

BAB 2

DASAR TEORI

2.1 Tinjauan Pustaka

Sebuah penelitian yang telah dilakukan oleh Ade dan Yuli pada tahun 2019 yang bertujuan untuk membangun sebuah sistem yang dapat membantu staf Kantor Desa Karangrau dalam mengelola arsip surat masuk dan keluar [4]. Sistem dibangun dengan menggunakan metode pengembangan SDLC model *waterfall*, *PHP* dan *Javascript* sebagai bahasa pemrograman dan *MySQL* sebagai basis data. Penelitian tersebut menghasilkan sebuah sistem yang dapat mengelola surat masuk dan surat keluar sesuai alur yang ditetapkan.

Penelitian lainnya telah dilakukan Saifudin dan Setiaji pada tahun 2019 [5]. Penelitian tersebut bertujuan untuk membangun sebuah sistem yang dapat menyelesaikan permasalahan kearsipan surat di Kantor Desa Karangsalam Kecamatan Baturaden. Sistem yang berjalan saat itu masih konvensional dan sering terjadi kesalahan dalam penyimpanan dan pencarian surat. Penelitian tersebut menggunakan model pengembangan *waterfall*, dan *black box* sebagai metode pengujian aplikasi. Penelitian tersebut berhasil meminimalisir kesalahan pengarsipan dan pencarian surat.

Penelitian sebelumnya juga telah dilakukan oleh Fadly pada tahun 2019 [6]. Penelitian tersebut bertujuan untuk membangun sistem yang dapat membantu perusahaan Sourci Indonesia dalam mengelola arsip dan menghindari duplikasi pengelolaan arsip. Penelitian ini menggunakan metodologi perancangan Scrum. Hasil dari penelitian tersebut adalah merancang sistem informasi pengarsipan yang dapat meminimalisir duplikasi arsip dan memudahkan dalam mengelola arsip secara efektif dan efisien.

Pada penelitian ini, penulis bermaksud membangun sistem yang dapat membantu BEM PNC dalam mengarsipkan dokumen-dokumen yang sangat mungkin dibutuhkan sebagai referensi dalam kegiatan maupun kepengurusan berikutnya. Metode pengembangan yang penulis

gunakan adalah model *waterfall*, dan *Black Box Testing* sebagai metode pengujian aplikasinya.

Tabel 2.1 Tabel Perbandingan Penelitian

No	Judul Penelitian	Tujuan	Hasil
1	Rancang Bangun Sistem Pengelolaan Arsip Surat Berbasis <i>Web</i> Menggunakan Metode <i>Waterfall</i>	Meningkatkan efektivitas dan efisiensi waktu proses pencarian data-data surat dan memperbaiki manajemen pengarsipan yang sedang berjalan	Sistem informasi yang dibuat dapat mengelola surat masuk dan keluar sesuai alur yang ditetapkan dan dapat menyelesaikan masalah yang ada saat ini.
2	Sistem Informasi Arsip (SINAU) Berbasis Web Pada Kantor Desa Karangsalam Kecamatan Baturaden	Mengubah cara penyimpanan surat dengan cara mengurangi penggunaan kertas (<i>paperless</i>) dan meningkatkan efisiensi waktu pencarian data surat	Sistem informasi berbasis <i>website</i> yang dapat mengelola arsip surat masuk dan surat keluar
3	Perancangan Sistem Informasi Kearsipan Di Sourci Indonesia	Mengurangi duplikasi pekerjaan dan meningkatkan pengelolaan arsip	Sistem informasi kearsipan yang dibuat dapat meminimalisir duplikasi dan mempermudah pengguna dalam mengelola kearsipan dengan lebih efektif dan efisien

2.2 Dasar Teori

2.2.1 Sistem Informasi

Sebuah Sistem Informasi (SI) bisa terdiri atas kombinasi yang terorganisasi apapun dari manusia, perangkat keras, perangkat lunak, jaringan komunikasi, sumber data, dan kebijakan serta prosedur yang terorganisasi yang menyimpan, mengambil, mengubah, dan memisahkan informasi dalam sebuah organisasi. Manusia bergantung pada sistem informasi modern untuk berkomunikasi dengan yang lainnya menggunakan berbagai perangkat fisik (perangkat keras), instruksi dan prosedur pemrosesan informasi (perangkat lunak), saluran komunikasi (jaringan), dan data yang tersimpan (sumber data). Walaupun sistem informasi saat ini umumnya dianggap sebagai sesuatu yang dilakukan dengan menggunakan komputer, kita sudah menggunakan sistem informasi sejak awalnya sebuah peradaban. Seperti contoh sistem informasi yang dimaksud adalah sinyal asap untuk berkomunikasi yang memiliki makna semacam instruksi yang ditransmisikan berdasarkan pola dari asap, katalog kartu di perpustakaan yang dirancang untuk menyimpan data tentang buku yang diletakkan secara teratur sehingga pembaca dapat menemukan lokasi dari sebuah buku tertentu berdasarkan judul, nama pengarang, atau berbagai pendekatan lainnya, dan mesin penghitung uang masuk di restoran cepat saji yang dapat melacak produk terjual, waktu penjualan, tingkat persediaan, dan jumlah uang dalam mesin kasir [7].

A. Fundamental Sistem Informasi

Ketika terdapat begitu banyak aplikasi perangkat lunak, terdapat tiga fundamental untuk semua aplikasi bisnis dari teknologi informasi, yaitu mendukung proses dan kegiatan operasi bisnis, mendukung pengambilan keputusan oleh karyawan dan manajer, dan mendukung strategi bagi keuntungan kompetitif. Kapanpun, sistem informasi yang dirancang untuk mendukung proses dan kegiatan operasi bisnis mungkin juga menyediakan data untuk, atau menerima data dari, sistem yang berfokus pada pengambilan keputusan bisnis atau mencapai keuntungan kompetitif. Hal yang sama juga berlaku untuk dua peranan fundamental dari SI yang lainnya. Organisasi yang ada saat ini bekerja keras secara konstan untuk mencapai integrasi sistem mereka yang memberikan

fleksibilitas dan dukungan bisnis yang lebih besar lagi dibandingkan yang bisa diberikan oleh sistem individual [7].

B. Definisi Sistem

Sistem didefinisikan sebagai seperangkat komponen yang saling terhubung, dengan sebuah batasan yang jelas, bekerja bersama untuk mencapai sebuah tujuan yang sama dengan menerima masukan dan menghasilkan keluaran dalam sebuah proses transformasi yang terorganisasi. Sistem konsep akan menjadi lebih berguna dengan menyertakan dua elemen tambahan yakni umpan balik dan kendali. Sebuah sistem dengan fungsi umpan balik dan kendali terkadang disebut dengan sistem sibernetik (*cybernetic*), yaitu sebuah sistem yang melakukan pengawasan dan pengaturan sendiri [7]. Berikut ini konsep sistem:

- a. Masukan, melibatkan penangkapan dan perakitan elemen yang masuk kedalam sistem untuk diproses. Sebagai contohnya adalah bahan baku, energi, data, dan usaha manusia yang harus diamankan dan diorganisasi untuk pengolahan,
- b. Pengolahan, melibatkan proses transformasi yang mengubah masukan menjadi keluaran. Contohnya proses manufaktur dan proses bernapas manusia,
- c. Keluaran, melibatkan pemindahan elemen yang telah dihasilkan oleh sebuah proses transformasi ke tujuan akhir mereka. Sebagai contohnya adalah produk jadi, jasa manusia, dan manajemen informasi harus dikirimkan kepada pengguna manusia.
- d. Umpan balik, yaitu mengembalikan data tentang kinerja dari sebuah sistem. Sebagai contoh, data mengenai kecepatan, ketinggian, dan arah dari pesawat terbang merupakan umpan balik bagi pilot pesawat terbang atau pilot otomatis.
- e. Kendali, melibatkan pengawasan dan pengevaluasian umpan balik untuk menentukan apakah sebuah sistem berusaha mencapai tujuannya. Fungsi kendali kemudian membuat penyesuaian yang diperlukan bagi sebuah masukan sistem dan komponen pemrosesan untuk memastikan bahwa hal itu menghasilkan keluaran yang sesuai. Sebagai contoh, seorang pilot atau pilot otomatis pesawat membuat berita acara

penyesuaian setelah mengevaluasi umpan balik dari instrumen untuk memastikan pesawat berada pada jalur yang diinginkan pilot.

C. Definisi Informasi

Informasi adalah suatu penambahan dalam ilmu pengetahuan yang menyumbangkan kepada konsep kerangka kerja yang umum dan fakta-fakta yang diketahui. Informasi bertumpu pada konteks dan pengetahuan umum si penerima untuk kepentingannya. Informasi yang dihasilkan dari proses pengolahan data dijadikan sebagai sumber daya. Secara tradisional sumber daya terdiri dari manusia, keuangan dan sumber daya material. Setelah dua dasawarsa, informasi diakui sebagai salah satu sumber daya yang sangat penting bagi manajemen dan bagi yang lainnya, didalam keadaan tertentu, informasi merupakan pengganti biaya yang efektif. Informasi dalam hal ini tidak berkaitan dengan penggunaan uang tetapi berkaitan dengan waktu, tanpa pamrih, dan dalam cara yang berbeda dengan sumber daya lainnya. Data yang diolah menjadi informasi harus segar (tidak kedaluwarsa), informasi harus tersedia setiap saat untuk suatu keputusan yang berguna dan nilai suatu informasi menurun sejalan dengan waktu berlalu. Anda dapat mempunyai segudang informasi, tetapi sistem informasi manajemen melindungi pengguna dari informasi yang berlebihan [7].

2.2.2 Pemrograman Berorientasi Objek

Pemrograman berorientasi objek (PBO) adalah paradigma pemrograman yang merepresentasikan objek-objek yang ada didunia nyata [8]. Ada empat konsep *object-oriented*, antara lain :

- a. Abstraksi (*Abstraction*)
Abstraksi bertujuan untuk menyaring properti dan operasi pada suatu objek, sehingga hanya tinggal properti dan operasi yang dibutuhkan saja.
- b. Pewarisan (*Inheritance*)
Pewarisan adalah kondisi dimana sebagai hasil dari instansiasi sebuah kelas, sebuah objek memiliki semua karakteristik dari kelasnya. Dengan demikian apapun atribut dan operasi dari kelas akan dimiliki pula oleh semua objek yang diwariskan dari kelas tersebut.
- c. Polimorfisme (*Polymorphism*)

Polimorfisme adalah konsep yang sangat handal bagi pengembang perangkat lunak untuk pemisahan secara jelas diantara sub-sistem yang berbeda. Dengan demikian sebuah sistem akan bisa dimodifikasi secara mudah karena hanya dibutuhkan antarmuka antar kelas. Pada intinya, sebuah kelas memiliki nama operasi yang sama namun hasil akhirnya akan berbeda tergantung parameter dari operasi tersebut.

d. Enkapsulasi (*Encapsulation*)

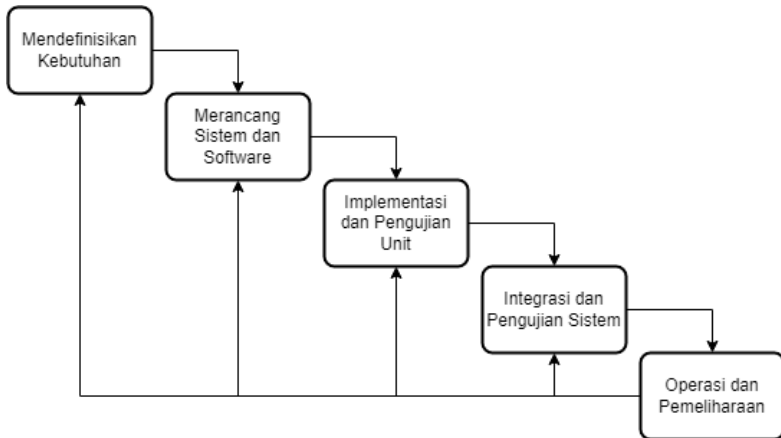
Enkapsulasi merupakan penyembunyian informasi, konsep ini sebenarnya lebih didasari pada fakta yang didunia nyata bahwa tidak semua hal perlu diperlihatkan. Contoh, untuk memperbesar volume suara pada TV, pengguna hanya perlu menekan satu tombol tertentu saja tanpa perlu tahu bagaimana proses dibelakang itu semua sehingga suara TV sesuai dengan harapan pengguna.

2.2.3 Rekayasa Perangkat Lunak

Rekayasa perangkat lunak (RPL) merupakan pembangunan dengan menggunakan prinsip atau konsep rekayasa dengan tujuan menghasilkan perangkat lunak yang bernilai ekonomi yang dipercaya dan bekerja secara efisien menggunakan mesin [9]. RPL lebih fokus pada praktik pengembangan perangkat lunak dan mengirimkan perangkat lunak yang bermanfaat kepada pengguna. Adapun ilmu komputer lebih fokus pada teori dan konsep dasar perangkat komputer. Rekayasa perangkat lunak lebih fokus pada bagaimana membuat perangkat lunak yang memenuhi kriteria berikut :

- 1) *Maintainability*, perangkat lunak dapat terus dipelihara setelah perangkat lunak selesai dibuat seiring berkembangnya teknologi dan lingkungan,
- 2) *Dependability robust*, perangkat lunak dapat diandalkan dengan proses bisnis yang dijalankan dan perubahan yang terjadi,
- 3) *Efficient*, perangkat lunak dinilai efisien dari segi sumber daya dan penggunaan, dan
- 4) *Usability*, kemampuan perangkat lunak untuk dipakai sesuai dengan kebutuhan.

A. Metode Pengembangan Sistem



Gambar 2.1 SDLC Model *Waterfall*

Software Development Life Cycle (SDLC) adalah proses mengembangkan atau mengubah suatu sistem perangkat lunak dengan menggunakan model-model dan metodologi yang digunakan orang untuk mengembangkan sistem-sistem perangkat lunak sebelumnya [9]. Model pertama yang diterbitkan dari proses pengembangan perangkat lunak berasal dari proses rekayasa sistem yang lebih umum. Model *waterfall* adalah contoh proses yang digerakkan oleh rencana pada prinsipnya, pengembang harus merencanakan dan menjadwalkan semua aktivitas proses sebelum mulai mengerjakannya. Tahapan utama model *waterfall* secara langsung mencerminkan kegiatan pengembangan mendasar [10]:

a) Mendefinisikan Kebutuhan

Layanan sistem, batasan, dan tujuan ditetapkan melalui konsultasi dengan pengguna sistem. Mereka kemudian didefinisikan secara rinci dan berfungsi sebagai spesifikasi sistem.

b) Merancang Sistem dan *Software*

Proses desain sistem mengalokasikan persyaratan untuk sistem *hardware* atau *software* dengan membangun arsitektur

sistem secara keseluruhan. Desain *software* melibatkan pengidentifikasian dan penggambaran abstraksi sistem *software* yang mendasar dan hubungannya.

c) Implementasi dan Pengujian Unit

Selama tahap ini, desain perangkat lunak direalisasikan sebagai satu set program atau unit program. Pengujian unit melibatkan verifikasi bahwa setiap unit memenuhi spesifikasinya.

d) Integrasi dan Pengujian Sistem

Unit program atau program individual diintegrasikan dan diuji sebagai sistem yang lengkap untuk memastikan bahwa persyaratan *software* telah dipenuhi. Setelah pengujian, sistem *software* dikirimkan ke pelanggan.

e) Operasi dan Pemeliharaan

Biasanya, ini adalah fase siklus hidup terpanjang. Sistem dipasang dan digunakan secara praktis. Pemeliharaan melibatkan koreksi kesalahan yang tidak ditemukan pada tahap awal siklus hidup, meningkatkan implementasi unit sistem dan meningkatkan layanan sistem saat persyaratan baru ditemukan.

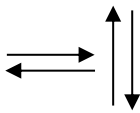


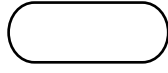


B. Metode Pengujian Sistem

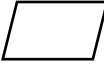





Sistem dalam pengembangannya harus diuji karena proses analisis, perancangan dan juga pemrogramannya yang tidak bebas dari kesalahan. Metode ini diharapkan memiliki mekanisme untuk menentukan data uji yang dapat menguji *software* secara lengkap dan memiliki kemungkinan tinggi untuk menjumpai kesalahan. Dalam pengujian, diperlukan strategi pengujian agar pengujian dapat terurut. Strategi ini secara terurut adalah pengujian unit, pengujian integritas dan pengujian sistem, perangkat lunak dapat diuji dengan dua cara, salah satunya *Black Box Testing* yaitu pengujian dilakukan dengan mengeksekusi data uji dan mengecek apakah fungsional *software* tanpa menguji desain dan kode program dari sistem. Pengujian ini untuk mengetahui fungsi-fungsi, masukkan dan keluaran dari *software* sesuai atau tidak dengan spesifikasi yang dibutuhkan [11].

C. Flowchart

Flowchart adalah diagram yang menampilkan langkah-langkah dan keputusan untuk melakukan sebuah proses dari suatu program. Setiap langkah digambarkan dalam bentuk diagram dan dihubungkan dengan garis atau arah panah [12]. Berikut adalah simbol-simbol yang sering digunakan dalam proses pembuatan *flowchart* :

Tabel 2.2 Simbol *Flowchart*

No.	Simbol	Nama	Keterangan
1.		<i>Flow</i>	Simbol yang digunakan untuk menggabungkan antara simbol yang satu dengan simbol yang lain.
2.		<i>On-Page Reference</i>	Simbol untuk keluar-masuk atau penyambungan proses dalam lembar kerja yang sama.
3.		<i>Off-Page Reference</i>	Simbol untuk keluar-masuk atau penyambungan proses dalam lembar kerja yang berbeda.
4.		<i>Terminator</i>	Simbol yang menyatakan awal atau akhir suatu program.
5.		<i>Process</i>	Simbol yang menyatakan suatu proses yang dilakukan komputer.
6.		<i>Decision</i>	Simbol yang menunjukkan kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban, yaitu ya dan tidak.

7.		<i>Input/Output</i>	Simbol yang menyatakan proses <i>input</i> atau <i>output</i> tanpa tergantung peralatan.
8.		<i>Manual Operation</i>	Simbol yang menyatakan suatu proses yang tidak dilakukan oleh komputer.
9.		<i>Document</i>	Simbol yang menyatakan bahwa <i>input</i> berasal dari dokumen dalam bentuk fisik, atau <i>output</i> yang perlu dicetak.
10.		<i>Predefine Process</i>	Simbol untuk pelaksanaan suatu bagian (sub-program) atau prosedur.
11.		<i>Display</i>	Simbol untuk menyatakan peralatan <i>output</i> yang digunakan.
12.		<i>Preparation</i>	Simbol yang menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberikan nilai awal.

D. *Unified Modelling Language (UML)*

Pemodelan adalah gambaran dari realita yang simpel dan dituangkan dalam bentuk pemetaan dengan aturan tertentu. Pemodelan dapat menggunakan bentuk yang sama dengan realitas misalnya jika seorang arsitek ingin memodelkan sebuah gedung yang akan dibangun maka arsitek tersebut akan memodelkannya dengan membuat sebuah maket (tiruan) arsitektur gedung yang akan dibangun dimana maket itu akan dibuat semirip mungkin dengan desain gedung yang akan dibangun agar arsitektur gedung yang diinginkan dapat terlihat [9].




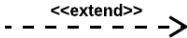
Unified Modelling Language (UML) adalah salah standar bahasa yang banyak digunakan didunia industri untuk mendefinisikan

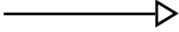
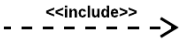
kebutuhan, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek. UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak. UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung [10]. Berikut ini dua diagram yang terdapat pada UML :

a) *Use Case Diagram*

Use case diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada didalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu [9]. Berikut ini merupakan komponen-komponen *use case diagram* :

Tabel 2.3 Simbol *Use Case Diagram*




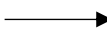

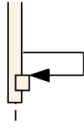

No	Notasi	Nama	Keterangan
1.	 Actor	<i>Actor</i>	Seseorang atau sesuatu yang berinteraksi dengan sistem yang sedang dikembangkan. Aktor berada diluar lingkup sistem, aktor bersifat eksternal.
2.	 Use Case	<i>Use Case</i>	<i>Use case</i> menggambarkan bagaimana aktor akan menggunakan/memanfaatkan sistem.
3.		<i>Association</i>	Relasi yang terjadi antar aktor dengan <i>use case</i> .
4.		<i>Extend</i>	Memungkinkan suatu <i>use case</i> memiliki kemungkinan untuk memperluas fungsionalitas yang disediakan <i>use case</i> lainnya. Mirip dengan <i>include</i> ,

			namun pada <i>extend</i> tidak harus terjadi apa yang diharapkan.
5.		<i>Generalization</i>	Generalisasi digunakan untuk memperlihatkan bahwa beberapa aktor atau <i>use case</i> memiliki sesuatu hal yang bersifat umum. Pada intinya, relasi ini berguna untuk mengelompokkan aktor atau <i>use case</i> .
6.		<i>Include</i>	Memungkinkan suatu <i>use case</i> untuk menggunakan fungsionalitas yang disediakan. <i>Include</i> dapat digunakan ketika terdapat dua atau lebih <i>use case</i> yang memiliki fungsi yang identik dapat dipisahkan menjadi suatu <i>use case</i> tersendiri.

b) *Sequence Diagram*

Sequence diagram adalah *diagram* yang menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambar *diagram* ini maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu [13]. Berikut ini merupakan simbol *sequence diagram* :

Tabel 2.4 Simbol *Sequence Diagram*

No.	Notasi	Nama	Keterangan
1.		<i>Entity Class</i>	Bagian dari sistem yang berisikan sekumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi dasar untuk menyusun basis data.
2.		<i>Boundary Class</i>	Berisi sekumpulan kelas yang menjadi tampilan antarmuka atau interaksi antara satu/lebih aktor dengan sistem.
3.		<i>Control Class</i>	Suatu objek yang berisi logika aplikasi yang tidak mempunyai tanggung jawab kepada entitas.
4.		<i>Message</i>	Simbol mengirim pesan antar <i>class</i>
5.		<i>Activation</i>	Berfungsi mewakili sebuah eksekusi operasi dari objek, panjang kotak dalam simbol ini berbanding lurus dengan durasi aktivasi sebuah operasi.
6.		<i>Recursive</i>	Menggambarkan pengiriman pesan untuk dikirim ke dirinya sendiri.
7.		<i>Lifeline</i>	Berupa garis titik-titik yang terhubung dengan objek, disepanjang garis ini terdapat <i>activation</i> .

2.2.4 Basis Data



Sistem basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia di saat dibutuhkan. Pada intinya basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah media untuk menyimpan data agar dapat diakses dengan mudah dan cepat. Sistem informasi tidak dapat dipisahkan dengan kebutuhan akan basis data apapun bentuknya, berupa *file* teks atau juga *Database Management System* (DBMS) [14].

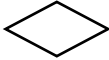

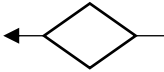
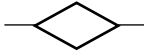
Database Management System (DBMS) adalah suatu sistem aplikasi yang digunakan untuk menyimpan, mengelola, dan menampilkan data. Suatu sistem aplikasi bisa disebut DBMS jika memenuhi kriteria [14] :

- a. Menyediakan fasilitas untuk mengelola akses data,
- b. Mampu menangani integritas data,
- c. Mampu menangani akses data yang dilakukan,
- d. Mampu menangani *backup* data.

Entity Relationship Diagram (ERD) diartikan sebagai sekumpulan cara atau peralatan untuk menggambarkan data-data maupun objek-objek yang dibuat berdasarkan dan berasal dari dunia nyata yang disebut dengan entitas (*entity*) serta hubungan (*relationship*) antar entitas tersebut dengan menggunakan beberapa notasi. Komponen pembentuk ERD antara lain [15] :

Tabel 2.5 Simbol *Entity Relationship Diagram*

No	Notasi	Nama	Keterangan
1.		Entitas	Individu yang mewakili suatu objek dan dapat dibedakan dengan objek lainnya.
2.		Atribut	Merupakan properti yang dimiliki oleh suatu entitas yang berfungsi mendeskripsikan karakteristik dari entitas tersebut.

3.		Relasi	Menunjukkan hubungan antara sejumlah entitas dengan entitas yang berbeda.
4.		Relasi 1:1	Relasi ini menunjukkan setiap entitas pada himpunan entitas pertama berhubungan dengan paling banyak satu entitas pada himpunan dengan paling banyak satu entitas pada himpunan entitas kedua.
5.		Relasi 1:N	Relasi ini menunjukkan hubungan antara entitas pertama dengan entitas kedua adalah 1 : banyak atau sebaliknya. Setiap entitis dapat berhubungan dengan banyak entitas pada himpunan entitas lain.
6.		Relasi N:N	Relasi ini menunjukkan setiap entitas pada himpunan entitas pertama dapat berhubungan dengan banyak entitas pada himpunan entitas yang kedua, demikian pula sebaliknya.

~Halaman ini sengaja dikosongkan~