BAB III

OBJEK DAN METODE PENELITIAN

3.1. Objek Penelitian

Objek penelitian merupakan hal awal (suatu permasalahan) yang harus ditentukan dalam kegiatan penelitian sehingga penelitian dapat dilakukan secara efektif dan efisien sesuai dengan tujuan penelitian. Dalam artian objek penelitian merupakan *variable* yang akan dikaji, yang menjadi *focus* penelitian disertai keberadaan/tempat dilakukannya objek penelitian tersebut.

Objek penelitian penting diungkapkan karena berperan penting dalam menunjang keberhasilan kegiatan selama penelitian, sehingga hal-hal yang diperlukan dalam penelitian akan mudah dicapai. Dengan demikian yang menjadi objek penelitian ini adalah "SMA Negeri 6 Cimahi**".** Selanjutnya untuk melengkapi objek penelitian ini, akan diuraikan tentang sejarah singkat, visi dan misi, struktur organisasi dan deskripsi tugasnya di bawah ini.

3.1.1. Sejarah Singkat Perusahaan

SMA Negeri 6 Cimahi sebelumnya disebut sebagai SMAN 13 Bandung. Angkatan pertama SMAN 13 Bandung pada saat itu masih bergabung dengan SMAN 4 Bandung di Jalan Gardujati no. 20 Bandung. Karena SMAN 13 Bandung masih baru dan masih bergabung dengan SMAN 4 Bandung maka belum bisa berjalan sendiri dalam menampung siswa didiknya. Begitu juga para staf pengajar kebanyakan masih guru-guru SMAN 4 Bandung. Pada saat itu

Kepala Sekolah SMAN 13 Bandung adalah juga Kepala Sekolah SMAN 4 Bandung yaitu Ibu. Dra. Nani Soekartini.

Setelah berpisah secara administrasi dengan SMAN 4 Bandung Kepala Sekolah SMAN 13 Bandung dipimpin oleh Bapak Komaruddin yang berasal dari Cikampek.

Pada saat itu pembangunan fisik Gedung Sekolah SMAN 13 Bandung di Cijerah masih berlangsung sehingga para siswa-siswi SMAN 13 Bandung sebagian belajar di Gedung SMAN 4 Bandung Jalan Gadujati no. 20 sebagian belajar di Gedung SMPN 1 Jalan Kesatrian Bandung.

Setelah Gedung Sekolah SMAN 13 Bandung di Cijerah selesai pada tahun 1982 maka semua siswa-siswi SMAN 13 Bandung belajar di Gedung baru SMAN 13 Bandung di Cijerah-Cimahi Selatan. SMAN 13 Bandung kemudian berubah menjadi SMA Negeri 6 Cimahi

3.1.2. Visi dan Misi Perpustakaan

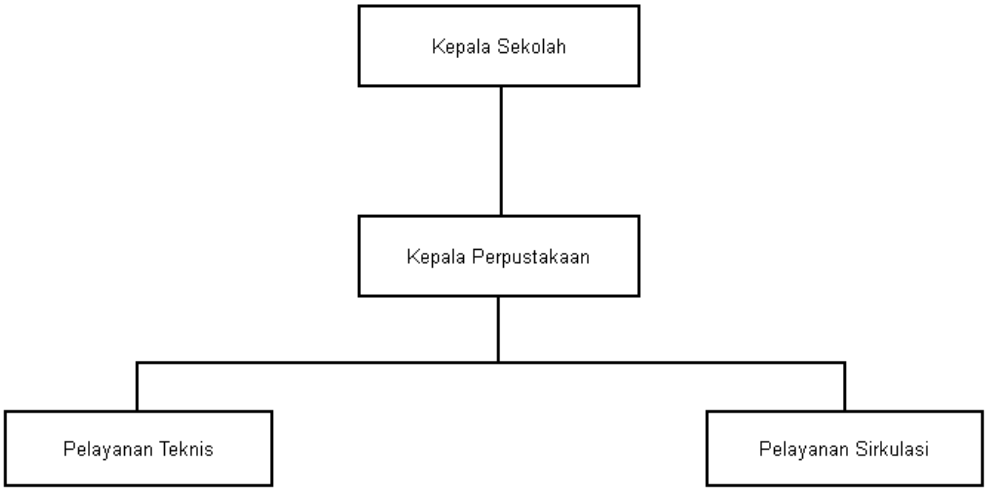
3.1.2.1. Visi

Menjadikan perpustakaan sekolah sebagai pusat informasi dengan melakukan cendikiawan yang unggul, berprestasi, berwawasan luas, cerdas, terampil dan berahklak mulia.

3.1.2.2. Misi

Mengembangkan sistem intruksional yang menjadi sarana terutama untuk/dalam meningkatkan efektifitas dan efisien kegiatan belajar mengajar.

3.1.3. Struktur Organisasi SMA Negeri 6 Cimahi



Gambar 3.1 **Struktur Organisasi Perpustakaan SMA Negeri 6 Cimahi**

3.1.4. Job Descriptions

1. Membuat perencanaan pengadaan buku bahan perpustakaan.
2. Mengurus pelayanan perpustakaan untuk siswa, guru dan karyawan.
3. Memelihara/memperbaiki buku-buku bahan perpustakaan.
4. Inventarisasi buku.
5. Penyampaian dan pengamanan buku.
6. Menggalang pendanaan pengadaan buku.

7. Membuat katalog.
8. Mengurus laporan kegiatan perpustakaan secara berkala.
9. Membina siswa menjadi pustakawan.

3.2. Metode Penelitian

Metode Penelitian oleh Cara ilmiah untuk mendapatkan data dengan tujuan dan kegunaan tertentu. Cara ilmiah didasarkan pada ciri-ciri keilmuan yaitu rasional, empiris, dan sistematis.

a. Rasional : Penelitian dilakukan dengan cara-cara yang masuk akal sehingga terjangkau oleh penalaran manusia.
b. Empiris : Cara yang digunakan dapat diamati dengan indera manusia.
c. Sistematis : Proses penelitian menggunakan langkah-langkah yang bersifat logis.

3.2.1. Desian Penelitian

Dalam melakukan suatu penelitian perlu dilakukan perencanaan penelitian agar penelitian yang dilakukan dapat berjalan dengan baik dan sistematis. Desain penelitian adalah semua proses yang diperlukan dalam perencanaan dan pelaksaan penelitian, dalam penelitian ini menggunakan jenis penelitian deskriptif dan penelitian tindakan (*action research*).

Penelitian deskriptif yaitu penelitian yang menggambarkan secara sistematika dan akurat mengenai keadaan-keadaan nyata yang berjalan pada saat penelitian. Gambaran tersebut diperoleh dengan cara mengumpulkan,

mengklasifikasikan, menyajikan, serta menganalisis data sehingga dapat ditarik suatu kesimpulan.

Penelitian tindakan (*Action Research*) ialah penelitian yang bertujuan untuk mengembangkan keterampilan-keterampilan baru, cara pendekatan baru, atau produk pengetahuan yang baru dan untuk memecahkan masalah dengan penerapan langsung di dunia aktual/lapangan. Peneliti memecahkan permasalahan yang ada dilapangan dengan mencari dan memilih alternatif metode pendekatan yang tepat.

3.2.2. Jenis dan Metode Pengumpulan Data

Pengumpulan data secara lengkap yang berhubungan dengan penelitian ini dilakukan dengan cara survei. Data yang diperlukan adalah data primer dan data sekunder.

3.2.2.1. Sumber Data Primer

Sumber data primer adalah biasanya data kualitatif yang diperoleh berdasarkan observasi dan wawancara. Data kualitatif adalah data yang tidak dihitung dengan hitungan bilangan, tetapi diukur dengan kata-kata bernilai, misalnya banyak, sedikit, kecil, rendah, manis, cantik, mahal, tinggi, panjang, berat dan sebagainya.

A. Metode Observasi (*Observation Methods*)

Teknik pengumpulan data dimana peneliti mengadakan pengamatan secara langsung terhadap gejala-gejala subjek yang diselidiki, baik pengamatan itu dilakukan didalam situasi yang sebenarnya maupun dilakukan didalam situasi buatan yang khusus diadakan pada masalah-masalah yang ada pada instansi

terkait. Dalam penelitian ini, peneliti melakukan pengamatan langsung terhadap sistem informasi perpustakaan di SMA Negeri 6 Cimahi.

1. Metode observasi merupakan metode pengumpulan data primer yang menggunakan pertanyaan lisan dan tertulis.
2. Metode ini memerlukan adanya kontak atau hubungan antara peneliti dengan subjek (*responden*) penelitian untuk memperoleh data yang diperlukan.
3. Data yang diperoleh sebagian besar merupakan data deskriptif, akan tetapi pengumpulan data dapat dirancang untuk menjelesakan sebab akibat atau mengungkapkan ide-ide.
4. Umumnya digunakan untuk mengumpulkan data yang sama dari banyak subjek.

B. Wawancara (*Interview*)

Wawancara merupakan teknik pengumpulan data dengan melakukan tanya jawab secara lisan dengan orang-orang yang berhubungan dengan penelitian. Peneliti mengumpulkan data dan menggali informasi dengan mengajukan tanya jawab secara lisan dengan bagian petugas perpustakaan di SMA Negeri 6 Cimahi.

1. Wawancara merupakan teknik pengumpulan data dalam metode survei yang menggunakan pertanyaan secara lisan kepada *responden* atau subjek penelitian.
2. Teknik wawancara dilakukan jika peneliti memerlukan komunikasi atau hubungan dengan *responden*.

3. Data yang dikumpulkan umumnya berupa masalah tertentu yang bersifat kompleks, sensitif atau kontroversial, sehingga kemungkinan jika dilakukan dengan kuesioner akan kurang memperoleh tanggapan *responden*.
4. Teknik ini terutama untuk *responden* yang tidak dapat membaca-menulis atau sejenis pertanyaan yang memerlukan penjelasan dari pewawancara atau memerlukan penerjemahan.
5. Teknik wawancara dapat dilakukan dengan (1) melalui tatap muka dan (2) melalui telepon.

3.2.2.2. Sumber Data Sekunder

Sumber data sekunder adalah data yag digunakan untuk mendukung data primer. Jenis data primer merupakan jenis data yang sudah diolah terlebih dahulu oleh pihak pertama. Pengumpulan data yang dilakukan penulis dengan cara membaca dan mempelajari buku-buku yang berhubungan dengan masalah yang diteliti untuk melengkapi dan menunjang data-data yang telah dipelajari dalam laporan. Data sekunder yang diperoleh dari instansi terkait seperti struktur organisasi, uraian tugas dan fungsi dari sruktur organisasi *(job Descriptions).*

3.2.3. Metode Pendekatan dan Pengembangan Sistem

3.2.3.1. Metode Pendekatan Sistem

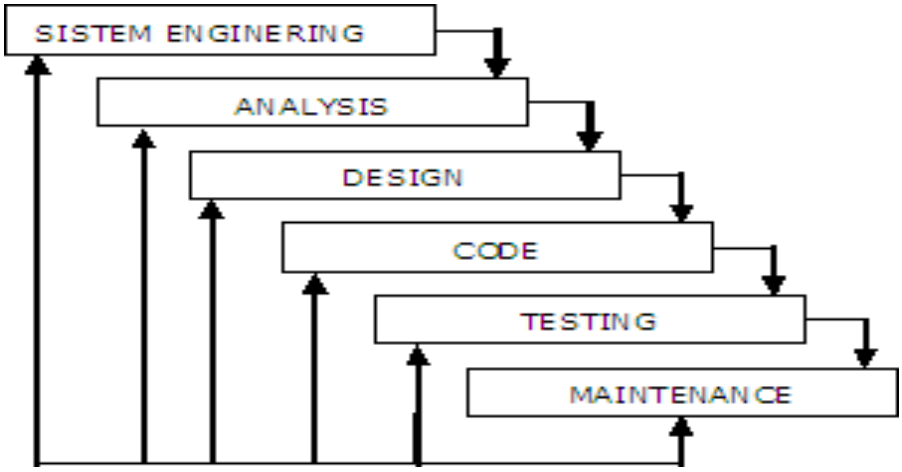
Metode pendekatan sistem dilakukan untuk mengetahui permasalahan yang dihadapi oleh sistem. Metode pendekatan perancangan terstruktur dimulai dari awal tahun 1970. Pendekatan terstruktur dilengkapi dengan alat-alat (*tools*) dan teknik-teknik (*techniques*) yang dibutuhkan dalam pengembangan sistem,

sehingga hasil akhir dari sistem yang dikembangkan akan diperoleh sistem yang strukturnya didefinisikan dengan baik dan jelas. Metode pendekatan yang dilakukan menggunakan pendekatan terstruktur. Menurut Jogiyanto (2005 : 57), "Melalui pendekatan terstruktur, permasalahan-permasalahan yang komplek di organisasi dapat dipecahkan dan hasil dari sistem akan mudah untuk dipelihara, fleksibel, lebih memuaskan pemakainya, mempunyai dokumentasi yang baik, tepat pada waktunya, sesuai dengan anggaran biaya pengembangannya, dapat meningkatkan produktivitas dan kualitasnya akan lebih baik".

3.2.3.2. Metode Pengembangan Sistem

Model *waterfall* mengusulkan sebuah pendekatan kepada perkembangan software yang sistematik dan sekuensial yang mulai pada tingkat dan kemajuan sistem pada seluruh analisis, desain, kode, pengujian, dan pemeliharaan. Model ini melingkupi aktivitas-aktivitas sebagai berikut : rekayasa dan pemodelan sistem/informasi, analisis kebutuhan, desain, *coding*, pemeliharaan dan pengujian.

Setiap *phase* pada *Waterfall* dilakukan secara berurutan namun kurang dalam iterasi pada setiap *level*. Dalam pengembangan Sistem Informasi berbasis Mobile, *Waterfall* memiliki kekakuan untuk ke iterasi sebelumnya. Dimana Sistem Informasi berbasis Mobile selalu berkembang baik teknologi ataupun lingkungannya.



Gambar 3.2 **Metode *Waterfall***

Pada perkembangan sistem informasi ini penulis menggunakan metode *waterfall* sebagai alternatif pengembangan sistem informasi dengan tujuan mendapatkan kualitas dari sistem yang dihasilkan akan baik.

Metode *waterfall* merupakan metode yang sering digunakan oleh penganalisa sistem pada umumnya. Inti dari metode *waterfall* adalah pengerjaan dari suatu sistem dilakukan secara berurutan atau secara linear. Jadi jika langkah satu belum dikerjakan maka tidak akan bisa melakukan pengerjaan langkah 2, 3 dan seterusnya. Secara otomatis tahapan ke-3 akan bisa dilakukan jika tahap ke-1 dan ke-2 sudah dilakukan.

Secara garis besar metode *waterfall* mempunyai langkah-langkah sebagai berikut : Analisa, *Design*, *Code* dan *Testing*, Penerapan dan Pemeliharaan.

Tahapan penelitian pada model *waterfall* meliputi metodologi berupa :

1. *System Engineering*

   Proses penilaian sistem lama yang sedang berjalan dan studi kelayakan pengembangan sistem baru berdasarkan aspek teknologi, ekonomis dan sumber daya manusia.

2. *Analisis*

   Perolehan kebutuhan pengguna sistem dari user serta pilihan solusi jenis system informasi yang akan dikembangkan.

3. *Design*

   Proses ini digunakan untuk mengubah kebutuhan-kebutuhan diatas menjadi representasi ke dalam bentuk software. Desain harus dapat mengimplementasikan kebutuhan yang telah disebutkan pada tahap sebelumnya maka proses ini juga harus didokumentasikan sebagai konfigurasi dari software.

4. *Coding Dan Testing*

   Desain harus diubah bentuknya menjadi bentuk yang dapat dimengerti oleh komputer, yaitu ke dalam bahasa pemrograman melalui proses *Coding*. Tahap ini merupakan implementasi dari tahap desain yang secara teknis akan dikerjakan oleh *programmer*. Proses *Coding* ini harus dilakukan Testing untuk menguji kesalahan-kesalahan program maupun fungsi dari sistem.

5. Implementasi

   Setelah semua fungsi-fungsi software harus di ujicoba agar software bebas dari kesalahan, dan hasilnya harus benar-benar sesuai dengan kebutuhan

yang sudah didefinisikan sebelumnya. Maka proses selanjutnya adalah bagaimana sistem baru akan diinstall dan dijalankan di perusahaan dengan pengoperasian yang dilakukan oleh user.

6. Pemeliharaan

   Pemeliharaan suatu software sangat diperlukan, termasuk di dalamnya adalah pengembangan, karena software yang dibuat tidak selamanya hanya seperti itu. Ketika dijalankan mungkin saja masih ada kesalahan kecil yang tidak ditemukan sebelumnya, atau ada penambahan fitur-fitur yang belum ada pada software tersebut. Pengembangan diperlukan ketika adanya perubahan dari perusahaan seperti ketika ada pergantian sistem operasi atau perangkat lainnya.

3.2.3.3. Alat Bantu Analisis dan Perancangan

Dalam suatu analisis dan perancangan suatu sistem diperlukan alat bantu untuk menunjang dari analisis dan perancangan itu sendiri, agar lebih terstruktur dalam pengerjaannya. Alat bantu yang digunakan peneliti adalah AMPPS dan NetBeans.

1. AMPPS adalah setumpuk Apache Web Server, MySQL, PHP, Perl, Python dan Softaculous. Dengan AMPPS dapat membuat beberapa database MySQL dan mengaturnya dengan mudah pada phpMyAdmin.
2. NetBeans adalah sebuah lingkungan pengembangan sebuah kakas untuk pemrogram menulis, mengompilasi, mencari kesalahan dan menyebarkan program. Netbeans IDE ditulis dalam Java namun dapat mendukung bahasa pemrograman lain.

3. iReport adalah sebuah Visual Designer untuk membuat laporan yang komplek, menggunakan jasperReport *Libarary*.

4. Use case Diagram

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah "apa" yang di perbuat *system*, dan bukan "bagaimana". Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use case* merupakan sebuah pekerjaan tertentu misalnya *login* sistem, meng-*create* sebuah daftar belanja dan sebagainya. Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu.

Use case diagram dapat sangat membantu bila kita sedang menyusun *requirement* sebuah sistem, mengkomunikasikan rancangan klien, dan merancang *test case* untuk semua fitur yang ada pada sistem. Sebuah *use case* dapat meng-*include* fungsionalitas *use case* lain sebagai bagian dari proses dalam dirinya. Secara umum diasumsikan bahwa *use case* yang di *include* akan di panggil setiap kali *use case* yang mengincluded eksekusi secara normal.

Sebuah *use case* dapat di include oleh lebih dari satu *use case* lain, sehingga duplikasi fungsionalitas dapat dihindari dengan menarik keluar fungsionalitas yang common. Sebuah *use case* juga dapat meng *extend use case* lain dengan behaviornya sendiri. Sementara hubungan generalisasi antar *use case* menunjukkan bahwa *use case* yang satu merupakan spesialisasi dari yang lain.

5. *Activity Diagram*

Activity Diagram atau yang di sebut Diagram Aktivitas digunakan untuk mendokumentasikan alur kerja pada sebuah *system*, yang di mulai dari pandangan *business level* hingga ke *operational level*, adapaun penjelasanya antara lain :

1. *Activity diagram* merupakan cara lain memodelkan aliran kejadian.
2. Kita gunakan *activity diagram* untuk menggambarkan *workflow*/aliran kerja proses bisnis.
3. *Activity diagram* menggambarkan berbagai alir aktivitas dalam sistem yang sedang di rancang bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir.

Activity diagram juga dapat menggambarkan proses parallel yang mungkin terjadi pada beberapa eksekusi.

6. *Class Diagram*

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek.

Class menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi). *Class diagram* menggambarkan struktur dan deskripsi *class, package* dan objek beserta belanja, dan sebagainya. Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan *system* untuk melakukan pekerjaan-pekerjaan tertentu.

Use case diagram dapat sangat membantu bula kita sedang menyusun *requirement* sebuah sistem, mengkomunikasikan rancangan dengan klien, dan

merancang *test case* untuk semua fitur yang ada pada sistem. Sebuah *use case* dapat menginclude fungsionalitas *use case* lain sebagai bagian dari proses dalam dirinya. Secara umum diasumsikan bahwa *use case* yang di *include* akan di panggil setiap kali *use case* yang meng *include* dieksekusi secara normal. Sebuah *use case* dapat di *include* oleh lebih dari satu *use case* lain, sehingga duplikasi fungsionalitas dapat dihindari dengan cara menarik keluar fungsionalitas yang *common*. Sebuah *use case* juga dapat meng *extend use case* lain dengan *behavior* sendiri. Sementara hubungan generalisasi antar *use case* menunjukan bahwa *use case* yang satu merupakan spesialisasi dari yang lain.

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metode/fungsi). *Class diagram* menggambarkan struktur dan deskripsi *class, package* dan objek beserta hubungan antar *class*.

7. Sequence Diagram

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, display, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait). *Sequence diagram* biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai *respons* dari sebuah *event* untuk menghasilkan *output* tertentu. Diawali dari apa yang men- *trigger* aktivitas tersebut, proses dan

perubahan apa saja yang terjadi secara *internal* dan *output* apa yang dihasilkan. Masing-masing objek, termasuk aktor, memiliki *lifeline* vertikal.

Message digambarkan sebagai garis berpanah dari satu objek ke objek lainnya. Pada *fase* desain berikutnya, *message* akan dipetakan menjadi operasi/metoda dari *class*. *Activation* bar menunjukkan lamanya eksekusi sebuah proses, biasanya diawali dengan diterimanya sebuah *message*.Untuk objek-objek yang memiliki sifat khusus, standar UML mendefinisikan icon khusus untuk objek *boundary*, *controller* dan *persistent entity*.

8. *Collaboration Diagram*

Collaboration diagram juga menggambarkan interaksi antar objek seperti *sequence diagram,* tetapi lebih menekankan pada peran masing-masing objek dan bukan pada waktu penyampaian *message*.

Setiap *message* memiliki *sequence number*, di mana *message* dari *level* tertinggi memiliki nomor 1. *Messages* dari *level* yang sama memiliki prefiks yang sama.

9. *Component Diagram*

Component diagram menggambarkan struktur dan hubungan antar komponen piranti lunak, termasuk ketergantungan *(dependency)* di antaranya. Komponen piranti lunak adalah modul berisi *code,* baik berisi *source code* maupun *binary code*, baik *library* maupun *executable*, baik yang muncul pada *compile time*, *link time,* maupun *run time*. Umumnya komponen terbentuk dari beberapa *class* dan/atau *package*, tapi dapat juga dari komponen-komponen yang lebih kecil.

Komponen dapat juga berupa *interface*, yaitu kumpulan layanan yang disediakan sebuah komponen untuk komponen lain.

10. Deployment Diagram

Deployment/physical diagram menggambarkan detail bagaimana komponen di-deploy dalam infrastruktur sistem, di mana komponen akan terletak (pada mesin, server atau piranti keras apa), bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi server, dan hal-hal lain yang bersifat fisikal . Sebuah node adalah server, *workstation*, atau perangkat keras lain yang digunakan untuk men-deploy komponen dalam lingkungan sebenarnya. Hubungan antar node (misalnya TCP/IP) dan *requitment* dapat juga didefinisikan dalam diagram ini.

3.2.4. Pengujian Software

Pengujian Software (perangkat Lunak) adalah elemen kritis dari jaminan kualitas perangkat lunak dan mempresentasikan kajian pokok dari spesifikasi, desain dan pengkodean.

Metode pengujian yang digunakan oleh penulis adalah *black-box testing*. *Black Box* Testing digunakan untuk menguji fungsi-fungsi dari perangkat lunak yang dirancang. Pengujian *black box* berfokus pada persyaratan fungsional perangkat lunak. Dengan demikian, pengujuan *black box* memungkinkan perekayasa perangkat lunak mendapat serangkaian kondisi *input* yang sepenuhnya menggunakan semua persyaratan fungsional untuk semua program. Pengujian *black box* merupakan pendekatan komlementer yang kemungkinan besar mampu mengungkap kelas kesalahan. Pengujian *black box* berusaha menemukan kesalahan dalam kategori sebagai berikut :

1. Fungsi yang tidak benar atau hilang.
2. Kesalahan antar muka.
3. Kesalahan dalam struktur data atau akses database *eksternal*.
4. Kesalahan kinerja dan Inisialisasi kesalahan terminasi.

3.3 Analisis Sistem Yang Berjalan

Analisis sistem merupakan gambaran tentang sistem yang saat ini sedang berjalan dan untuk mempelajari sistem yang ada. Analisis sistem diperlukan untuk menggambarkan aliran-aliran informasi dari bagian-bagian yang terkait, baik dari dalam maupun dari luar organisasi, untuk mengidentifikasikan dan mengevaluasi permasalahan-permasalahan, hambatan-hambatan yang terjadi dan kebutuhan-kebutuhan yang diharapkan, sehingga dapat diusulkan perbaikan-perbaikan yang akan dilakukan pada sistem tersebut.

Pembahasan dalam analisis sistem yang akan dilakukan adalah pada bagian perpustakaan. Adapun ruang lingkup kegiatannya adalah meminjam dan mengembalikan buku pada perpustakaan di SMAN 6 Cimahi.

3.3.1. Analisis Dokumen

Analisis dokumen merupakan penjelasan mengenai dokumen-dokumen apa saja yang digunakan dalam sistem informasi pelayanan perpustakaan SMA Negeri 6 Cimahi. dalam analisis.

dokumen akan menjelaskan hal-hal berikut:

Nama dokumen : Untuk menjelaskan nama dokumen tersebut

Fungsi : Untuk menjelaskan kegunaan informasi yang digunakan

Sumber : Asal dokumen

Distribusi : Menjelaskan kebagian mana informasi mengalir

Bentuk : Dokumen yang digunakan dalam bentuk apa

Rangkap : Berapa rangkap laporan yang ada dibuat

3.3.2. Analisis Prosedur Yang Sedang Berjalan

Analisis prosedur yang sedang berjalan menguraikan secara sistematis aktifitas-aktifitas yang terjadi dalam sistem informasi perpustakaan, yang meliputi pemberitahuan tentang absensi, peminjaman buku, pengembalian buku dan membaca buku di perpustakaan SMA Negeri 6 Cimahi.

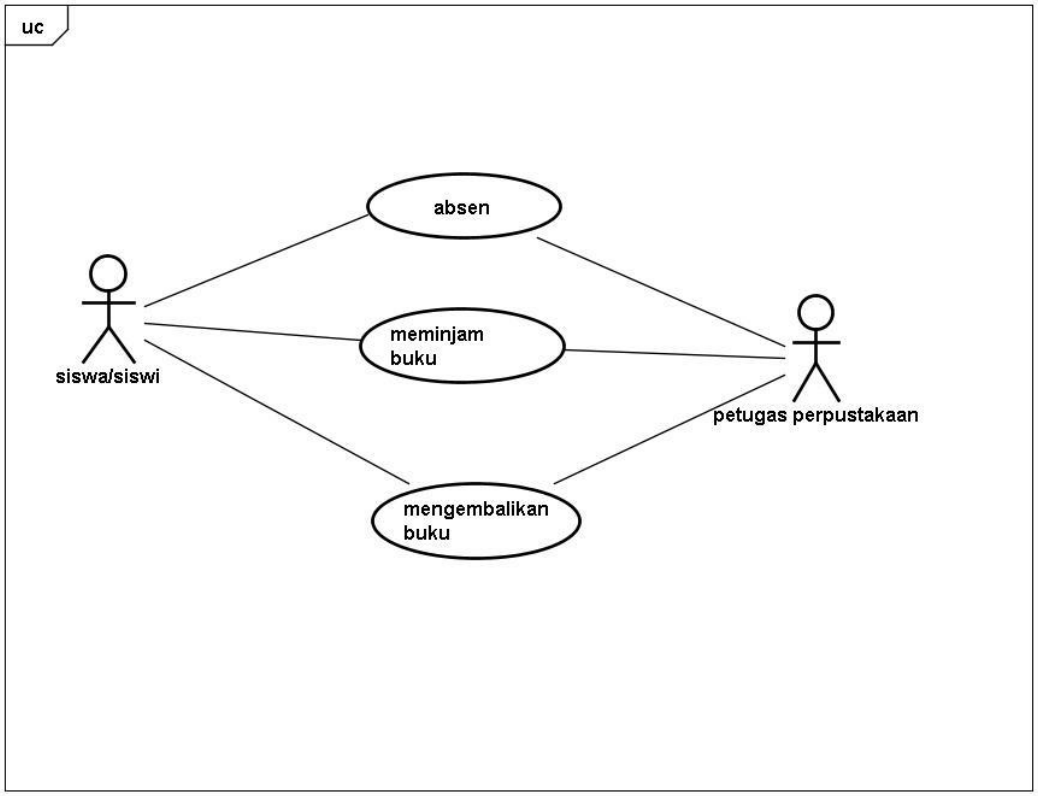
3.3.2.1. Use Case Diagram Yang Sedang Berjalan

Diagram *use case* adalah deskripsi fungsi dari sebuah sistem dari perspektif/sudut pandang para pengguna sistem. *Use case* mendefinisikan "apa" yang dilakukan oleh sistem dan elemen-elemennya, bukan "bagaimana" sistem dan elemen-elemennya aling berinteraksi. *Use case* bekerja dengan menggunakan "*scenario*", yaitu deskripsi urutanurutan langkah yang menerangkan apa yang dilakukan penggunan terhadap sistem maupun sebaliknya.

Diagram *use case* mengidentifikasikan fungsionalitas yang dipunyai oleh sistem (*use case*), user yang berinteraksi dengan sistem (*actor*) dan asosiasi/keterhubungan antara user dengan fungsionalitas sistem. Komponen notasi dasar yang dipunyai oleh diagram *use case* adalah *actor*, *usecase*, dan *association*. Berikut adalah notasi yang terdapat pada diagram *use case*. Diagram *use case* memiliki sebuah model khusus yang terbatas untuk kondisi tertentu yang disebut *stereotype*. Untuk menunjukkan *stereotype* digunakan symbol "<<"

diawalnya dan ditutup dengan “>>” diakhirnya. Terdapat dua *stereotype* paling sering digunakan dalam *use case diagram* yaitu <<extend>> dan <<include>>. <<extend>> digunakan untuk menunjukkan bahwa satu *use case* merupakan tambahan fungsional dari *use case* yang lain jika kondisi atau syarat tertentu dipenuhi. Sedangkan <<include>> digunakan untuk menggambarkan bahwa suatu *use case* seluruhnya merupakan fungsionalitas dari *use case* lainnya.

Contoh salah satu *use case* model pada sebuah pemesanan buku diperpustakaan :



Gambar 3.3 *Use Case Diagram* Yang Sedang Berjalan

3.3.2.2. Definisi Aktor dan Deskripsinya Yang Sedang Berjalan

Aktor adalah segala hal diluar sistem yang akan menggunakan sistem tersebut untuk melakukan sesuatu. Tidak semua aktor adalah manusia, bisa saja sistem lain yang berinteraksi dengan sistem yang dibuat.

Tabel 3.1 Definisi Aktor dan Deskripsi Yang Berjalan

No	Aktor	Deskripsi
1	Siswa	Pihak yang melakukan proses peminjaman dan pengembalian buku ke perpustakaan
2	Petugas Perpustakaan	Pihak yang bertugas melakukan proses pencatatan laporan peminjaman dan pengembalian buku.

3.3.2.3. Definisi Use Case Yang Sedang Berjalan

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah "apa" yang diperbuat sistem, dan bukan "bagaimana". Sebuah *use case* merepresentasikan sebuah interaksi antara satu atau lebih *actor* dengan sistem.

Tabel 3.2 Definisi *Use Case* dan Deskripsi Yang Berjalan

No	Use Case	Deskripsi
1	peminjaman	Merupakan proses dimana siswa/siswi meminjam buku diperpustakaan.
2	pengembalian	Merupakan proses dimana siswa/siswa megembalikan buku

		ke perpustakaan.
3	Absensi	Merupakan proses pengolahan absensi siswa/siswi yang ingin membaca, meminjam dan mengembalikan buku ke perpustakaan.

3.3.2.4. Skenario Use Case Yang Sedang Berjalan

Skenario *use case* berfungsi untuk mengetahui alur cerita dari *activity* diagram tersebut agar lebih mudah dimengerti. Berdasarkan *use case* diagram yang sedang berjalan pada SMA Negeri 6 Cimahi diatas, maka skenario *use case*nya sebagai berikut :

1. Skenario Use Case Absensi Yang Sedang Berjalan

Adapun skenario *use case* absensi dalam sistem yang berjalan adalah sebagai berikut:

Tabel 3.3 Skenario *Use Case* Absensi Yang Sedang Berjalan

Identifikasi		
Nama Use Case	Absensi	
Tujuan	Menjelaskan proses pengolahan absensi siswa yang datang ke perpustakan.	
Deskripsi		
Aktor	Siswa dan petugas perpustakaan.	
Skenario Utama		
Kondisi Awal	Siswa datang ke perpustakaan	
Aksi Aktor		**Reaksi Sistem**

1. Siswa mengisi absensi	
2. Data absen siswa tersebut diserahkan kepada bagian petugas perpustakaan	
	3. Bagian petugas perpustakaan menerima data absensi
Kondisi Akhir	Petugas perpustakaan menyimpan data absensi siswa/siswi

2. Skenario Use Case Peminjaman Yang Sedang Berjalan

Adapun skenario *use case* peminjaman dalam sistem yang berjalan adalah sebagai berikut:

Tabel 3.4 Skenario *Use Case* peminjaman Yang Sedang Berjalan

Identifikasi	
Nama Use Case	Peminjaman
Tujuan	Menjelaskan proses peminjaman buku di perpustakaan.
Deskripsi	
Aktor	Siswa dan petugas perpustakaan.
Skenario Utama	
Kondisi Awal	Siswa mencari buku yang akan dipinjam
Aksi Aktor	**Reaksi Sistem**
1. Siswa memberikan buku yang akan dipinjam ke petugas perpustakaan.	
	2. Petugas mencatat dan menyimpan

	data peminjaman.
Kondisi Akhir	Petugas perpustakaan memberikan buku kepada siswa/siswi.

3. Skenario Use Case Pengembalian Yang Sedang Berjalan

Adapun skenario *use case* pengembalian dalam sistem yang berjalan adalah sebagai berikut:

Tabel 3.5 Skenario *Use Case* pengembalian Yang Sedang Berjalan

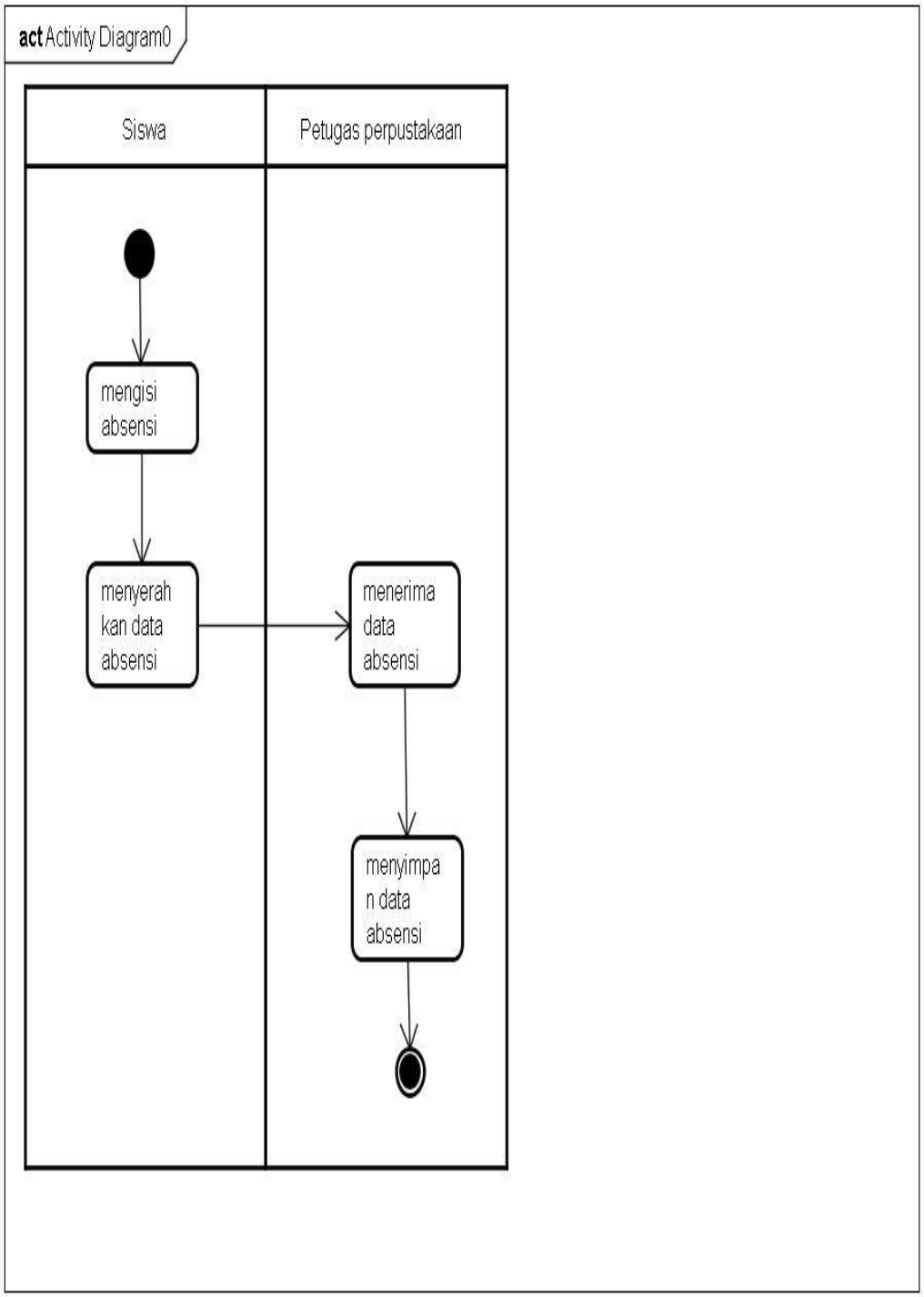
Identifikasi	
Nama Use Case	Pengembalian
Tujuan	Menjelaskan proses mengembalikan buku ke perpustakaan.
Deskripsi	
Aktor	Siswa dan petugas perpustakaan.
Skenario Utama	
Kondisi Awal	Siswa datang ke perpustakaan.
Aksi Aktor	**Reaksi Sistem**
1. Siswa mengembalikan buku yang sudah dipinjam ke perpustakaan.	
	3. Petugas perpustakaan memeriksa buku dan data peminjaman, jika pengembalian terlambat siswa dikenakan denda.

Kondisi Akhir	Petugas perpustakaan menyimpan data pengembalian.

3.3.3. Activity Diagram Yang Sedang Berjalan

Activity diagram menggambarkan berbagai alur aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. Berikut adalah gambar *activity diagram* pada proses yang sedang berjalan :

1. Prosedur absensi yang sedang berjalan
    a. Siswa/siswi mengisi absensi.
    b. Siswa/siswi memeberikan absensi ke petugas perpustakaan.
    c. Petugas perpustakaan menerima data absensi.
    d. Petugas menyimpan data absensi.



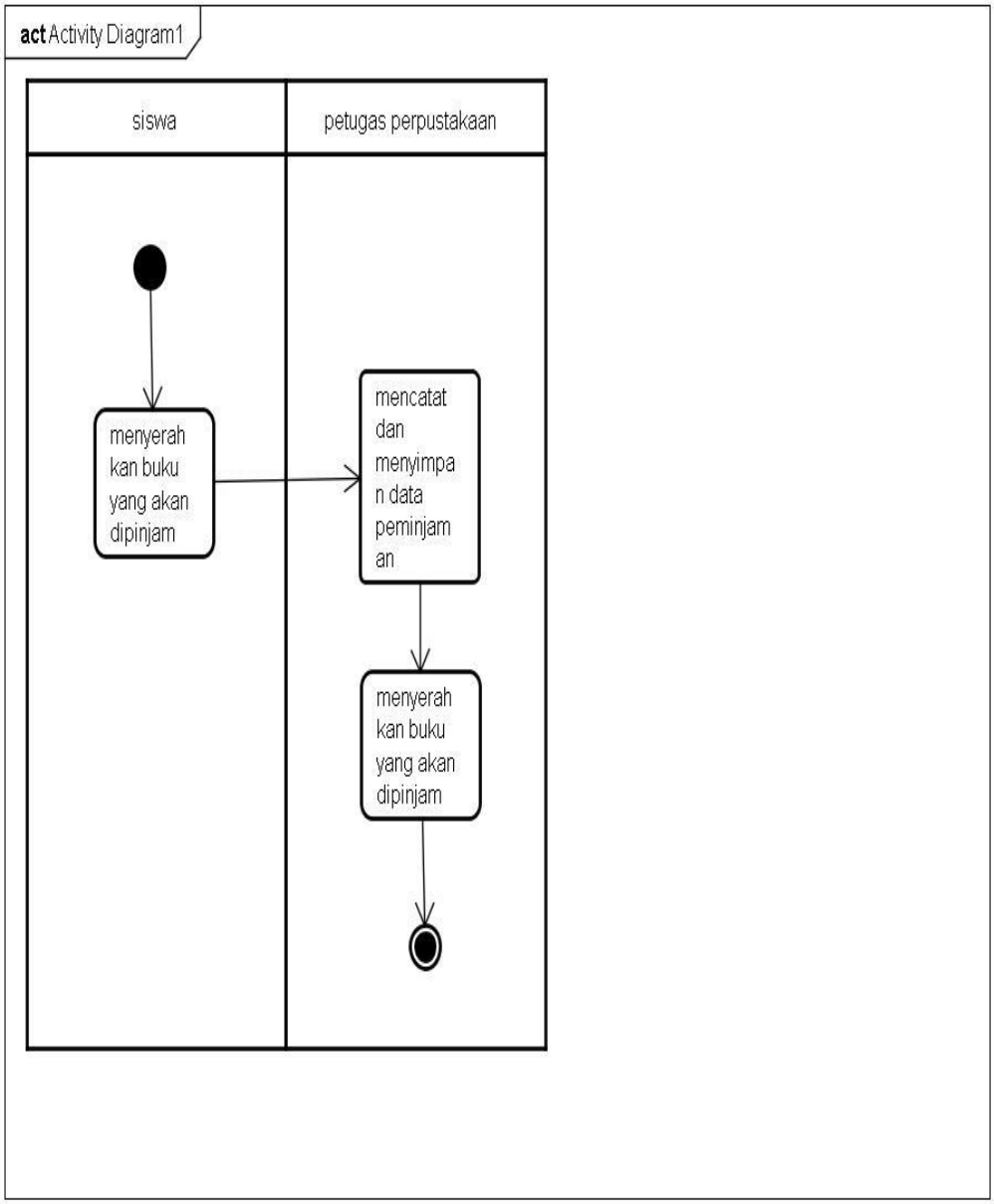
Gambar 3.4 *Activity Diagram* Absensi Yang Sedang Berjalan.

2. Prosedur peminjaman yang sedang berjalan

   a. Siswa memberikan buku yang akan dipinjam ke petugas perpustakaan.

   b. Petugas mencatat dan menyimpan data buku yang akan dipinjam.

   c. Petugas memberikan buku pada siswa.



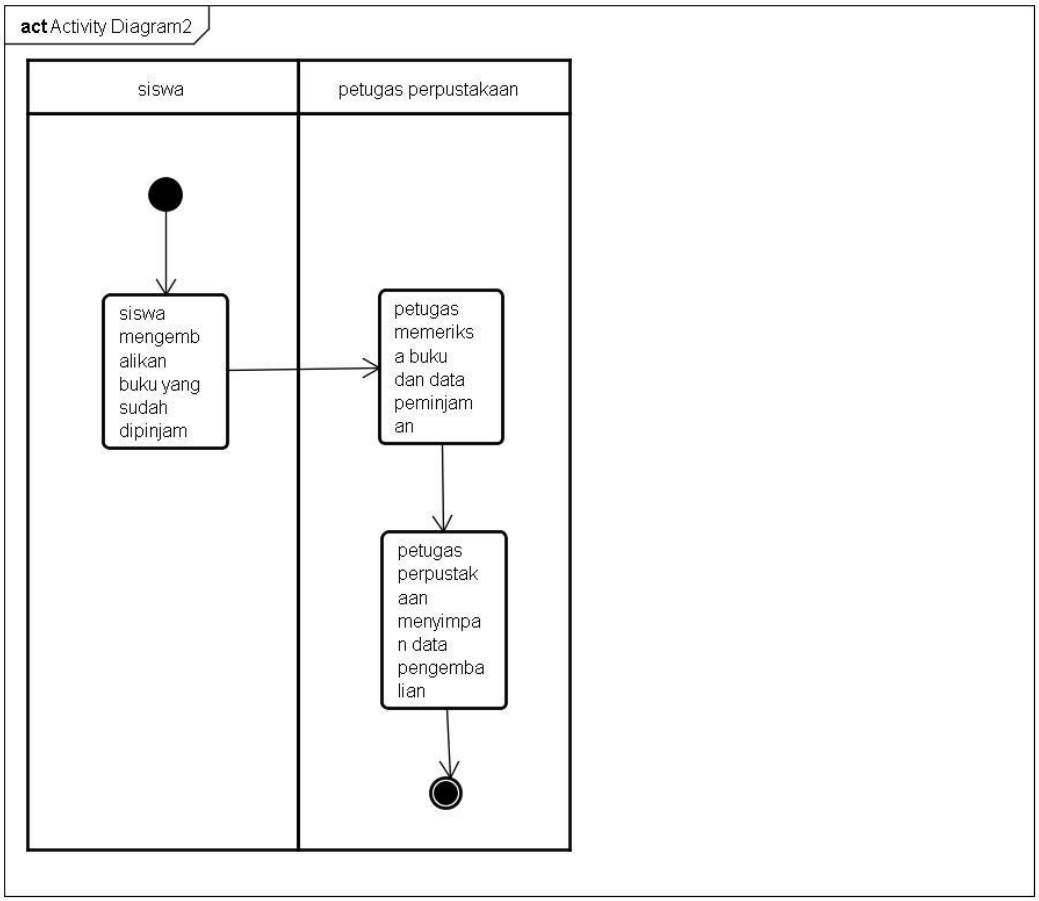
Gambar 3.5 *Activity Diagram* Peminjaman Yang Sedang Berjalan.

3. Prosedur pengembalian yang sedang berjalan

   a. Siswa mengembalikan buku yang sudah dipinjam.

   b. Petugas memeriksa buku dan data peminjaman bila siswa tersebut terlambat mengembalikan maka akan terkena denda.

   c. Petugas perpustakaan menyimpan data pengembalian.



Gambar 3.6 *Activity Diagram* Pengembalian Yang Sedang Berjalan.

3.3.4. Evaluasi Sistem Yang Sedang Berjalan

Berikut ini eveluasi sistem berdasarkan analisis di atas :

Tabel 3.6 Evaluasi Sistem Yang Sedang Berjalan

No.	Masalah	Rencana Pemecahan
1	Pada sirkulasi peminjaman dan pengembalian masih di catat dalam buku besar peminjaman.	Membangun Aplikasi Sistem Informasi perpustakaan Berbasis Dekstop yang dapat membantu dalam penyajian data perpustakaan, sehingga dapat meminimalisir resiko kehilangan dan keamanan data yang terjamin.
2	Pada pembuatan laporan masih dilakukan di dalam buku laporan.	Membuat sistem pengolahan data laporan sehingga petugas perpustakaan tidak perlu menulis di buku besar.
3	Belum adanya sistem pencarian data buku.	Membuat sistem informasi perpustakaan yang memudahkan petugas perpustakaan mencari data buku