

BAB II

LANDASAN TEORI

2.1 Sistem Pengolahan Data Penduduk Desa Saragih

Desa Saragih adalah salah satu instansi pemerintah daerah yang bergerak dibidang pelayanan masyarakat. Menangani berbagai kegiatan seperti penerbitan dokumen penting, kependudukan dan pencatatan sipil. Saat ini pendataan penduduk masih manual karena belum menggunakan sistem informasi berbasis teknologi. Saat penduduk ingin mengurus administrasi di kantor kepala desa, maka penduduk harus meminta berkas yang dibutuhkan ke bagian sekretaris desa. Setelah diisi kemudian meminta tandatangan kepala desa yang kemudian diserahkan kembali kepada bagian sekretaris untuk disimpan berkasnya. Proses tersebut terbilang kurang efisien. Pengolahan data penduduk ini membutuhkan adanya komputer dalam menjalankan kegiatan dan untuk menyelesaikan permasalahan yang dihadapi serta mengurangi tingkat kesalahan manusia dalam mengolah data penduduk. Kegiatan pengolahan data penduduk di desa Saragih masih menggunakan sistem manual dengan menggunakan Microsoft Word dan Excel untuk pengolahan surat dan administrasi serta buku induk desa yang digunakan untuk menyimpan data penduduk desa.

2.2 Pengertian Sistem

Sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari unsur, komponen, atau variable yang terorganisir, saling berinteraksi, saling bergantung satu sama sama lain dan terpadu karena sistem memiliki sasaran yang berbeda untuk setiap kasus yang terjadi dalam sistem tersebut [4].

2.3 Pengertian Informasi

Informasi adalah hasil pemrosesan data yang diperoleh dari setiap elemen sistem tersebut menjadi bentuk yang mudah dipahami dan merupakan pengetahuan

yang relevan yang dibutuhkan oleh setiap orang untuk menambah pemahamannya terhadap fakta-fakta yang ada [5].

2.4 Pengertian Sistem Informasi

Sistem informasi merupakan suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung suatu fungsi organisasi yang bersifat manajerial dalam kegiatan strategi dari suatu organisasi untuk dapat menyediakan kepada pihak luar dengan laporan laporan yang diperlukan [6].

2.5 Komponen Sistem Informasi

Sistem informasi dalam mendukung beberapa komponen yang fungsinya sangat vital di dalam sistem informasi. Komponen-komponen sistem informasi tersebut adalah *hardware*, *software*, prosedur, pengguna dan *database*. Secara rinci komponen- komponen system informasi dapat dijelaskan sebagai berikut:

1. Perangkat keras (Hardware), mencakup peranti-peranti fisik seperti monitor dan printer.
2. Perangkat lunak (software) atau program: sekumpulan intruksi yang memungkinkan perangkat keras untuk dapat memproses data
3. Prosedur: sekumpulan aturan yang dipakai untuk mewujudkan pemrosesan data dan data keluaran yang dikendaki.
4. Pengguna: semua pihak yang bertanggung jawab dalam pengembangan system informasi, pemrosesan, dan penggunaan keluaran sistem informasi.
5. Database: merupakan kumpulan dari data yang saling berhubungan dengan data lainnya, tersimpan diperangkat keras komputer dan digunakan perangkat lunak untuk memanipulsinya, diantaranya; data, user dan sistem. [7].

2.6 Pengertian Aplikasi

Aplikasi adalah suatu program yang siap untuk digunakan yang dibuat untuk melaksanakan suatu fungsi bagi pengguna jasa aplikasi serta penggunaan aplikasi lain yang dapat digunakan oleh suatu sasaran yang akan dituju [8]. Aplikasi desktop adalah suatu aplikasi yang dapat berjalan sendiri atau independen tanpa menggunakan *browser* atau koneksi internet di suatu komputer otonomi dengan sistem operasi atau *platform* tertentu [9].

2.7 Java

Java merupakan teknologi dimana teknologi tersebut mencakup java sebagai bahasa pemrograman yang memiliki sintaks dan aturan pemrograman tersendiri, juga memiliki *virtual machine* dan *library* yang diperlukan untuk menulis dan menjalankan program yang ditulis dengan bahasa pemrograman java[10]. Java adalah bahasa pemrograman yang multi platform dan multi device. Sekali anda menuliskan sebuah program dengan menggunakan Java, anda dapat menjalankan hampir di semua computer dan perangkat lain yang mensupport java, dengan sedikit perubahan atau tanpa perubahan sama sekalidalam kodenya.

2.8 NetBeans

NetBeans yaitu software yang digunakan dalam pengembangan sistem berbasis Java , open source , terintegrasi (IDE) yang memungkinkan untuk mengembangkan aplikasi desktop, seluler dan web[11].

2.9 MySQL

MySQL merupakan software *open source*. *Opensource* berarti semua orang diijinkan untuk menggunakan dan memodifikasinya. Semua orang dapat menggunakan secara gratis dan tidak perlu membayar. Sebagai Program penghasil Database, MySql tidak berjalan sendiri tanpa adanya sebuah aplikasi lain baik yang OpenSource maupun tidak, seperti PHP, VB, Delphi dan lainnya. Untuk mengaktifkan dilingkungan Windows makaharus dijalankan server MySql untukAdministrasi database seperti pembuatan database, pembuatan table, dan sebagainya, dapat digunakan aplikasi berbasis Web seperti PHP MyAdmin.

2.10 MySQL

MySQL merupakan software *open source*. *Opensource* berarti semua orang diijinkan untuk menggunakan dan memodifikasinya. Semua orang dapat menggunakan secara gratis dan tidak perlu membayar. Sebagai Program penghasilDatabase, MySql tidak berjalan sendiri tanpa adanya sebuah aplikasi lain baik yangOpenSource maupun tidak, seperti PHP, VB, Delphi dan lainnya. Untuk mengaktifkan dilingkungan Windows makaharus dijalankan server MySql untukAdministrasi database seperti pembuatan database, pembuatan table, dan sebagainya, dapat digunakan aplikasi berbasis Web seperti PHP MyAdmin.

2.11 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 bahas programan PHP dan Perl.

2.12 UML (Unified Modeling Language)

Ialah sebuah metode untuk memvisualisasikan sebuah perancangan system berorientasi objeksebuah metode pemodelan yang digunakan untuk memvisualisasikan bantu yang digunakan dalam perancangan berorientasi objek berbasis UML adalah sebagai beriku

2.12.1 Use Case Diagram






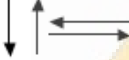
Use case diagram merupakan pemodelan untuk kelakuan (behavior) sistem informasi yang akan dibuat. Use case digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Berikut tabel simbol dalam Usecase Diagram

GAMBAR	NAMA	KETERANGAN
	<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
	<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri.
	<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
	<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara <i>eksplisit</i> .
	<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
	<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
	<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
	<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
	<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (<i>sinergi</i>).
	<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

Gambar 2. 1 Simbol Usecase Diagram

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis.

Berikut gambar *table* Diagram Aktivitas:

NO	GAMBAR	NAMA	KETERANGAN
1		Activity	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2		Action	State dari sistem yang mencerminkan eksekusi dari suatu aksi
3		Initial Node	Bagaimana objek dibentuk atau diawali.
4		Activity Final Node	Bagaimana objek dibentuk dan diakhiri
5		Decision	Digunakan untuk menggambarkan suatu keputusan / tindakan yang harus diambil pada kondisi tertentu
6		Line Connector	Digunakan untuk menghubungkan satu simbol dengan simbol lainnya







Gambar 2. 2 Simbol Diagram Activity

2.12.2 Diagram Kelas (*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 atribut-atribut dan operasi-operasi dari sebuah kelas dan constraint yang berhubungan dengan objek yang dikoneksikan.

Berikut gambar tabel simbol Kelas Diagram:

Daftar Simbol

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
2		<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3		<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4		<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
5		<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
6		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan memengaruhi elemen yang bergantung padanya elemen yang tidak mandiri
7		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya

Gambar 2. 3 Simbol Diagram Class

