

BAB II

TINJAU PUSTAKA DAN LANDASAN TEORI

2.1. Tinjauan Pustaka

Pembuatan aplikasi mengenai sistem informasi pengarsipan juga sudah pernah dilakukan sebelumnya dalam skripsi yang berjudul “Perancangan Sistem Informasi Arsip Surat Menyurat di Universitas U’Budiyah Indonesia Menggunakan *Php* dan *MySQL*”. Skripsi tersebut bertujuan untuk memperbaiki sistem sebelumnya yang masih manual dan yaitu semua data surat masuk dan keluar diarsipkan pada satu tempat dan laporannya masih disimpan pada sebuah buku besar. Perancangan sistem tersebut menggunakan metode SSAD (*Structured Systems Analisis and Design*). Sedangkan pembuatan aplikasi tersebut menggunakan *MySQL* sebagai *database* dan *Php* sebagai media pembuatan tampilan aplikasi (Junidar 2012).

Aplikasi pengarsipan serupa juga pernah dibuat dalam skripsi yang berjudul “Pengembangan Sistem Informasi Kearsipan Tata Usaha Menggunakan Metode Agile di MTs Arrosyidin Secang Magelang”. Pembuatan aplikasi pada skripsi ini menggunakan bahasa pemrograman *Php* dan *MySQL* sebagai *databasenya*. Sedangkan untuk pengembangan sistemnya menggunakan metode *agile* (Lukman Hasan 2014).

Fuji dan Lelia Apriana (2011) mengembangkan sistem informasi “Administrasi dan Dokumen Ekspor Batu Baru pada PT Usaha Maju Makmur Palembang” alat yang digunakan adalah *Microsoft Visual Basic.Net*, *Database MySQL* dan *Crystal Report.Net 2008*. Sistem yang akan dikembangkan ini, dapat

memberikan kemudahan serta ketepatan bagi para karyawan dalam proses pengolahan data.

Berdasarkan penelitian terlebih dahulu yang telah dijelaskan diatas, dapat disimpulkan bahwa persamaannya adalah:

1. Membuat aplikasi berbasis *Desktop* sebagai media untuk memudahkan pendataan di instansi masing-masing.
2. Mempercepat informasi yang diperlukan di instansi masing-masing.
3. Menggunakan *MySQL* sebagai *databasenya*.

Sedangkan perbedaan yang dapat disimpulkan penulis adalah:

1. Penulis menggunakan Bahasa pemrograman *Java* sedangkan penelitian sebelumnya menggunakan Bahasa pemrograman *Php*.
2. Penelitian sebelumnya menggunakan alat *Microsoft Visual Basic.Net*, sedangkan penulis menggunakan alat *NetBeans IDE*.

2.2. Landasan Teori

2.2.1. Pengertian Sistem Informasi

Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi, dan menyediakan pihak luar tertentu dengan laporan – laporan yang diperlukan. (Jogiyanto, 2005).

Jadi dapat disimpulkan bahwa sistem informasi adalah suatu alat yang membantu dalam menyediakan informasi bagipenerimaanya dan untuk membantu

dalam pengambilan keputusan bagi manajemen didalam operasi perusahaan sehari-hari dan informasi yang layak untuk pihak luar perusahaan.

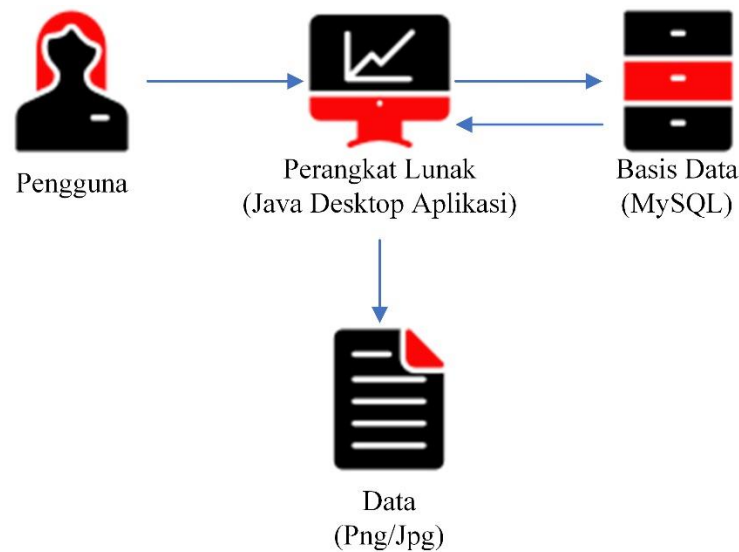
2.2.2. Aplikasi Berbasis *Desktop*

Aplikasi berbasis *desktop* merupakan aplikasi yang dapat berjalan sendiri atau *independent* tanpa menggunakan *browser* atau koneksi internet disuatu PC. (Dew Omenn, 2013).

Tidak perlu koneksi internet, karena semua file yang diperlukan untuk menjalankan aplikasinya sudah terinstal sebelumnya. Dapat dengan mudah memodifikasi setingnya dan prosesnya lebih cepat. Tetapi aplikasi berbasis *desktop* ini apabila akan menjalankan aplikasi harus terinstal terlebih dahulu di komputer. Bermasalah dengan lisensi, hal ini membutuhkan lisensi yang banyak pada setiap computer, dan biasanya memerlukan hardware dengan spesifikasi tinggi.

2.2.3. Arsitektur Perangkat Lunak

Arsitektur perangkat lunak adalah sekumpulan pernyataan yang menggambarkan komponen perangkat lunak dan fungsi-fungsi yang ada pada komponen tersebut. Ia menggambarkan struktur teknis, Batasan-batasan, ciri-ciri, serta antarmuka pada komponen-komponen tersebut. Arsitektur merupakan rancangan fisik sistem dan oleh karena itu membutuhkan rencana yang matang (Krafzig *et al*,2004). Arsitektur perangkat lunak pada aplikasi dapat dilihat pada Gambar 2.1.



Gambar 2. 1 Arsitektur Perangkat Lunak

2.2.4. *Unified Modelling Language (UML) Diagram*




Unified Modelling Language (UML) adalah sebuah bahasa permodelan visual yang dirancang khusus untuk pengembangan dan analisis sistem berorientasi objek dan desain. UML pertama kali dikembangkan oleh Grady Booch, Jim Rumbaugh, dan Ivars Jacobson pada pertengahan tahun 1990. (Keng Siau and Qing Cao, 2001:26).

Model UML yang dipakai dalam pembuatan aplikasi pengajuan dana belanja lab TI antara lain adalah:

1. *Use Case Diagram*

Use case diagram adalah sebuah diagram yang menunjukkan hubungan antara *actors* dan *use cases*. Digunakan untuk analisis dan desain sebuah system. (Scott W. Ambler, 2005:33). Simbol-simbol yang digunakan dalam *use case* diagram dapat dilihat pada Tabel 2.1 dan Tabel 2.2.

Tabel 2. 1 Simbol-simbol dalam *Use Case Diagram*

NO	Gambar	Nama Gambar	Keterangan
1.		<i>Use Case</i>	Merupakan fungsionalitas yang disediakan sistem sebagai unit yang bertukar pesan dengan <i>actor</i> .
2.		<i>Actor</i>	Merupakan <i>abstraction</i> dari orang yang mengaktifkan fungsi dari target sistem dan merupakan orang yang berinteraksi dengan <i>use case</i> .
3.		<i>Association</i>	Digambarkan dengan garis tanpa panah yang mengindikasikan siapa yang berinteraksi secara langsung dengan sistem.





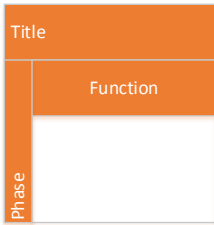
Tabel 2. 2 Simbol-simbol dalam *Use Case Diagram* (lanjutan)

NO	Gambar	Nama Gambar	Keterangan
4.		<i>Generalization</i>	Mengindikasikan siapa yang berinteraksi secara pasif dengan sistem.
5.		<i>Include</i>	Mengidentifikasi hubungan antar dua <i>use case</i> dimana satu <i>usecase</i> memanggil <i>usecase</i> yang lain.
6.		<i>Extend</i>	Merupakan perluasan dari <i>use case</i> jika kondisi atau syarat terpenuhi.

2. Activity Diagram

Activity Diagram adalah suatu diagram yang menggambarkan konsep aliran data atau control, aksi terstruktur serta dirancang dengan baik dalam suatu sistem. (Conrad Bock, 2003:45). Simbol-simbol yang digunakan dalam *activity diagram* dapat dilihat pada Tabel 2.3.

Tabel 2. 3 Simbol-simbol dalam *Activity Diagram*

NO	Gambar	Nama Gambar	Keterangan
1.		<i>Start Point</i>	Merupakan awal dalam aktifitas
2.		<i>End Point</i>	Merupakan akhir dalam aktifitas.
3.		<i>Activities</i>	Menggambarkan suatu proses atau kegiatan bisnis
4.		<i>Decision Point</i>	Menggambarkan pilihan untuk pengambilan keputusan dalam aktifitas.
5.		<i>Swimlane</i>	Digunakan untuk pembagian <i>activity</i> diagram yang menunjukkan siapa yang melakukan aktifitas.

3. *Class Diagram*

Class diagram adalah sebuah diagram yang menunjukkan hubungan antar *class* yang didalamnya terdapat atribut dan fungsi dari suatu objek. (Scott W. Ambler, 2005:47)

Class diagram mempunyai tiga relasi dalam penggunaannya, yaitu:

a. *Assosiation*

Assosiation adalah sebuah hubungan yang menunjukkan adanya interaksi antar *class*. Hubungan ini dapat ditunjukkan dengan garis dengan mata panah terbuka di ujungnya yang mengindikasikan adanya aliran pesan dalam satu arah.

b. *Generalization*

Generalization adalah sebuah hubungan antar *class* yang bersifat dari khusus ke umum.

c. *Constraint*

Constraint adalah sebuah hubungan yang digunakan dalam sistem untuk memberi Batasan pada sistem sehingga didapat aspek yang tidak fungsional.

2.2.5. *Entity Relationship Diagram (ERD)*

Entity relationship diagram adalah sebuah alat dalam suatu lingkup database yang digunakan dalam menganalisis dan mendesain data. ERD telah berhasil dibuktikan kehandalannya dalam hal menghubungkan *data field*, *variables*, maupun *database*. (Ronald Gege Allan, 2005:51).

Beberapa kelebihan ERD menurut Business Intelligence Journal: Ronald Gege Allan (2005:51) dibanding diagram sejenis seperti *use case* diagram dan *sequence* diagram adalah:

1. Mudah untuk dioperasikan dalam hal mencari *data field* yang dibutuhkan.
2. Mudah dalam hal melakukan Analisa terhadap data.
3. Kemampuan dalam hal menghilangkan redundansi data.

2.2.6. Basis Data

Basis data merupakan salah satu teknologi yang harus dimiliki sebuah perusahaan, institusi, ataupun organisasi adalah teknologi yang dapat memproses data. Secara konsep basis data atau *database* adalah kumpulan dari data-data yang membentuk suatu berkas (*file*) yang saling berhubungan (*relation*) dengan tata cara tertentu untuk membentuk data baru atau informasi. Kumpulan dari data yang saling berhubungan (relasi) antara satu dengan lainnya yang diorganisasikan berdasarkan skema atau struktur tertentu (Supriyanto, 2005). Untuk mengelola *database* diperlukan suatu perangkat lunak yang disebut DBMS (*Data Base Management System*). DBMS merupakan suatu sistem perangkat lunak yang memungkinkan pengguna untuk membuat, memelihara, mengontrol, dan mengakses *database*. Dengan DBMS, pengguna dapat mengontrol dan memanipulasi data yang ada.

2.2.7. MySQL

Menurut Adi Nugroho (2004: 1) *MySQL* adalah sebuah program *database* server yang mampu menerima dan mengirimkan datanya dengan sangat cepat, multi user serta menggunakan perintah standar SQL.

MySQL sebenarnya produk yang berjalan pada platform Linux. Karena sifatnya *open source*, MySQL dapat dijalankan pada semua platform baik Windows maupun Linux. Selain itu, MySQL juga merupakan program pengakses *database* yang bersifat jaringan sehingga dapat digunakan untuk aplikasi multiuser (banyak pengguna). Saat ini *database* MySQL telah digunakan hampir oleh semua programmer *database*, apalagi dalam pemrograman *web*.

[Sumber: <http://www.mysql.com> diakses pada 29 Oktober 2017]

2.2.8. SQLite JDBC

Untuk mengakses data yang ada di dalam *database SQLite*, maka diperlukan sebuah API yaitu *SQLite JDBC*. Fungsi dari *SQLite JDBC* sendiri adalah untuk mengeksekusi *statement-statement SQL* (seperti mengelompokkan data, menambah sebuah *record*, menciptakan tabel, dan lain sebagainya).

JDBC mendukung secara penuh aktifitas untuk:

1. Melakukan koneksi ke sebuah *Driver Database* Relasional
2. Dengan koneksi yang dibangun, JDBC dapat mengirim *statement-statement SQL* ke *driver database* relasional (seperti: *select*, *CRUD*, *resultset*, *metadata*, dan lain-lain).

3. Menampilkan sebuah set tabel dan relasi yang diinginkan. [Sumber: <http://www.sqlite.org/about.html> diakses pada 29 Oktober 2017]

2.2.9. Pengertian Java

Java dikembangkan oleh perusahaan Sun Microsystem. *Java* menurut definisi dari Sun Microsystem adalah nama untuk sekumpulan teknologi untuk membuat dan menjalankan perangkat lunak pada komputer *standalone* ataupun pada lingkungan jaringan. *Java 2* adalah generasi kedua dari *java platform*. (Rosa A.S dan M.Shalahuddin, 2014:103, *Rekaya Perangkat Lunak*. Jakarta: PT Elex Media Komputindo)

Java merupakan bahasa berorientasi objek untuk pengembangan aplikasi mandiri, aplikasi berbasis internet, aplikasi untuk perangkat cerdas yang dapat berkomunikasi lewat internet atau jaringan komunikasi. Melalui teknologi *java*, dimungkinkan perangkat *audio streo* dirumah terhubung jaringan komputer. *Java* tidak lagi hanya untuk membuat *applet* yang memerintah halaman *web* tapi *java* telah menjadi bahasa untuk pengembangan aplikasi skala *interprise* berbasis jaringan besar. (Bambang Haryanto, 2011:2, *Esensi-esensi Bahasa Pemrograman Java*. Yogyakarta: Andi)

Dari pengertian diatas maka dapat disimpulkan bahwa *Java* merupakan bahasa pemrograman berorientasi objek yang dapat digunakan untuk membuat dan menjalankan perangkat lunak pada komputer dan berbagai *platform*.

2.2.10. NetBeans

NetBeans awalnya dibangun pada tahun 1996 sebagai Xelfi (untuk pemrograman Delphi) oleh seorang mahasiswa dari Charles University di Paraguai.

Pada tahun 1997, Roman Stanek membangun sebuah perusahaan dan merilis versi komersial dari NetBeans hingga akhirnya dibeli oleh Sun Microsystems pada 1999. Hingga saat ini platform NetBeans telah banyak berkembang di bawah SunMicrosystem.

NetBeans merupakan platform framework dan IDE (*integrated development environment*) yang digunakan untuk pengembangan aplikasi *desktop* yang menggunakan Bahasa Java, dan beberapa bahasa lain, seperti Groovy, C, C++ dan banyak lagi.

NetBeans IDE dibangun menggunakan Bahasa Java dan dapat dijalankan pada Windows, OS X, Linux, Solaris dan sistem operasi lain yang mendukung JVM. IDE NetBeans merupakan alat pengembangan aplikasi yang terintegrasi. NetBeans IDE mendukung pengembangan Program yang menggunakan Bahasa Java dari semua versi (Java SE, Java ME, Java EE).

Platform Netbeans memperbolehkan pembangunan aplikasi dengan menggunakan modul-modul. Aplikasi yang dibangun menggunakan NetBeans dapat dikembangkan oleh pihak ketiga. Platform NetBeans merupakan platform yang dapat digunakan ulang (*reusable*) untuk mempermudah pembangunan program menggunakan Bahasa Java.

[Sumber:http://elib.unikom.ac.id/files/disk1/657/jbptunikompp-gdl-inechyntia-32807-10-unikom_i-i.pdf diakses pada 29 Oktober 2017]

2.2.11. IReport

IReport merupakan sebuah aplikasi *designer* yang digunakan untuk membuat laporan yang didalamnya terdapat *Library JasperReport*. *Library*

JasperReport sendiri merupakan JAR (*Java Library*) yang bersifat *open source* dan dirancang untuk menambahkan kemampuan pelaporan (*reporting capabilities*) pada aplikasi java.

JasperReport memiliki sejumlah fitur, antara lain :

1. *Layout* dan desain laporan yang fleksibel.
2. Dapat menampilkan laporan dalam bentuk teks maupun gambar (*chart*).
3. Dapat menghasilkan *report* dalam berbagai format : html, pdf, rtf, xls, csv.
4. Dapat menerima data dari berbagai sumber data : JDBC, Bean Collection, ResultSet, CSV, XML, Hibernate.

[Sumber:http://jasperforge.org/jaspersoft/opensource/business_intelligence/ireport diakses pada 29 Oktober 2017]

2.2.12. Aplikasi

Menurut Anisyah (2000), aplikasi adalah penerapan, penggunaan atau penambahan. Dari pengertian aplikasi, dapat disimpulkan bahwa aplikasi merupakan *software* yang berfungsi untuk melakukan berbagai bentuk pekerjaan atau tugas-tugas tertentu seperti penerapan, penggunaan dan penambahan data.

Beberapa aplikasi yang digabung bersama menjadi suatu paket kadang disebut sebagai suatu paket atau suite aplikasi (*application suite*). Contohnya adalah *Microsoft Office* dan *OpenOffice.org* yang menggabungkan suatu aplikasi pengolah kata, lembar kerja, serta beberapa aplikasi lainnya.

Aplikasi dalam suatu paket biasanya memiliki antarmuka pengguna yang memiliki kesamaan sehingga memudahkan pengguna untuk mempelajari dan menggunakan tiap aplikasi. Sering kali mereka memiliki kemampuan untuk saling

berinteraksi satu sama lain sehingga menguntungkan pengguna. Contohnya, suatu lembar kerja dapat dibenamkan dalam suatu dokumen pengolah kata walaupun dibuat pada aplikasi lembar kerja yang terpisah.

2.2.13. Black Box Testing

Dalam pengujian perangkat lunak ada dua yaitu *white box testing* dan *black box testing*. Dari kedua metode itu, pada skripsi dipilih menggunakan *black box testing* karena dianggap lebih tepat dibanding *white box testing*. Perangkat lunak memerlukan seperangkat tes untuk pencarian kesalahan fungsi-fungsi dalam aplikasi sehingga dalam hal ini *black box testing* lebih sesuai. Pengujian ini digunakan untuk mengetahui apakah fungsi-fungsi dalam perangkat lunak sudah sesuai dengan yang diharapkan.

Menurut Roger S. Pressman (2010), *black box testing* berfokus pada persyaratan fungsional perangkat lunak yang memungkinkan *engineer* untuk memperoleh *input* yang sepenuhnya akan melaksanakan persyaratan fungsional untuk sebuah program. *Black box testing* berusaha untuk menemukan kesalahan dalam kategori berikut:

1. Fungsi yang tidak benar atau fungsi yang hilang.
2. Kesalahan antarmuka.
3. Kesalahan dalam struktur data atau akses *database eksternal*.
4. Kesalahan kinerja.

Kesalahan inisialisasi dan pemutusan kesalahan.