Hubungan (*link*), aktor mana saja yang terlibat dalam *use case* ini.

2. *Class Diagram*

Diagram kelas adalah inti dari proses pemodelan objek. Baik *forward engineering* maupun *reverse engineering* memanfaatkan diagram ini. *Forward engineering* adalah proses perubahan model menjadi kode program, sedangkan *reverse engineering* sebaliknya merubah program menjadi model.

3. *Package Diagram*

Paket menyediakan sarana menyatukan elemen-elemen UML. Sebagai contoh kita dapat meletakkan elemen-elemen yang melakukan proses penggambaran tiga dimensi (*3D Rendering*) dalam satu paket yang bernama *3Dgraphics*. Diagram paket juga dapat memvisualisasikan ketergantungan antar bagian dalam suatu sistem dan bermanfaat dalam mencari letak permasalahan dalam kompilasi.

4. *Deployment Diagram*

Model diagram *deployment* bagian-bagian perangkat lunak suatu sistem ke perangkat keras yang akan mengeksekusinya. Elemen-elemen perangkat lunak seperti komponen, kelas, paket dan sebagainya dimanifestasikan menggunakan artifak serta dipetakan ke perangkat keras yang akan menjalankannya dengan titik (*nodes*). Komunikasi antar titik dimodelkan dengan jalur komunikasi (*communications paths*).

Walaupun diagram komponen dan deployment merupakan bagian dari spesifikasi UML, bagi sebagian besar perancang merupakan hal baru. Terkadang dalam implementasinya tidak mengikuti aturan yang ada, oleh karena itu kita diharuskan mengambil keputusan yang tepat, berlatih, berkonsentrasi terhadap keuntungan yang diperoleh dari sistem yang kita buat tanpa begitu saja mengikuti standar yang ada.

5. *Statechart Diagram*

Diagram *statechart* dalam UML kadang disebut dengan istilah diagram *state machine*. Diagram ini menggambarkan perilaku sistem perangkat lunak yang kita buat dan perilaku kelas, subsistem dan seluruh aplikasi. Selain itu diagram *state machine* bermanfaat juga untuk menyediakan cara yang baik dalam memodelkan komunikasi yang terjadi dengan entitas luar via *protocol* atau sistem dasarnya.

UML memiliki dua state machine antara lain:

- a. *Behavioral State Machine*. Mesin ini digunakan untuk menunjukkan perilaku elemen yang dimodelkan, misalnya suatu objek. Suatu

behavioral state machine merepresentasikan implementasi khusus suatu elemen.

- b. *Protocol State Machine*. Mesin ini digunakan untuk menunjukkan perilaku protokol. *Protocol state machine* menunjukkan bagaimana suatu elemen memicu perubahan kondisi protokol yang berdampak pada suatu sistem.

6. *Activity Diagram*

Diagram aktivitas lebih memfokuskan diri pada eksekusi dan alur sistem dari pada bagaimana sistem itu dirakit. Diagram ini tidak hanya memodelkan *software* melainkan memodelkan model bisnis juga. Diagram aktivitas menunjukkan aktivitas sistem dalam bentuk kumpulan aksi-aksi.

Ketika digunakan dalam pemodelan *software*, diagram aktivitas mempresentasikan pemanggilan suatu fungsi tertentu misalnya *call*.

Sedangkan bila digunakan dalam pemodelan bisnis, diagram ini menggambarkan aktivitas yang dipicu oleh kejadian-kejadian diluar seperti pemesanan atau kejadian-kejadian *internal* misalnya proses penggajian tiap jumat sore.

7. *Component Diagram*

Manfaat diagram komponen adalah bila ada salah satu komponen yang rusak atau tidak sesuai dengan tujuan sistem, kita tinggal mengganti komponen itu dengan komponen yang lain.

Agar komponen dapat dibongkar pasang harus memenuhi kriteria sebagai berikut:

- a. Memiliki kerja *internal* yang tersembunyi. Objek yang berada di dalamnya harus bebas akses oleh objek di luar komponen.
- b. Memiliki antar muka (*interface*). *Interface* mendiskripsikan operasi apa yang harus diambil terhadap suatu komponen dan bukan bagaimana operasi itu dikerjakan. Menyediakan *interface* adalah suatu cara untuk menyembunyikan kerja *internal* suatu komponen dari objek di luar komponen.
- c. Komponen di dalam harus *independent*. Kita harus yakin bahwa objek di dalam komponen kita tidak tahu menahu dengan objek lain diluar komponen sebab jika tidak saat kita mengganti objek tersebut, sistem akan terganggu.
- d. Antar muka terhadap komponen lain harus tersedia. Manfaatnya agar objek *internal* berhubungan dengan objek dikomponen lain lewat antar muka. Jadi saat objek *internal* ingin berhubungan dengan objek lain diluar komponen alurnya adalah lewat antar muka komponennya kemudian menuju antar muka komponen objek sasaran yang diteruskan ke objek itu.

Hubungan antara komponen dengan kelas adalah bahwa komponen merupakan subsistem berisi kelas-kelas yang bekerja sama merealisasikan antar muka yang di instruksikan.

8. *Sequence Diagram*

Diagram sequence dimaksudkan untuk mengembangkan komunikasi antara objek, bukan memanipulasi data saat berkomunikasi. Bersifat dinamis. Diagram sequence adalah diagram interaksi yang menekankan pada pengiriman pesan dalam suatu waktu tertentu.

9. *Communication Diagram*

Diagram komunikasi merupakan nama lain dari diagram kolaborasi pada UML 1.4 atau versi sebelumnya. Diagram komunikasi dimaksudkan untuk melengkapi diagram urutan dengan memberikan tampilan visual pada pesan yang disampaikan antar objek. Dasar dari diagram komunikasi adalah diagram objek. Tiap objek dalam diagram komunikasi disebut garis hidup objek (*object lifeline*).

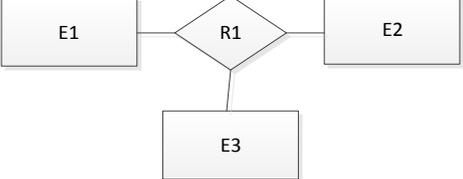
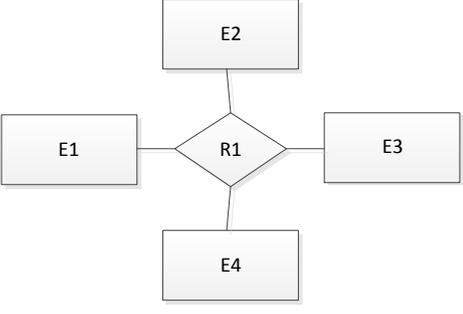
B. ERD (*Entity Relationship Diagram*)

Menurut Sukamto dan Shalahuddin (2015:50) “Erd (*Entity Relationship Diagram*) dikembangkan berdasarkan teori himpunan dalam bidang matematika. ERD digunakan untuk pemodelan basis data relasional. ERD biasanya memiliki hubungan *binary* (satu relasi menghubungkan dua buah entitas). Beberapa metode perancangan ERD menoleransi hubungan relasi *ternary* (satu relasi menghubungkan tiga buah relasi) atau *N-ary* (satu relasi menghubungkan banyak entitas), tapi banyak metode perancangan ERD yang tidak mengizinkan hubungan *ternary* atau *N-ary*”.

Berikut ini adalah contoh bentuk hubungan relasi dalam ERD:

Tabel II.1

Relasi Dalam ERD

Nama	Gambar
<i>Binary</i>	
<i>Ternary</i>	
<i>N-ary</i>	

2.2. Penelitian Terkait

Beberapa penelitian tentang perancangan *website* telah dilakukan sebelumnya, dalam penelitian ini penulis memaparkan tiga penelitian terdahulu yang berhubungan dengan permasalahan yang akan diteliti. Beberapa penelitian tersebut adalah:

Pringgo Winoto (2014), melakukan pembuatan *website* profil Sekolah Dasar Negeri 01 Blumbang. Perancangan *web* untuk Sekolah Dasar Negeri 01 Blumbang bertujuan untuk menghasilkan *website* profil sekolah untuk menyampaikan informasi dan mempromosikan tentang Sekolah Dasar Negeri 01 Blumbang kepada masyarakat umum

Sugiyanto (2013), melakukan pembuatan *website* profil Sekolah Menengah Kejuruan Pembangunan Nusantara Gabus Grobogan. Perancangan *web* Sekolah Menengah Kejuruan Pembangunan Nusantara Gabus Grobogan bertujuan untuk:

1. Menjadi sarana komunikasi paling mudah dan murah antara staf pengajar, siswa, orang tua, alumni dan pihak-pihak yang terlibat dalam dunia pendidikan, dalam mengakses informasi tentang SMK Pembangunan Nusantara Gabus Grobogan.
2. Dapat mengangkat serta meningkatkan citra positif SMK Pembangunan Nusantara Gabus Grobogan.
3. Mempersiapkan sekolah dalam mengimbangi kemajuan teknologi internet.

Dina Khusnia (2014), melakukan pembuatan *website* profil Sekolah Menengah Pertama Negeri 1 Kalitidu Bojonegoro. Perancangan *website* profil Sekolah Menengah Pertama Negeri 1 Kalitidu Bojonegoro bertujuan untuk menghasilkan *website* profil Sekolah Menengah Pertama Negeri 1 Kalitidu Bojonegoro yang dinamis, yang dapat digunakan sebagai media penyampaian informasi yang lebih efektif dan efisien.