

BAB II TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1. Tinjauan Pustaka

Pada penelitiannya mengembangkan “Sistem Informasi Pengelolaan Arsip Inaktif Berbasis Web Pada Badan Pengelolaan Keuangan Dan Aset Daerah Kabupaten Tanah Bumbu”. Permasalahan disana yaitu arsip Inaktif yang disimpan di gedung arsip hingga saat ini belum pernah diusulkan untuk dimusnahkan sehingga terjadi penumpukan berkas. Lokasi yang terpisah dari kantor utama mengakibatkan sulitnya pemantauan aktivitas kegiatan di gedung arsip. Tujuan pembuatan sistem informasi ini yakni untuk membantu pekerjaan pengelola arsip dalam mengelola arsip inaktif serta dapat menampilkan laporan kinerja. Sistem informasi ini dibuat menggunakan bahasa pemrograman *PHP* dengan menggunakan *framework CodeIgniter* dan diolah dengan menggunakan manajemen basis data *MySQL*. Penggunaan metode *waterfall* membuat perancangan sistem menjadi lebih mudah [7].

Penelitian dengan judul “Perancangan Sistem Informasi Arsip Salatiga Berbasis Web dengan *CodeIgniter* dan *Materialize*”. Perpustakaan dan Arsip Daerah Kota Salatiga menyediakan informasi mengenai koleksi arsip tentang Kota Salatiga dalam bentuk arsip foto, arsip tekstual, arsip buku dan arsip film dokumenter. Penyimpanan informasi mengenai koleksi arsip Salatiga belum terpusat dan masih terpisah sehingga proses pengelolaan dan penemuan kembali arsip mengalami kesulitan serta informasi yang dihasilkan kurang cepat dan tepat. Informasi arsip Salatiga hanya dapat diakses di ruang lingkup perpustakaan saja menyebabkan terbatasnya ruang akses bagi masyarakat untuk mendapatkan informasi mengenai arsip Salatiga. Berdasarkan masalah tersebut maka dilakukan perancangan sistem informasi berbasis web untuk arsip Salatiga. Metode perancangan sistem menggunakan *Prototype Model*. Sistem dibangun menggunakan *framework CodeIgniter* dengan pola arsitektur *MVC* serta *Materialize CSS* untuk membangun web *responsive*. Hasil yang diperoleh adalah sebuah sistem informasi berbasis web yang mampu mengelola arsip dan memberikan informasi mengenai koleksi arsip Salatiga kepada masyarakat [8].

Penelitian dengan judul “Sistem Informasi Arsip Surat (Sinau) Berbasis Web Pada Kantor Desa Karangsalam Kecamatan Baturraden”.

Pengelolaan surat yang menggunakan penyimpanan manual sering terjadi kesalahan dalam penyimpanan data dan pencarian surat. Sistem informasi arsip surat ini sangat diperlukan. Sistem informasi arsip surat ini bertujuan mengubah cara penyimpanan surat dengan cara mengurangi penggunaan kertas (*paperless*). Sistem informasi ini dibuat dengan menggunakan aplikasi *Notepad++*. Sedangkan bahasa pemrograman menggunakan *PHP*, *Xampp* digunakan untuk menjalankan web servernya, *MySQL* digunakan untuk pengelolaan *database*. Model alur hidup yang digunakan adalah *waterfall*, sedang pengujian unit menggunakan *blackbox* testing. Dengan adanya penyimpanan surat secara *paperless*, kesalahan yang terjadi selama ini dapat diminimalisasi. Tujuan pembuatan aplikasi ini untuk membuat suatu sistem informasi yang mengolah data arsip surat sehingga lebih mudah dalam penggunaannya [9].

Pada Penelitian ini penulis bermaksud membangun Sistem Informasi Pengelolaan Arsip Inaktif di Politeknik Negeri Cilacap berbasis web. Perbedaan dari penelitian sebelumnya adalah sistem ini dapat melakukan pengecekan retensi arsip, pencarian arsip, pengelolaan data peminjaman arsip dan pengembalian arsip, serta berita acara peminjaman dan pengembalian arsip, pengecekan atau *verifikasi* kelengkapan arsip dan pemberitahuan batas waktu peminjaman arsip, serta *verifikasi* dan berita pemindahan arsip. Sistem ini diharapkan mampu digunakan untuk petugas arsip dalam mengelola data arsip, dan mempermudah pengguna yang akan melakukan peminjaman dan pengembalian arsip, serta pencarian arsip. Dengan adanya aplikasi ini maka pengelolaan data arsip menjadi lebih mudah. Sistem ini berbasis web dibuat dengan menggunakan bahasa pemrograman *PHP* dengan framework *laravel* dan integrasi data dengan database *MySQL* bertujuan untuk memberi kemudahan bagi petugas arsip maupun pegawai yang membutuhkan data arsip tersebut, serta dapat mempersingkat waktu dalam hal mendapatkan data arsip.

2.2. Landasan Teori

2.2.1. Rekayasa Perangkat Lunak

Rekayasa perangkat lunak adalah Penerapan dan pemanfaatan prinsip-prinsip rekayasa untuk menghasilkan perangkat lunak yang ekonomis, andal dan bekerja secara efisien pada mesin yang nyata [10].

Tujuan Rekayasa Perangkat Lunak dapat dijelaskan sebagai berikut:

- a) Memperoleh biaya produksi perangkat lunak yang rendah.

- b) Menghasilkan perangkat lunak yang kinerjanya tinggi, andal dan tepatwaktu.
- c) Menghasilkan perangkat lunak yang dapat bekerja pada berbagai jenis platform.
- d) Menghasilkan perangkat lunak yang biaya perawatannya rendah.

Ruang lingkup Rekayasa Perangkat lunak meliputi :

- a) Software Requirement merupakan kegiatan yang dilakukan untuk mengidentifikasi dan menganalisis kebutuhan perangkat lunak. Hasil akhir tahapan ini adalah spesifikasi dan model perangkat lunak.
- b) Software Design adalah tahapan perancangan arsitektur, komponen, antar muka, dan karakteristik lain dari perangkat lunak
- c) Software Construction berhubungan dengan detail pengembangan perangkat lunak, termasuk algoritma, pengkodean, pengujian dan pencarian kesalahan
- d) Software Testing meliputi pengujian pada keseluruhan perilaku perangkat lunak
- e) Software Maintenance mencakup upaya-upaya perawatan ketika perangkat lunak telah dioperasikan
- f) Software Configuration management berhubungan dengan usaha perubahan konfigurasi perangkat lunak untuk memenuhi kebutuhan tertentu
- g) Software Engineering Management berkaitan dengan pengelolaan dan pengukuran RPL, termasuk perencanaan proyek perangkat lunak
- h) Software Engineering Tools and Methods mencakup kajian teoritis tentang alat bantu dan metode RPL
- i) Software Engineering Process berhubungan dengan definisi, implementasi pengukuran, pengelolaan, perubahan dan perbaikan proses RPL
- j) Software Quality menitik beratkan pada kualitas dan daur hidup perangkat lunak .

2.2.2. Sistem Informasi

Sistem informasi merupakan serangkaian komponen berupa prosedur, data dan teknologi yang digunakan untuk melakukan sebuah proses untuk menghasilkan informasi dalam pengambilan keputusan. Dua

kelompok pendekatan didalam mendefinisikan sistem yaitu, yang menekankan pada prosedurnya dan yang menekankan pada komponen. Pendekatan sistem yang lebih menekankan pada prosedur mendefinisikan sistem sebagai berikut. Suatu sistem adalah jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran yang tertentu. Prosedur didefinisikan oleh Richard F. Neuschel sebagai berikut. Suatu prosedur adalah suatu urutan-urutan operasi klerikal (tulis-menulis), biasanya melibatkan beberapa orang didalam satu atau lebih departemen, yang diterapkan untuk menjamin penanganan yang seragam dari transaksi bisnis yang terjadi. Pendekatan sistem yang lebih menekan pada elemennya mendefinisikan sistem sebagai berikut. Sistem adalah kumpulan elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu [11].

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

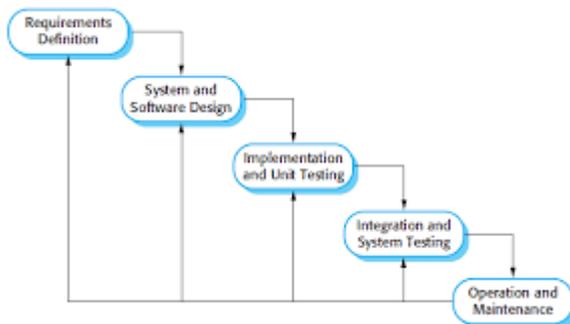
1. Komponen sistem (component)
Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang artinya saling bekerjasama membentuk suatu kesatuan.
2. Batasan sistem (boundary)
Batasan sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya.
3. Lingkungan luar sistem (environment)
Lingkungan luar dari suatu sistem adalah apapun diluar batas dari sistem yang mempengaruhi operasi.
4. Penghubung sistem (interface)
Penghubung merupakan media penghubung antara suatu subsistem dengan subsistem yang lainnya.
5. Masukan sistem (input)
Masukan sistem adalah energy yang dimasukkan ke dalam sistem. Sinyal input adalah energi yang diproses untuk mendapatkan keluaran dari sistem.
6. Pengolahan sistem (process)
Suatu sistem harus memiliki suatu bagian pengolah yang akan merubah masukan menjadi keluaran.
7. Keluaran sistem (output)
Keluaran sistem adalah hasil dari energi yang diolah dan di klarifikasikan menjadi keluaran yang bagus.

8. Sasaran sistem (objectives)

Suatu sistem mempunyai tujuan atau sasaran, jika sistem tidak memiliki sasaran maka sistem tidak ada. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuannya. Sasaran sangat berpengaruh pada masukan dan keluaran yang dihasilkan

2.2.3. Metode Pengembangan Sistem

Waterfall merupakan salah satu model dalam pengembangan perangkat lunak dimana semua proses kegiatan harus direncanakan dan dijadwalkan sebelum dikerjakan. Proses dari metode *Waterfall* model antara lain :



Gambar 2.1 Model *Waterfall*

a. *Requirements Definition*

Proses pengumpulan kebutuhan dilakukan secara intensif untuk menspesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh user. Spesifikasi kebutuhan perangkat lunak pada tahap ini perlu untuk didokumentasikan.

b. *System and Software Design*

Desain perangkat lunak adalah proses multi langkah yang fokus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak representasi antar muka, 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.

c. Implementation and Unit Testing

Desain harus ditranslasikan ke dalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain.

d. Intergration and System Testing

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.

e. Operation and Maintenance

Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirim 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.

2.2.4. Metode Black Box Testing

Pengujian black box merupakan sebuah metode pengujian software dimana penguji tidak mengetahui internal struktur, desain, dan implementasian dari suatu bagian yang sedang diuji. Pengujian sistem dilakukan untuk melihat apakah sistem yang dibangun sudah sesuai dengan keinginan dari pelanggan dan layak atau digunakan. Tujuan dari metode blackbox untuk mengetahui apakah sistem telah benar menampilkan kesalahan yang ada jika terjadi human error. Pada pengujian blackbox hanya mengambil hasil output melalui data uji dan mengecek fungsionalitas dari software. Penguji tidak perlu memiliki pengetahuan tentang Bahasa pemrograman hanya saja menguji pada tampilan luar (interface).

Pengujian menggunakan masukan data acak bertujuan untuk memastikan sistem menolak untuk menyimpan data masukan pada database, sehingga sistem dikatakan layak untuk digunakan. Berikut cara pengujian setiap tipe dari metode Black Box :

1. Equivalence Partitioning Pengujian ini dilakukan pada form yang sudah ada pada sistem informasi surat keluar masuk dengan memasukkan data yang tidak sesuai dengan tipe data atau memasukkan data acak.
2. Boundary Value Analysis Pengujian ini untuk memastikan bahwa masukkan data yang melebihi batas yang sudah ditentukan tidak dapat tersimpan dengan baik pada database.
3. Comparison Testing Membandingkan tampilan interfaces sistem pada web browser yang berbeda
4. Sample Testing Pengujian ini untuk memastikan nilai yang terpilih dapat menghasilkan data yang baik dan sesuai dengan data masukkan dari user.
5. Robustness Testing Penguji akan memasukkan data acak untuk membuktikan bahwa tidak ada kesalahan jika masukan tidak valid.
6. Behavior Testing Pengujian ini dilakukan dengan cara membuat data baru secara berkali – kali untuk menghindari data stack.
6. Performance Testing Pengujian ini mengevaluasi kemampuan program untuk beroperasi dengan benar dipandang dari aliran pemakaian memori,
7. Requirement Testing Tipe ini hanya melihat spesifikasi kebutuhan dari sistem mulai dari sistem pembuatan sampai pengujian.
8. Endurance Testing Tipe ini untuk memastikan apakah hasil operasi matematika pada sistem ini benar atau salah.
9. Cause – Effect Relationship Testing Pengujian yang melibatkan kondisi input dan aliran data mulai dari Input, View, Update, Delete dan Search.

2.2.5. Pengujian Usability

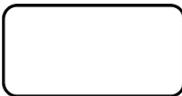
Usability adalah sebuah analisa kualitatif yang menentukan seberapa mudah user menggunakan antarmuka suatu aplikasi, aplikasi dapat dikatakan usable apabila fungsinya dapat dijalankan secara efektif, efisien, dan memuaskan. Pengujian dalam usability dapat dilakukan dengan melibatkan pengguna atau tidak sama sekali. Pengujian dengan melibatkan pengguna menggunakan sistem serta permasalahan yang

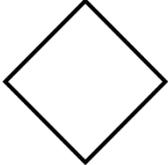
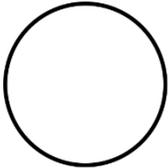
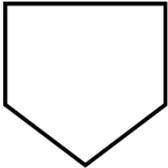
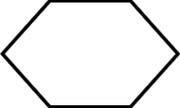
dihadapi. Pengujian dilakukan dengan cara menggunakan kuisisioner yang dapat mengolah data yang berhubungan dengan learnability, flexibility, effectiveness, dan attitude dalam penggunaan aplikasi tersebut. Metode Kuisisioner dilakukan dengan membagikan kuisisioner untuk mengukur kepuasan pengguna terhadap aplikasi. Salah satu metode yang digunakan adalah skala likert yang digunakan dalam merancang skala pengukuran pada penelitian perilaku. Skala likert memiliki empat atau lebih butir pertanyaan yang dikombinasikan sehingga membentuk sebuah skor atau nilai yang mempresentasikan sifat individu. Aspek Learnability merupakan aspek yang berhubungan dengan kemudahan pengguna untuk menyelesaikan permasalahan dasar yang dihadapi serta mudah dimengerti. Aspek Flexibility berhubungan dengan ketersediaan fitur yang ada pada sistem bagi pengguna. Aspek Effectiveness berkaitan dengan keberhasilan untuk mencapai tujuan dalam penggunaan aplikasi, dan Aspek Attitude berhubungan dengan tingkat kepuasan pengguna [12].

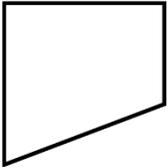
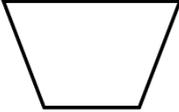
2.2.6. *Flowchart*

Flowchart atau bagan alir adalah penggambaran secara grafik dari langkah-langkah dan urutan-prosedur dari suatu program. Simbol *flowchart* dipakai sebagai alat untuk membantu dalam mendefinisikan sebuah proses. Proses dalam *flowchart* digambarkan dari halaman atas ke bawah dan dari kiri ke kanan. Aktifitas yang digambarkan kemudian didefinisikan dengan jelas sehingga dimengerti oleh pembacanya [13].

Tabel 2.1 Simbol Flowchart

No	Gambar Simbol	Nama	Keterangan
1.		Terminal	Menyatakan permulaan atau akhir suatu program.
2.		Arus/ <i>Flow</i>	Untuk menyatakan jalannya arus suatu proses.

3.		Proses	Menunjukkan sebuah kegiatan atau proses yang dilakukan sistem.
4.		<i>Decision /Logika</i>	Menunjukkan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan (ya/tidak).
5.		<i>Input output</i>	Menyatakan proses <i>input/output</i> tanpa tergantung dengan jenis peralatannya.
6.		<i>Connector</i>	Menyatakan sambungan dari suatu proses ke proses lainnya dalam halaman/lembar yang sama.
7.		<i>Offline Connector</i>	Menyatakan sambungan dari suatu proses ke proses lainnya dalam halaman/lembar yang berbeda.
8.		<i>Predefined Proses</i>	Menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberi harga awal.

9.		Manual <i>input</i>	Memasukkan data secara manual dengan menggunakan <i>online keyboard</i> .
10.		Manual	Menyatakan suatu Tindakan (proses) yang tidak dilakukan oleh komputer (manual).
11.		<i>Document</i>	Mencetak laporan ke printer.

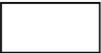
2.2.7. UML (*Unified Modeling Language*)

Unified Modeling Language merupakan salah satu metode pemodelan visual yang digunakan dalam perancangan dan pembuatan sebuah *software* yang berorientasikan pada objek. *UML* merupakan sebuah standar penulisan atau semacam *blueprint* dimana didalamnya termasuk sebuah bisnis proses, penulisan kelas-kelas dalam sebuah bahasa yang spesifik [14].

A. *Use Case Diagram*

Use case digunakan untuk menggambarkan fungsi dasar dari sebuah sistem informasi. *Use case* mendeskripsikan cara sistem bisnis berinteraksi dengan lingkungannya. Berikut adalah elemen-elemen dari *use case diagram* :

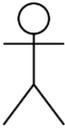
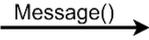
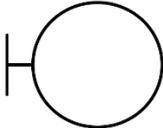
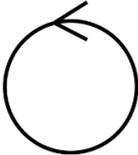
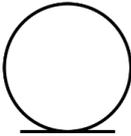
Tabel 2.2 Simbol *Use Case Diagram*

No	Simbol	Fungsi
1.		Actor/role adalah orang atau sistem lain yang berinteraksi dengan sistem saat ini.
2.		Memasukkan satu use case dalam use case lainnya. Perilaku (behavior) yang harus terpenuhi agar sebuah event dapat terjadi, di mana kondisi ini sebuah use case adalah bagian dari use case lainnya. Tanda panah mengarah dari base use case (pusat) menuju ke use case yang di-includ.
3		Menggambarkan navigasi antar class, berupa banyak objek lain yang berhubungan dengan satu objek, dan apakah suatu class menjadi bagian dari class lainnya.
4.		Berisi nama dari sistem yang diletakkan di dalam atau di bagian atas boundary. Mewakili ruang lingkup sistem. Actor berada di luar ruang lingkup sistem.
5.		Use case adalah bagian utama dari fungsionalitas sistem. Bisa extend (memperluas) use case lainnya. Ditempatkan di dalam system boundary (batasan sistem). Dilabeli dengan kata kerja – frase kata benda.

B. Sequence Diagram

Sequence *diagram* menunjukkan pesan yang lewat di antara objek untuk use case tertentu dari waktu ke waktu. *Sequence diagram* mengilustrasikan objek-objek yang berpartisipasi di dalam suatu use case. Berikut adalah elemen-elemen dari *sequence diagram*.

Tabel 2.2 Simbol *Sequence Diagram*

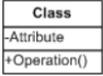
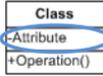
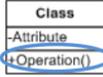
No	Simbol	Fungsi
1.		Merupakan persegi panjang sempit panjang yang ditempatkan di atas lifeline. Menunjukkan kapan suatu objek mengirim atau menerima pesan.
2..		Adalah orang atau sistem yang memperoleh manfaat dari dan berada di luar sistem. Berpartisipasi dalam suatu urutan dengan mengirim dan / atau menerima pesan. Ditempatkan di bagian atas diagram.
3.		Spesifikasi dari komunikasi antar objek yang memuat informasi – informasi tentang aktivitas yang terjadi.
4.		Menggambarkan tampilan program.
5.		Menggambarkan <i>controller</i> .
6.		Digunakan untuk menggambarkan hubungan kegiatan yang akan dilakukan.

C. *Class Diagram*

Class diagram adalah model *statis* yang menunjukkan kelas dan hubungan di antara kelas yang tetap konstan dalam sistem dari waktu ke

waktu. *Class diagram* menggambarkan kelas, yang meliputi perilaku dan keadaan, dengan hubungan antar kelas. Berikut adalah elemen-elemen dari *class diagram*:

Tabel 2.3 Simbol *Class Diagram*

No	Simbol	Fungsi
1.		<i>Class</i> , mewakili orang, tempat, atau hal-hal yang dibutuhkan sistem untuk menangkap dan menyimpan informasi. Memiliki nama yang diketik dengan huruf tebal dan berpusat di bagian atas kotak. Memiliki daftar atribut di kotak tengahnya. Memiliki daftar operasi di kotak bawahnya. Tidak secara eksplisit menunjukkan operasi yang tersedia untuk semua kelas..
2.		<i>Attribute</i> , merupakan properti yang menggambarkan keadaan suatu objek. Dapat diturunkan dari atribut lain, ditampilkan dengan menempatkan garis miring sebelum nama atribut.
3		<i>Operation</i> , mewakili tindakan atau fungsi yang dapat dilakukan oleh kelas. Dapat diklasifikasikan sebagai konstruktor, permintaan, atau operasi pembaruan. Termasuk tanda kurung yang mungkin berisi parameter atau informasi yang diperlukan untuk melakukan operasi.

2.2.8. Basis Data (*Database*)

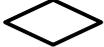
Basis data adalah suatu kumpulan data terhubung yang disimpan secara bersama-sama pada suatu media, tanpa adanya suatu kerangkapan data, sehingga mudah untuk digunakan kembali, dapat digunakan oleh satu atau lebih program aplikasi secara optimal, data disimpan tanpa mengalami ketergantungan pada program yang akan menggunakannya, data disimpan sedemikian rupa sehingga apabila ada penambahan, pengambian dan modifikasi data dapat dilakukan dengan mudah dan terkontrol [15].

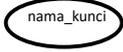
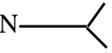
2.2.9. Entity Relationship Diagram (ERD)

ERD merupakan suatu cara untuk menjelaskan kepada para pemakai tentang hubungan antar data dala basis data secara logic dengan presepsi bahwa real world terdiri dari objek-objek dasar yang saling berhubungan dengan cara memvisualisasikan ke dalam bentuk simbol-simbol grafis . Pada dasarnya ada tiga simbol yang digunakan, yaitu [15]:

1. Entitiy
Entity merupakan objek yang mewakili sesuatu yang nyata dan dapat dibedakan dari sesuatu yang lain. Simbol dari entity ini biasanya digambarkan dengan persegi panjang.
2. Atribut
Setiap entitas pasti mempunyai atribut yang berfungsi untuk mendeskripsikan karakteristik dari entitas tersebut. Isi dari atribut mempunyai sesuatu yang dapat mengidentifikasi isi elemen satu dengan yang lain. Gambar atribut biasanya digambarkan dengan elips.
3. Hubungan/relasi
Hubungan antara sejumlah entitas yang berasal dari himpunan entitas yang berbeda. Berikut simbol-simbol yang digunakan dalam ERD , sesuai pada Tabel 2.6.

Tabel 2. 1 Simbol-Simbol ERD (Entity Relationship Diagram)

No.	Simbol	Nama	Keterangan
1.		Entity	Data inti yang akan disimpan.
2.		Relasi	Relasi yang menghubungkan antar entitas.
3.		Atribut	Kolom data yang butuh disimpan dalam suatu entitas.

4.		Atribut kunci primer	Kolom data yang butuh disimpan dalam suatu entitas data digunakan sebagai kunci akses record yang diinginkan.
5.		Atribut multivalai	Kolom data yang butuh disimpan dalam suatu entitas yang dapat memiliki nilai lebih dari satu.
6.		Asosiasi	Penghubung antar relasi dan entitas dimana di kedua ujungnya memiliki multiplicity kemungkinan jumlah pemakaian.

Relasi antar tabel sebagai berikut :

- a) Hubungan One-to-One adalah dengan menggabungkan ke entitas yang membutuhkan atau ke entitas yang mempunyai jumlah atribut yang lebih sedikit.
- b) Hubungan One-to-Many adalah dengan menggabungkan ke arah entitas berderajat banyak (many), sehingga penggabungannya tidak perlu melihat jumlah atribut.
- c) Hubungan Many-to-Many adalah suatu relasi atau hubungan akan membentuk relasi atau tabel tersendiri atau relasi baru.

2.2.10.Sistem Berkas

Sistem berkas merupakan sistem penyimpanan, pengorganisasian, pengelolaan data pada alat penyimpan eksternal, dengan menggunakan teknik organisasi data tertentu. Konsep sistem berkas menghasilkan informasi yang dibutuhkan (sesuai dengan requirement user) dalam waktu yang masih dapat diterima oleh user. Sistem berkas mengacu pada Waktu komputasi pada volume data yang diproses/diolah. Pengelolaan data dalam jumlah besar membutuhkan effort lebih, baik dalam hal storage device dan pengelolaannya (cara penyimpanan dan pengaksesannya) [16]
Data harus :

1. Dapat diakses oleh multi user
2. Selalu tersedia setiap saat dibutuhkan untuk pemrosesan
3. Waktu pengaksesan relatif cepat
4. tersimpan dalam media penyimpanan sekunder.

Berdasarkan jenisnya sistem berkasada 2 jenis:

a. Secara Logik

Penggambaran data di level konseptual, misal penggambaran data dengan metode E-R, model objek, model semantik, dan lain-lain.

b. Secara Fisik

Penggambaran data di level fisik, bagaimana data direpresentasikan dalam media penyimpanan.

Berdasarkan Klasifikasi Data sistem berkas dibagi menjadi beberapa bagian :

1. Data Tetap

Kelompok data yang tidak mengalami perubahan, paling tidak dalam kurun waktu yang lama contohnya data pribadi mahasiswa.

2. Data Tidak Tetap

Kelompok data yang secara rutin mengalami perubahan, contohnya data rencana studi mahasiswa.

3. Data Yang bertambah menurut waktu

Kelompok data ini biasanya merupakan data akumulasi dari kelompok data tetap dan data tak tetap contohnya data transkrip.

2.2.11.Arsip

Arsip dinamis adalah arsip yang masih dibutuhkan secara langsung dalam perencanaan, pelaksanaan, penyelenggaraan kehidupan kebangsaan secara umum atau arsip yang digunakan secara langsung dalam administrasi negara. Arsip dinamis dilihat dari kegunaannya dibedakan atas arsip aktif dan arsip inaktif sebagai berikut [17]:

1. Arsip aktif

Arsip aktif merupakan arsip yang secara langsung dan terus menerus dibutuhkan dan digunakan dalam penyelenggaraan administrasi sehari-hari dan masih dikelola oleh unit pengelolaan.

2. Arsip inaktif

Arsip inaktif Adalah arsip yang tidak secara langsung dan tidak menerus diperlukan dan digunakan dalam penyelenggaraan administrasi sehari-hari serta dikelola oleh pusat arsip.

Setelah arsip tercipta dan telah digunakan, arsip perlu disimpan dengan benar. Pada dasarnya, sistem penyimpanan arsip dapat dilakukan berdasarkan urutan alfabet dan urutan angka. Sistem penyimpanan alfabet meliputi sistem nama, sistem geografi dan sistem subjek. Sedangkan sistem penyimpanan berdasarkan urutan angka termasuk sistem numerik, sistem kronologis dan sistem subjek numerik. Proses penggunaan arsip ini dilakukan melalui tahap peminjaman arsip. Peminjaman arsip adalah keluarnya arsip dalam tempat penyimpanan karena dipinjam oleh atasan, teman seunit kerja, ataupun oleh kolega sekerja dari unit kerja lain dalam organisasi. Dalam kegiatan peminjaman arsip harus memperhatikan hal-hal sebagai berikut:

1. Peminjam arsip diharuskan mengisi daftar/formulir peminjaman arsip.
2. Menaruh kartu substitusi/kartu bukti pinjam arsip (out guide/out folder) atau lembar peminjaman arsip dua (hijau) di tempat arsip tersebut diambil,
3. atau disimpan dalam kotak peminjaman sesuai dengan tanggal pengambilannya.
4. Hanya sekretaris dan petugas yang disertai tugas untuk dapat mengambil arsip.
5. Adanya tindak lanjut terhadap arsip-arsip yang dipinjam.

~ Halaman ini sengaja dikosongkan ~