

BAB II

LANDASAN TEORI

2.1 Sistem

Sistem merupakan komponen yang memiliki unsur keterkaitan dan terintegrasi antara satu dengan yanglainnya dan saling bekerja sama untuk mencapai tujuan. Umumnya komponen-komponen pada sebuah sistem salingberhubungan dan berintegrasi untuk mencapai tujuan.(Achmad Nazrul :2013)

2.2 Informasi

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 (EdhySutanta: 2011).

Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya (Yakub: 2012)

2.3 Sistem Informasi

Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan data transaksi harian, mendukung operasi, bersifat manajerial, dan kegiatan strategi dari suatu organisasi serta menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan (Yakub:2012)

2.4 Pendaftaran

Pendaftaran adalah pencatatan hal atau identitas seperti nama, alamat dan sebagainya dalam suatu lembaga pendidikan, pendaftaran merupakan suatu hal yang sangat penting. Pengertian pendaftaran disini pada dasarnya hanya untuk memperlancar dan mempermudah dalam pendaftaran sehingga terorganisir, teratur dengan capat atau tepat. (Wirda Asrar : 2013)

2.5 Internet

Istilah internet berasal dari bahasa Latin inter, yang berarti "antara". Secara kata perkata internet berarti jaringan antara atau penghubung. Memang itulah fungsinya, internet menghubungkan berbagai jaringan yang tidak saling bergantung pada satu sama lain sedemikian rupa sehingga dapat berkomunikasi.

Jadi dapat diambil kesimpulan bahwa internet merupakan hubungan antar berbagai jenis komputer dan jaringan di dunia yang berbeda sistem operasi maupun aplikasinya dimana hubungan tersebut memanfaatkan kemajuan media komunikasi (telepon dan satelit) yang menggunakan protokol standar dalam berkomunikasi yaitu protokol TCP/IP. (Achmad Nazrul :2013)

2.6 Website

Website atau situs adalah kumpulan halaman yang menampilkan informasi data teks, data gambar diam atau gerak, data animasi, suara, video dan atau gabungan dari semuanya, baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait dimana masing-masing dihubungkan dengan jaringan-jaringan halaman (*Hyperlink*). (Achmad Nazrul :2013)

2.7 XAMPP

XAMPP adalah perangkat lunak bebas yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program. Fungsinya adalah sebagai server yang berdiri sendiri (localhost), yang terdiri atas program Apache HTTP Server, MySQL Database, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl. (Sarwandi :2017)

2.7.1 Apache

Apache merupakan aplikasi web server. Tugas utama apache adalah menghasilkan halaman web yang benar kepada user berdasarkan kode PHP yang dituliskan oleh pembuat halaman web. (Sarwandi :2017)

2.7.2 MySQL

MySQL merupakan aplikasi database server. Perkembangannya disebut SQL yang merupakan kepanjangan dari *Structured Query Language*. SQL merupakan bahasa terstruktur yang digunakan untuk membuat dan mengelolah database beserta isinya. (Sarwandi :2017)

2.7.3 PHP

PHP merupakan bahasa pemrograman web. Bahasa pemrograman PHP merupakan bahasa pemrograman untuk membuat web yang bersifat *server side scripting*. PHP memungkinkan kita untuk membuat halaman web yang bersifat dinamis. (Sarwandi :2017)

PHP merupakan bahasa *server-side* yang cukup handal, yang akan disatukan dengan HTML (*Hypertext Markup Language*) dan berada di *server*. Artinya, sintaks dan perintah yang diberikan akan sepenuhnya dijalankan di server sebelum dikirim ke komputer *klien*. Pada awal tahun 1995, Rasmus Ledorf membuat produk bernama PHP/FI PHP ini ditulis dengan menggunakan bahasa C, dan memiliki kemampuan untuk berkomunikasi dengan database serta membuat halaman dinamis.(Achmad Nazrul :2013)

2.7.3.1 Variabel PHP

Variabel PHP digunakan untuk menyimpan data yang nilainya dapat berubah-ubah. Dalam bahasa PHP, variabel dimulai dengan tanda "\$". Aturan penulisan variabel antara lain sebagai berikut :

- a. Hanya ada 3 karakter yang dapat digunakan untuk nama variable, yaitu huruf, angka dan garis bawah.
- b. Karakter pertama setelah tanda "\$" harus huruf atau garis bawah.
- c. Jika nama variable lebih dari satu kata, tidak boleh ada tanda spasi diantara keduanya.(Achmad Nazrul :2013)

2.7.3.2 Keunggulan PHP

PHP Hypertext Processor adalah skrip yang berjalan dalam server side yang ditambahkan dalam HTML. Skrip ini akan membuat suatu aplikasi dapat diintegrasikan ke dalam HTML sehingga suatu halaman HTML tidak lagi bersifat statis, namun menjadi bersifat dinamis. Sifat server side ini membuat pekerjaan skrip tersebut dikerjakan di server sedangkan yang dikirimkan ke browser adalah hasil proses dari skrip tersebut yang sudah berbentuk HTML. Keunggulan dari sifatnya yang server-side tersebut adalah :

- a. Tidak diperlukan adanya kompatibilitas browser atau harus menggunakan browser tertentu, karena serverlah yang akan mengerjakan skrip tersebut. Hasil yang dikirimkan kembali ke browser biasanya dalam bentuk teks ataupun gambar sehingga dapat dikenali oleh browser apapun.
- b. Dapat memanfaatkan sumber-sumber aplikasi yang dimiliki oleh server, contoh : hubungan kedalam database
- c. Skrip asli tidak terlihat sehingga keamanan lebih terjamin.(Achmad Nazrul :2013)

2.8 *Unified Modeling Language (UML)*

Bahasa pemodelan untuk sistem atau perangkat lunak yang berparadigma berorientasi objek. Pemodelan (*modeling*) sesungguhnya digunakan untuk penyederhanaan permasalahan yang kompleks sehingga lebih mudah untuk dipelajari dan dipahami. Tujuan pemodelan dalam kerangka pengembangan sistem adalah sebagai sarana analisis, pemahaman, visualisasi, dan komunikasi antar tim pengembang yang beranggotakan beberapa/banyak anggota.

Beberapa diagram dalam *UML* yang akan digunakan dalam membantu pengembangan sistem yaitu (Jurnal Informatika IBI DARMAJAYA Vol.13 No.2)






:

2.8.1 *Use Case Diagram*

Merupakan unit koheren dari fungsionalitas sistem yang tampak dari luar dan diekspresikan sebagai urutan pesan-pesan yang dipertukarkan unit-unit sistem dengan satu atau lebih *actor* yang sda diluar sistem. Kegunaan

use case sesungguhnya adalah untuk mendefinisikan suatu bagian perilaku sistem yang bersifat koheren tanpa perlu menyiapkan struktur internal sistem yang sedang dikembangkan. Definisi *use case* di dalamnya mencakup semua perilaku yang ada dalam sistem yang sedang kita kembangkan. Simbol dan keterangan *use case diagram* seperti pada tabel 2.1.





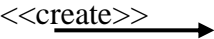
Tabel 2.1 Simbol dan Keterangan *Use Case Diagram*

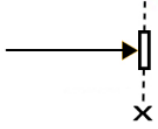
SIMBOL	KETERANGAN
<p data-bbox="411 757 488 786"><i>Actor</i></p>  <p data-bbox="497 954 587 976">NewClass</p>	<p data-bbox="715 757 1353 842">Mewakili peran orang, <i>sistem</i> yang lain atau alat ketika berkomunikasi dengan <i>use case</i>.</p>
<p data-bbox="411 1081 531 1111"><i>Use case</i></p>  <p data-bbox="507 1223 598 1238">NewUseCase</p>	<p data-bbox="715 1081 1331 1111">Abstraksi dari interaksi antara <i>sistem</i> dan <i>actor</i>.</p>
<p data-bbox="411 1290 576 1319">Generalisasi</p> 	<p data-bbox="715 1290 1353 1375">Menunjukkan spesialisasi <i>actor</i> untuk dapat berpartisipasi dalam <i>use case</i>.</p>
<p data-bbox="411 1480 507 1509"><i>Extend</i></p> 	<p data-bbox="715 1480 1353 1621">Mempesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.</p>
<p data-bbox="411 1671 512 1700"><i>Include</i></p> 	<p data-bbox="715 1671 1353 1756">Menunjukkan bahwa suatu <i>use case</i> seluruhnya merupakan fungsionalitas dari <i>use case</i> lainnya.</p>

2.8.2 Sequence Diagram

Sequence Diagram menekankan pada urutan waktu penerimaan *message*, kita menjumpai garis hidup objek yaitu garis tegas *vertical* yang mencerminkan ekstensi sebuah objek sepanjang periode waktu. Kebanyakan objek yang hadir pada *interaction diagram* akan eksis sepanjang durasi tertentu dari interaksi, sehingga objek itu diletakkan di atas diagram dengan ‘garis hidup’ digambarkan dari atas hingga ke bagian bawah diagram. Simbol dan keterangan *sequence diagram* seperti pada tabel 2.2.

Tabel 2.2 Simbol dan Keterangan *Sequence Diagram*

SIMBOL	KETERANGAN
Objek 	Berpartisipasi secara berurutan dengan mengirimkan atau menerima pesan dan ditempatkan di bagian atas diagram.
Garis hidup objek 	Menandakan kehidupan obyek selama urutan dan diakhiri tanda X pada titik dimana kelas tidak lagi berinteraksi.
Waktu aktif 	menandakan ketika suatu objek mengirim atau menerima pesan.
Pesan 	Objek mengirim satu pesan ke objek lainnya.
Create 	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.





<p><i>Destroy</i></p> 	<p>Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah objek yang diakhiri.</p>
---	--




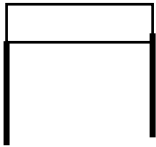
Tabel 2.2 Simbol dan Keterangan *Sequence Diagram (Lanjutan)*

2.8.3 Activity Diagram

Activity Diagram atau Diagram Aktivitas menggambarkan alur aktivitas dalam aplikasi, menjelaskan proses masing-masing alur berawal dan proses aplikasi berakhir. Diagram aktivitas juga menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. Simbol dan keterangan *activity diagram* seperti pada tabel 2.3

Tabel 2.3 Simbol dan Keterangan *Activity Diagram*

SIMBOL	KETERANGAN
Status Awal 	Status awal aktivitas <i>sistem</i> , sebuah diagram aktivitas memiliki sebuah status awal.
Aktivitas 	Aktivitas yang dilakukan <i>sistem</i> , aktivitas biasanya diawali dengan kata kerja.
Percabangan 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
Status akhir 	Status akhir yang dilakukan <i>sistem</i> , sebuah diagram aktivitas memiliki sebuah status akhir.



	
<i>Action</i> 	Memperlihatkan masing-masing kelas antar muka saling berinteraksi satu sama lain.
<i>Fork Node</i> 	Satu aliran pada tahap tertentu berubah menjadi beberapa aliran.
<i>Swimlane</i> 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.


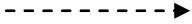
Tabel 2.3 Simbol dan Keterangan *Activity Diagram (Lanjutan)*

2.8.4 Class Diagram

Class Diagram atau Diagram Kelas merupakan diagram yang memodelkan sekumpulan kelas, *interface*, kolaborasi dan relasinya. Diagram kelas digambarkan dengan bentuk kotak. Simbol dan keterangan *class diagram* seperti pada tabel 2.4

Tabel 2.4 Simbol dan Keterangan *Class Diagram*

SIMBOL	KETERANGAN
<i>Class</i> 	Himpunan dari objek-objek yang berbagai atribut serta operasi yang sama.
<i>Nary Association</i> 	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.

<i>Generalization</i> 	Hubungan dimana objek anak (<i>descendent</i>) berbagai perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>oncestor</i>).
<i>Realization</i> 	Operasi yang benar-benar dilakukan oleh suatu objek.

Tabel 2.4 Simbol dan Keterangan *Class Diagram*(Lanjutan)

2.9 Metode Pengembangan Sistem

Prototyping adalah pengembangan yang cepat dan pengujian terhadap model kerja (*prototipe*) dari aplikasi baru melalui proses interaksi dan berulang-ulang yang biasa digunakan ahli sistem informasi dan ahli bisnis. *Prototyping* disebut juga desain aplikasi cepat (*rapid application design/RAD*) karena menyederhanakan dan mempercepat desain sistem. Proses pengembangan *prototipe* diuraikan dalam tahap sebagai berikut (Jurnal Informatika IBI DARMAJAYA Vol.14 No.1):

A. Analisis Kebutuhan Sistem

Pembangunan sistem informasi memerlukan penyelidikan dan analisis mengenai alasan timbulnya ide atau gagasan untuk membangun dan mengembangkan sistem informasi. Analisis dilakukan untuk melihat berbagai komponen yang dipakai sistem yang sedang berjalan meliputi *hardware*, *software*, jaringan dan sumber daya manusia. Analisis juga mendokumentasikan aktivitas sistem informasi meliputi *input*, pemrosesan, *output*, penyimpanan dan pengendalian. Selanjutnya melakukan studi kelayakan (*feasibility study*) untuk merumuskan informasi yang dibutuhkan pemakai akhir, kebutuhan sumber daya, biaya, manfaat dan kelayakan proyek yang diusulkan. Analisis kebutuhan sistem sebagai bagian dari studi awal bertujuan mengidentifikasi masalah dan kebutuhan spesifik sistem. Kebutuhan spesifik sistem adalah spesifikasi mengenai hal-hal yang akan dilakukan sistem ketika diimplementasikan.

B. Desain Sistem.

Analisis sistem (*systemanalysis*) mendeskripsikan apa yang harus dilakukan sistem untuk memenuhi kebutuhan informasi pemakai. Desain sistem (*systemdesign*) menentukan bagaimana sistem akan memenuhi tujuan tersebut. Desain sistem terdiri dari aktivitas desain yang menghasilkan spesifikasi fungsional. Desain sistem dapat dipandang sebagai desain *interface*, data dan proses dengan tujuan menghasilkan spesifikasi yang sesuai dengan produk dan metode *interface* pemakai, struktur database serta pemrosesan dan prosedur pengendalian.

Pada tahap ini akan dilakukan perancangan *prototipe*. *Prototipe* akan dirancang menggunakan alat bantu berupa *UseCase* dan *Activity* diagram. *UseCase* diagram digunakan untuk menggambarkan interaksi antara pengguna dan sistem, sedangkan *Activity* diagram digunakan untuk menggambarkan aktivitas yang terjadi di dalam sistem ketika digunakan. Hasil perancangan kemudian digambarkan dalam bentuk tampilan pengguna atau *UserInterface*.

C. Pengujian Sistem.

Paket *softwareprototipe* diuji, diimplementasikan, dievaluasi dan dimodifikasi berulang-ulang hingga dapat diterima pemakainya. Pengujian sistem bertujuan menemukan kesalahan-kesalahan yang terjadi pada sistem dan melakukan revisi sistem. Tahap ini penting untuk memastikan bahwa sistem bebas dari kesalahan.

Prototipe yang dihasilkan kemudian akan di evaluasi untuk melihat apakah *prototipe* yang dihasilkan sudah sesuai dengan persyaratannya, tidak lagi terdapat kesalahan terhadap kode programnya ataupun jika terdapat fungsi-fungsi lain yang perlu untuk ditambahkan pengembang sesuai dengan kebutuhan pengguna.

D. Implementasi

Setelah *prototipe* diterima maka pada tahap ini merupakan implementasi sistem yang siap dioperasikan dan selanjutnya terjadi proses pembelajaran terhadap sistem baru dan membandingkannya dengan sistem lama, evaluasi secara teknis dan operasional serta interaksi pengguna, sistem dan teknologi informasi.