

BAB II

LANDASAN TEORI

2.1 Tinjauan Pustaka

Dalam sub bab ini peneliti akan memaparkan tinjauan peneliti atas beberapa penelitian dan kajian ilmiah terdahulu serta beberapa konsep yang memiliki keterkaitan dengan penelitian ini, tinjauan pustaka pertama peneliti lakukan atas jurnal yang berjudul “Sistem Pengolahan Data Kegiatan Penelitian dan Pengabdian Kepada Masyarakat di Universitas Respati Yogyakarta”. (Hamzah, 2016)

Penelitian yang dilakukan tersebut bertujuan menghasilkan luaran sistem informasi yang dapat melakukan proses pengolahan data penelitian dan kegiatan pengabdian kepada masyarakat yang dikelola oleh Unit Penelitian dan Pengabdian pada Masyarakat di Universitas Respati Yogyakarta.

Metode pengumpulan data yang digunakan dalam penelitian tersebut adalah studi pustaka dan studi lapangan dengan melakukan wawancara. Sedangkan metode pengembangan sistem yang digunakan adalah metode *Waterfall*. Berdasarkan hasil analisis, kesimpulan yang dapat diperoleh dari kegiatan penelitian yang dilakukan oleh Hamzah tersebut adalah sistem pengolahan data yang dikembangkan adalah *web database* yang dapat mengolah data kegiatan penelitian dan pengabdian kepada masyarakat di Universitas Respati Yogyakarta. Sistem yang dibangun berjalan sesuai dengan kebutuhan Universitas tersebut.

Penelitian yang dilakukan oleh Hamzah tersebut memiliki kesamaan dengan penelitian yang dilakukan oleh penulis, yaitu sama-sama membangun suatu sistem informasi yang dapat mendukung kinerja lembaga penelitian dan pengabdian kepada masyarakat..Persamaan lainnya terdapat pada metode pengumpulan data

dan metode pengembangan sistem yang digunakan. Perbedaan pertama yang terdapat pada penelitian yang dilakukan oleh Hamzah dengan penelitian yang dilakukan oleh penulis yaitu, tempat penelitian. Penelitian yang dilakukan oleh Hamzah bertempat di Universitas Respati Yogyakarta, sedangkan penulis melakukan penelitian di LPPM STT-PLN Jakarta. Perbedaan kedua yaitu sistem informasi yang dibuat oleh Hamzah melingkupi semua hal yang berhubungan dengan kegiatan penelitian dan pengabdian kepada masyarakat, sedangkan sistem informasi yang dibuat oleh penulis hanya mengacu kepada hal yang berhubungan dengan penelitian dosen.

Penelitian kedua yang dijadikan tinjauan pustaka oleh peneliti adalah penelitian yang berjudul “Pengembangan Sistem Informasi Kegiatan Penelitian dan Pengabdian Masyarakat Universitas Hasanuddin”. (Kusumaputra., 2015)

Tujuan dari penelitian tersebut adalah membangun sistem informasi sebagai wadah untuk menyimpan dan mengolah data penelitian dan pengabdian masyarakat pada pihak LP2M Universitas Hasanuddin.

Metodologi pengembangan sistem yang digunakan pada penelitian ini adalah metode *Waterfall*. Dalam penelitian tersebut, para peneliti menggunakan teknik pengumpulan data berupa studi lapangan yaitu wawancara langsung kepada pihak pegawai administrasi LP2M. Metode wawancara yang digunakan adalah wawancara bebas, metode ini digunakan dengan tetap membatasi topik pembicaraan agar didapatkan seluruh kendala teknis yang dihadapi oleh pengguna sehubungan dengan pengumpulan data penelitian dan pengabdian.

Untuk memperoleh data yang bersifat detail dan numerik, para peneliti tersebut juga melakukan wawancara tertutup dengan dosen pembimbing selaku dosen yang pernah melakukan penelitian ataupun pengabdian kepada masyarakat. Setelah melakukan pengumpulan data, dilakukan proses analisa.

Berdasarkan hasil analisis, kesimpulan yang dapat diperoleh dari penelitian ini yaitu pembuatan sistem informasi sebagai wadah untuk menyimpan dan

mengolah data penelitian dan pengabdian kepada masyarakat di LP2M Universitas Hasanuddin berjalan dengan baik dan sistem berjalan sesuai dengan kebutuhan.

Penelitian ini memiliki kesamaan dengan penelitian yang dilakukan oleh penulis, yaitu sama-sama ingin membuat suatu sistem informasi yang digunakan sebagai wadah untuk menyimpan dan mengolah data penelitian dosen pada suatu lembaga. Persamaan kedua terletak pada metodologi pengembangan sistem yang digunakan yaitu metode *Waterfall*. Perbedaan antara penelitian yang dilakukan oleh Kusumaputra dan tim dengan penelitian yang dilakukan oleh penulis yaitu mengenai tempat penelitian, sasaran penelitian, dan aktor-aktor yang diperlukan sistem. Kusumaputra dan tim melakukan penelitian di LP2M Universitas Hasanuddin, sedangkan penulis melakukan penelitian di LPPM STT-PLN Jakarta. Perbedaan yang kedua terletak pada sistem informasi yang dibuat oleh Kusumaputra dan tim yang memuat bidang penelitian dan pengabdian masyarakat sedangkan sistem informasi yang akan dibuat oleh penulis hanya membahas bidang penelitian dosen. Perbedaan yang ketiga yaitu pada aktor yang digunakan pada sistem informasi yang dibuat oleh Kusumaputra dan tim yang tidak menggunakan aktor *reviewer* sedangkan penulis menggunakan aktor *reviewer* pada sistem informasi yang akan dibuat, dimana aktor ini bertugas untuk melakukan penilaian terhadap proposal penelitian yang diajukan oleh dosen.

Penelitian ketiga yang dijadikan tinjauan pustaka oleh peneliti adalah penelitian yang berjudul “Sistem Informasi Penelitian Dosen”. (Latuamury., 2014)

Tujuan dari penelitian tersebut adalah memberikan sebuah kemudahan bagi LPPM dan dosen Universitas Narotama Surabaya dalam memantau dan mendata keseluruhan data penelitian dan laporan hasil penelitian yang dihasilkan oleh sistem.

Teknik pengumpulan data yang diterapkan dalam penelitian ini adalah teknik pengumpulan data dengan metode wawancara. Wawancara ini dilakukan kepada kepala LPPM yang membidangi penelitian dosen untuk mendapatkan

data-data yang kongkrit mengenai tatacara penelitian. Setelah pengumpulan data selesai dilakukan, maka tahap selanjutnya yaitu analisis permasalahan.

Berdasarkan hasil analisis, kesimpulan yang dapat diambil dari penelitian ini adalah kerja sistem informasi penelitian yang dibuat hanya difokuskan pada proses pendataan, sehingga sistem tidak melakukan proses penilaian kelayakan usulan tim, judul, dan proposal serta laporan tahap I dan tahap II.

Penelitian tersebut mempunyai kesamaan dengan penelitian yang dilakukan oleh penulis yaitu sama-sama membuat sistem informasi penelitian dosen pada suatu LPPM. Perbedaan antara penelitian yang dilakukan oleh Latuamury dan tim dengan penelitian yang dilakukan oleh penulis yaitu tempat penelitian dan cakupan kerja sistem. Penelitian Latuamury dan tim dilakukan pada LPPM Universitas Narotama Surabaya sedangkan penelitian penulis dilakukan pada LPPM STT-PLN Jakarta. Kemudian, kerja sistem informasi yang dibuat pada penelitian tersebut hanya difokuskan pada proses pendataan sedangkan kerja sistem informasi yang akan dibuat oleh penulis mencakup proses penilaian kelayakan proposal pengajuan oleh LPPM, penilaian pemenang oleh *reviewer*, hingga publikasi luaran hasil penelitian yang akan ditampilkan pada *web* ini.

2.2 Landasan Teori

2.2.1 Pengertian Sistem

Sistem merupakan sekumpulan elemen yang saling berkaitan dan saling mempengaruhi dalam melakukan berbagai kegiatan bersama untuk mencapai suatu tujuan. Sistem terbagi menjadi dua, yaitu dilihat dari pendekatan yang menekankan pada prosedur dan pendekatan yang menekankan pada elemen komponennya.

Menurut Jogiyanto (2015) pendekatan sistem yang menekankan pada prosedur mendefinisikan sistem sebagai jaringan kerja dan prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan sasaran tertentu.

Adapun pendekatan sistem yang menekankan pada elemen atau komponennya mendefinisikan sistem sebagai kumpulan dan elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu.

Dari kedua pengertian diatas maka jelaslah bahwa sistem merupakan kumpulan elemen-elemen atau jaringan kerja dan prosedur-prosedur yang saling berinteraksi satu sama lain untuk mencapai tujuan atau sasaran tertentu.

1. Karakteristik Sistem

Sistem mempunyai karakteristik atau sifat-sifat tertentu, yakni : komponen-komponen (*components*), batas sistem (*boundary*), lingkungan luar sistem (*environments*), penghubung (*interface*), masukan (*input*), keluaran (*output*), pengolahan (*process*), dan sasaran (*objectives*), atau tujuan (*goal*). (Jogiyanto, 2005)

a. Komponen Sistem

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang artinya saling bekerja sama membentuk satu kesatuan. Komponen-komponen sistem atau elemen-elemen sistem dapat berupa suatu sub sistem atau bagian-bagian dari sistem. Setiap sub sistem mempunyai sifat-sifat dari sistem untuk menjalankan suatu fungsi tertentu mempengaruhi proses sistem secara keseluruhan

b. Batasan Sistem

Batasan sistem (*boundary*) merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batasan sistem ini memungkinkan suatu sistem dipandang sebagai satu kesatuan. Batasan suatu sistem menunjukkan ruang lingkup (*scope*) dari sistem tersebut.

c. Lingkungan Luar Sistem

Lingkungan luar (*environment*) dari suatu sistem adalah apapun diluar dari sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem dapat bersifat menguntungkan dan dapat juga bersifat merugikan sistem tersebut. Lingkungan luar yang menguntungkan merupakan energi dari sistem dan dengan demikian harus tetap dijaga dan dipelihara. Sedangkan lingkungan yang merugikan harus ditahan dan dikendalikan, kalau tidak maka akan mengganggu keberlangsungan hidup dari sistem.

d. Penghubung Sistem

Penghubung (*interface*) merupakan media penghubung antara satu sub sistem dengan sub sistem lainnya. Melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari sub sistem ke sub sistem lain. Keluaran (*output*) dari sub sistem akan menjadi masukan (*input*) untuk sub sistem lain melalui penghubung.

e. Masukan Sistem

Masukkan adalah energi yang dimasukkan kedalam sistem, yang dapat berupa perawatan (*maintenance input*), dan masukkan sinyal (*signal input*). Maintenance input adalah energi yang dimasukkan agar sistem dapat beroperasi. Signal input adalah energi yang diproses untuk didapatkan keluaran. Contoh dalam sistem komputer program adalah maintenance input sedangkan data adalah signal input untuk diolah menjadi informasi.

f. Keluaran Sistem

Keluaran (*output*) adalah hasil energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan. Contoh komputer menghasilkan panas yang merupakan sisa pembuangan, sedangkan informasi adalah keluaran yang dibutuhkan.

g. Pengolahan Sistem

Suatu sistem menjadi bagian pengolah yang akan merubah masukan menjadi keluaran. Sistem produksi akan mengolah bahan baku menjadi bahan jadi, sistem akuntansi akan mengolah data menjadi laporan-laporan keuangan.

h. Sasaran Sistem

Suatu sistem pasti mempunyai tujuan (*goal*) atau sasaran (*objective*). Sasaran dari sistem sangat menentukan input yang dibutuhkan sistem dan keluaran yang akan dihasilkan sistem.

2.2.2 Pengertian Informasi

Menurut Fatta (2007) “Informasi adalah data yang telah diolah menjadi bentuk yang berarti bagi penerimanya dan bermanfaat dalam pengambilan keputusan saat ini atau mendatang. Dengan demikian informasi berarti data yang telah diproses, atau data yang memiliki arti”.

Sedangkan menurut Jogiyanto (2005) “Informasi diartikan sebagai data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya”.

Kualitas dari suatu informasi tergantung dari 3 hal, yaitu :

1. Informasi harus akurat (*accurate*)

Informasi harus bebas dari kesalahan dan tidak biasa atau menyesatkan. Akurat juga berarti bahwa informasi harus jelas mencerminkan maksudnya.

2. Tepat Waktu (*timeline*)

Informasi yang sampai pada si penerima tidak boleh terlambat. Informasi yang sudah usang tidak akan mempunyai nilai lagi. Karena informasi merupakan landasan didalam pengambilan keputusan.

3. Relevan (*relevance*)

Informasi tersebut mempunyai manfaat untuk pemakainya. Relevansi untuk setiap orang, satu dengan yang lainnya adalah berbeda. (Sutabri, 2005)

2.2.3 Pengertian Penelitian

Penelitian adalah kegiatan yang dilakukan menurut kaidah dan metode ilmiah secara sistematis untuk memperoleh informasi, data, dan keterangan yang berkaitan dengan pemahaman dan pembuktian kebenaran atau ketidakbenaran suatu asumsi dan/atau hipotesis di bidang ilmu pengetahuan dan teknologi serta menarik kesimpulan ilmiah bagi keperluan kemajuan ilmu pengetahuan dan teknologi. (Indonesia, 2002)

2.2.4 Pengetian LPPM

Proses berlangsungnya LPPM pada dasarnya terdiri dari dua lembaga yang berbeda yaitu lembaga penelitian dan lembaga pengabdian, akan tetapi saat ini rata-rata perguruan tinggi telah menyatukan dua lembaga tersebut dengan nama LPPM yang terdiri dari satu Ketua LPPM dengan dibantu oleh Wakil Ketua Bidang Penelitian dan Wakil Ketua Bidang Pengabdian kepada Masyarakat. (Farhan, 2011)

2.2.5 UML (*Unified Modeling Language*)

UML (Unified Modeling Language) adalah sebuah bahasa yang berdasarkan grafik atau gambar untuk memvisualisasi, menspesifikasikan, membangun, dan pendokumentasian dari sebuah sistem pengembangan *software* berbasis OO (*Object-Oriented*). UML sendiri juga memberikan standar penulisan sebuah sistem *blue print*, yang meliputi konsep bisnis proses, penulisan kelas-kelas dalam bahasa program yang spesifik, skema database, dan komponen-komponen yang diperlukan dalam sistem *software*.

Seperti bahasa-bahasa lainnya, UML mendefinisikan notasi dan syntax / semantik. Notasi UML merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai diagram piranti lunak. Setiap bentuk memiliki makna tertentu, dan UML syntax mendefinisikan bagaimana bentuk-bentuk tersebut dapat dikombinasikan. Notasi UML diturunkan dari 3 notasi yang telah ada sebelumnya : Grady Booch - *Object-Oriented Design* (OOD), Jim Rumbaugh - *Object Modeling Technique* (OMT), dan Ivar Jacobson – *Object Oriented Software Engineering* (OOSE). (Mathiassen, 2000)

a. *Use Case Diagram*

Use-case diagram adalah gambaran *graphical* dari beberapa atau semua *actor*, *use-case*, dan interaksi diantara komponen-komponen tersebut yang memperkenalkan suatu sistem yang akan dibangun. *Use case diagram* menjelaskan manfaat suatu sistem jika dilihat menurut pandangan orang yang berada di luar sistem. Diagram ini menunjukkan fungsionalitas suatu sistem atau kelas dan bagaimana sistem tersebut berinteraksi dengan dunia luar.






Use case diagram dapat digunakan selama proses analisis untuk menangkap *requirement* sistem dan untuk memahami bagaimana sistem seharusnya bekerja. Selama tahap desain, *use case diagram* berperan untuk menetapkan perilaku (*behavior*) sistem saat diimplementasikan. Dalam sebuah model mungkin terdapat satu atau beberapa *use case diagram*. Kebutuhan atau *requirements system* adalah fungsionalitas apa yang harus disediakan oleh sistem kemudian didokumentasikan pada model *use case* yang menggambarkan fungsi sistem yang diharapkan (*use case*), dan yang mengelilinginya (*actor*), serta hubungan antara *actor* dengan *use case* (*use case diagram*) itu sendiri.






Cara menentukan *use case* dalam suatu sistem :

1. Pola perilaku perangkat lunak aplikasi
2. Gambaran tugas dari sebuah *actor*
3. Sistem atau “benda” yang memberikan sesuatu yang bernilai kepada *actor*.

4. Apa yang dikerjakan oleh suatu perangkat lunak (“bukan bagaimana cara mengerjakannya”)

Tabel 2.1 Simbol *Use Case Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (<i>independent</i>).
3		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
4		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara <i>eksplisit</i> .
5		<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.

6		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor
9		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
10		<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

b. 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 hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain. *Class* memiliki tiga area pokok :

1. Nama (*stereotype*)
2. Atribut
3. Metoda

Atribut dan metoda dapat memiliki salah satu sifat berikut :



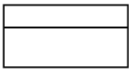


- *Private*, tidak dapat dipanggil dari luar *class* yang bersangkutan
- *Protected*, hanya dapat dipanggil oleh *class* yang bersangkutan dan anak-anak yang mewarisinya
- *Public*, dapat dipanggil oleh siapa saja

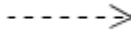

Hubungan Antar *Class*

1. Asosiasi, yaitu hubungan statis antar *class*. Umumnya menggambarkan *class* yang memiliki atribut berupa *class* lain, atau *class* yang harus mengetahui eksistensi *class* lain. Panah *navigability* menunjukkan arah *query* antar *class*.
2. Agregasi, yaitu hubungan yang menyatakan bagian (“terdiri atas”).
3. Pewarisan, yaitu hubungan hirarki antar *class*. *Class* dapat diturunkan dari *class* lain dan mewarisi semua atribut dan metoda *class* asalnya dan menambahkan fungsionalitas baru, sehingga ia disebut anak dari *class* yang diwarisinya. Kebalikan dari pewarisan adalah generalisasi.

4. Hubungan dinamis, yaitu rangkaian pesan (*message*) yang di-*passing* dari satu *class* kepada *class* lain. Hubungan dinamis dapat digambarkan dengan menggunakan *sequence diagram* yang akan dijelaskan kemudian.

Tabel 2.2 Simbol *Class Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
2		<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3		<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4		<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
5		<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.

6		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (independent) akan mempegaruhi elemen yang bergantung padanya elemen yang tidak mandiri
7		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya


c. *Activity Diagram*

Activity diagram menggambarkan berbagai alir 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.

Activity diagram merupakan *state diagram* khusus, di mana sebagian besar *state* adalah *action* dan sebagian besar transisi di *trigger* oleh selesainya *state* sebelumnya (*internal processing*). Oleh karena itu *activity diagram* tidak menggambarkan *behaviour* internal sebuah sistem (interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum. Sebuah aktivitas dapat direalisasikan oleh satu *use case* atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara *use case* menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas. Sama seperti *state*, standar UML menggunakan segiempat dengan sudut membulat untuk menggambarkan aktivitas. *Decision* digunakan untuk menggambarkan *behaviour* pada kondisi tertentu. Untuk mengilustrasikan proses-

proses paralel (*fork* dan *join*) digunakan titik sinkronisasi yang dapat berupa titik, garis horizontal atau vertikal. *Activity diagram* dapat dibagi menjadi beberapa *object swimlane* untuk menggambarkan objek mana yang bertanggung jawab untuk aktivitas tertentu.

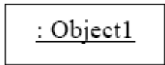

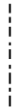

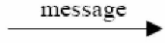
Tabel 2.3 Simbol *Activity Diagram*

No	Simbol	Nama	Keterangan
1.		Initial Activity	sebagai awal dari aktivitas modul sistem aplikasi.
2.		Activity	menunjukkan aktivitas yang dilakukan.
3.		Final Activity	menunjukkan akhir dari aktivitas.
4.		Decisions	menunjukkan aktivitas yang harus dipilih apakah pilihan pertama atau kedua.
5.		Signal	sebagai pengirim dan penerima pesan dari aktivitas yang terjadi. Sinyal terdiri dari sinyal penerima yang digambarkan dengan poligon terbuka dan sinyal pengirim dengan yang digambarkan dengan convex poligon.
6.		Concurrent Activities	menggambarkan atktivitas yang dilakukan bersamaan atau paralel.

d. *Sequence Diagram*

Diagram interaksi yang menekankan pada waktu pengiriman *message*. *Sequence* diagram menunjukkan sekumpulan objek dan pengiriman serta penerimaan *message* antar objek. Objek yang umumnya memiliki nama atau instansiasi dari *class*, tapi dapat pula merupakan turunan dari yang lain, seperti *collaboration*, *component* dan *node*. Diagram ini digunakan untuk mengilustrasikan *dynamic view* dari sistem.

Tabel 2.4 Simbol *Sequence Diagram*

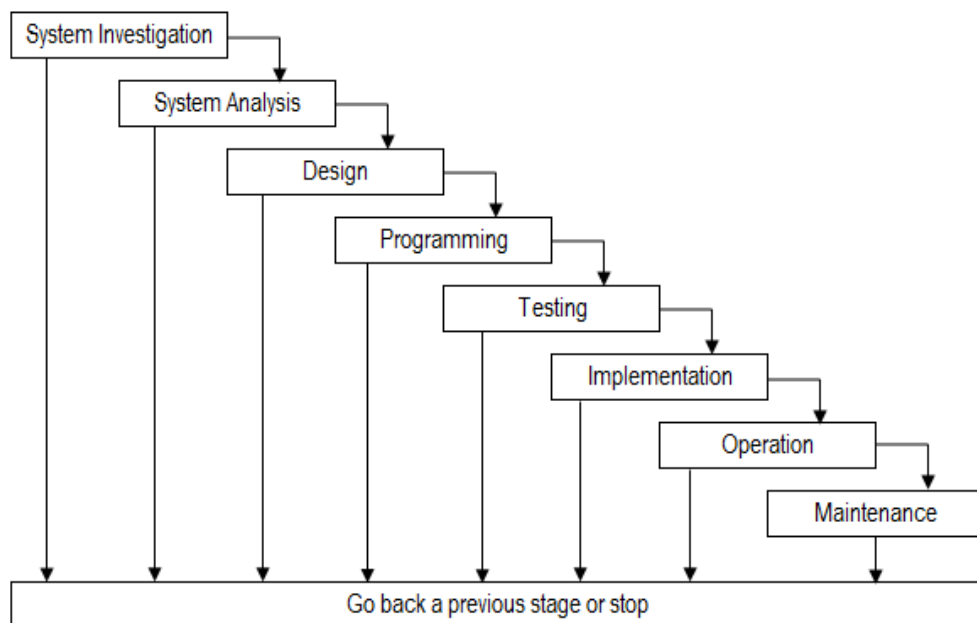
No.	Simbol	Nama	Keterangan
1.		Object	<i>Object</i> merupakan <i>instance</i> dari sebuah <i>class</i> dan dituliskan tersusun secara horizontal.
2.		Actor	<i>Actor</i> juga dapat berkomunikasi dengan <i>object</i> , maka <i>actor</i> juga dapat diurutkan sebagai kolom.
3.		Lifeline	<i>Lifeline</i> mengindikasikan keberadaan sebuah <i>object</i> dalam basis waktu.
4.		Activation	<i>Activation</i> dinotasikan sebagai sebuah kotak segi empat yang digambar pada sebuah <i>lifeline</i> .
5.		Message	<i>Message</i> digambarkan dengan anak panah horizontal antara <i>activation</i> .

2.2.6 Metode Pengembangan Sistem

“Metodologi Pengembangan Sistem adalah metode-metode, prosedur-prosedur, konsep-konsep pekerjaan, aturan-aturan dan postulat-postulat yang akan digunakan untuk mengembangkan suatu sistem informasi”. (Jogiyanto, 2010)

Pengembangan sistem didefinisikan sebagai aktivitas untuk menghasilkan sistem informasi berbasis komputer untuk menyelesaikan persoalan organisasi atau memanfaatkan kesempatan yang timbul.

Model air terjun (*waterfall*) biasa juga disebut *Software Development Life Cycle* (SDLC) merupakan siklus yang menggambarkan perangkat lunak yang dibangun. Contoh penerapan *SDLC* adalah metode *Waterfall* dengan tahapan-tahapan sebagai berikut :



Gambar 2.1 Langkah-langkah metode *waterfall*

Secara garis besar metode *waterfall* mempunyai langkah-langkah sebagai berikut :

1. *System Investigation*

Pada tahap ini, dilakukan studi kelayakan untuk menentukan kemungkinan suksesnya proyek pengembangan sistem yang diajukan dan menaksir kelayakan proyek dari segi teknis, ekonomis dan perilaku.

2. *System Analysis*

Software merupakan bagian dari sebuah sistem yang besar, maka pengerjaan dimulai dengan mengumpulkan kebutuhan bagi semua elemen-elemen sistem kemudian mengalokasikan beberapa subset dari kebutuhan-kebutuhan tersebut ke *software*. Hal ini sangat penting ketika *software* harus berhubungan dengan elemen lain seperti *hardware*, manusia dan basis data. Tahap ini meliputi pengumpulan kebutuhan pada tingkat sistem dengan sedikit analisa dan perancangan ditingkat atas.

Proses pengumpulan elemen sistem ditingkatkan dan dipusatkan secara khusus pada *software* untuk mengerti karakteristik dari program yang akan dibuat. Sistem analisis *software* harus mengerti ruang lingkup informasi yang ingin dicakup dalam pembuatan *software* seperti fungsi-fungsi yang dibutuhkan, karakteristik, kinerja dan tampilan.

3. *Design*

Adalah proses bertahap yang berfokus pada 4 atribut program yang berbeda yaitu struktur data, arsitektur perangkat lunak, implementasi kebutuhan dan detail *procedural* (algoritma). Proses perancangan menerjemahkan kebutuhan elemen sistem yang direpresentasikan kedalam suatu *software* yang diperkirakan kualitasnya sebelum dilakukan pengkodean.

4. *Programming*

Pada tahap ini rancangan diterjemahkan kedalam bentuk yang dapat dibaca oleh mesin jika perancangan dilaksanakan secara detail. Pengkodean dapat dilakukan secara mekanis.

5. *Testing*

Jika kode telah dibuat atau ditulis maka diadakan pengujian program. Pengujian juga dilakukan untuk memastikan masukan yang diberikan menghasilkan keluaran yang diinginkan.

6. *Implementation and Operation*

Setelah konversi, sistem baru akan beroperasi dalam periode tertentu. Saat operasi sistem baru telah stabil, audit dilakukan guna menaksir kapabilitas sistem dan menentukan apakah penggunaanya tepat.

7. *Maintenance*

Software yang telah dilaksanakan secara tidak langsung akan mengalami perbaikan hal ini disebabkan masih ditemukan adanya perubahan di dalam lingkungan ekstrenalnya (seperti perubahan yang disebabkan oleh sistem operasi atau peralatan baru). Dengan adanya pemeliharaan *software*, perubahan yang terjadi akan lebih mudah dilakukan dibandingkan harus membuat ulang program baru.

2.2.7 PHP (Hypertext Preprocessor)

PHP adalah bahasa server-side –scripting yang menyatu dengan HTML untuk membuat halaman web yang dinamis. Karena PHP merupakan server-side-scripting maka sintaks dan perintah-perintah PHP akan dieksekusi di server kemudian hasilnya akan dikirimkan ke browser dengan format HTML. Dengan demikian kode program yang ditulis dalam PHP tidak akan terlihat oleh user sehingga keamanan halaman web lebih terjamin. PHP dirancang untuk membuat

halaman web yang dinamis, yaitu halaman web yang dapat membentuk suatu tampilan berdasarkan permintaan terkini, seperti menampilkan isi basis data ke halaman web. (Arief, 2011)

PHP atau singkatan dari Personal Home Page merupakan bahasa skrip yang tertanam dalam HTML untuk dieksekusi bersifat server side". PHP termasuk dalam open source product, sehingga source code PHP dapat diubah dan didistribusikan secara bebas. Versi terbaru PHP dapat diunduh secara gratis melalui situs resmi PHP : <http://www.php.net>. PHP juga dapat berjalan pada berbagai web server seperti IIS (Internet Information Server), PWS (Personal Web Server), Apache, Xitami. PHP juga mampu berjalan di banyak sistem operasi yang beredar saat ini, diantaranya : Sistem Operasi Microsoft Windows (semua versi), Linux, Mac Os, Solaris. PHP dapat dibangun sebagai modul web server Apache dan sebagai binary yang dapat berjalan sebagai CGI (Common Gateway Interface). PHP dapat mengirim HTTP header, dapat mengatur cookies , mengatur authentication dan redirect user. (Nugroho, 2006)

Salah satu keunggulan yang dimiliki PHP adalah kemampuannya untuk melakukan koneksi ke berbagai macam software sistem manajemen basis data atau *Database Management Sistem* (DBMS), sehingga dapat menciptakan suatu halaman *web* dinamis. PHP mempunyai koneksitas yang baik dengan beberapa DBMS seperti Oracle, Sybase, mSQL, MySQL, Microsoft SQL Server, Solid, PostgreSQL, Adabas, FilePro, Velocis, dBase, Unix dbm, dan tidak terkecuali semua database ber-interface ODBC. Hampir seluruh aplikasi berbasis *web* dapat dibuat dengan PHP. Namun kekuatan utama adalah konektivitas basis data dengan *web*. Dengan kemampuan ini kita akan mempunyai suatu sistem basis data yang dapat diakses.

2.2.8 Pengertian Web atau Situs

Website atau situs dapat diartikan sebagai kumpulan halaman halaman yang digunakan untuk menampilkan informasi, teks, gambar diam atau bergerak,

animasi, suara, dan atau gabungan dari semuanya itu, baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling berkait dimana masing masing dihubungkan dengan jaringan jaringan halaman (*hyperlink*). (Utama, 2011)

2.2.9 MYSQL

“MySQL adalah salah satu jenis database server yang sangat terkenal dan banyak digunakan untuk membangun aplikasi web yang menggunakan database sebagai sumber dan pengolahan datanya”. (Arief, 2011)

MySQL dikembangkan oleh perusahaan swedia bernama MySQL AB yang pada saat ini bernama Tcx DataKonsult AB sekitar tahun 1994-1995, namun cikal bakal kodenya sudah ada sejak tahun 1979. Awalnya Tcx merupakan perusahaan pengembang software dan konsultan database, dan saat ini MySQL sudah diambil alih oleh Oracle Corp. Kepopuleran MySQL antara lain karena MySQL menggunakan SQL sebagai bahasa dasar untuk mengakses databasenya sehingga mudah untuk digunakan, kinerja query cepat, dan mencukupi untuk kebutuhan database perusahaan-perusahaan yang berskala kecil sampai menengah, MySQL juga bersifat open source (tidak berbayar) .

MySQL merupakan database yang pertama kali didukung oleh bahasa pemrograman script untuk internet (PHP dan Perl). MySQL dan PHP dianggap sebagai pasangan software pembangun aplikasi web yang ideal. MySQL lebih sering digunakan untuk membangun aplikasi berbasis web, umumnya pengembangan aplikasinya menggunakan bahasa pemrograman script PHP.

MySQL didistribusikan dengan licensi open source GPL (General Public License) mulai versi 3.23 pada bulan juni 2000. Software MySQL bisa diunduh melalui website resminya di <http://www.MySQL.org> atau di <http://www.mysql.com>.

2.2.10 Metode Simple Additive Weighting (SAW)

Metode *Simple Additive Weighting* (SAW) sering juga dikenal istilah metode penjumlahan terbobot. Konsep dasar metode SAW adalah mencari penjumlahan

terbobot dari rating kinerja pada setiap alternatif pada semua atribut (Fishburn, 1967) dan (MacCrimmon, 1968). Metode SAW membutuhkan proses normalisasi matriks keputusan (X) ke suatu skala yang dapat diperbandingkan dengan semua rating alternatif yang ada. Metode ini merupakan metode yang paling terkenal dan paling banyak digunakan dalam menghadapi situasi *Multiple Attribute Decision Making* (MADM). MADM itu sendiri merupakan suatu metode yang digunakan untuk mencari alternatif optimal dari sejumlah alternatif dengan kriteria tertentu. Metode SAW ini mengharuskan pembuat keputusan menentukan bobot bagi setiap atribut. Skor total untuk alternatif diperoleh dengan menjumlahkan seluruh hasil perkalian antara rating (yang dapat dibandingkan lintas atribut) dan bobot tiap atribut. Rating tiap atribut haruslah bebas dimensi dalam arti telah melewati proses normalisasi matriks sebelumnya.

Kelebihan dari metode SAW, yaitu menemukan nilai bobot untuk setiap atribut, kemudian dilanjutkan dengan proses perankingan yang akan menyeleksi alternatif terbaik dari sejumlah alternatif, penilaian akan lebih tepat karena didasarkan pada nilai kriteria dari bobot preferensi yang sudah ditemukan, dan adanya perhitungan normalisasi matriks sesuai dengan nilai atribut (antara nilai *benefit* dan *cost*). Sedangkan kekurangan dari metode SAW yaitu digunakan pada pembobotan lokal dan perhitungan dilakukan dengan menggunakan bilangan crisp dan *fuzzy*.

Berikut ini akan dijelaskan langkah penyelesaian *menggunakan metode Simple Additive Weighting* :

- a. Menemukan kriteria-kriteria yang akan dijadikan acuan dalam pengambilan keputusan, yaitu C_i .
- b. Menentukan rating kecocokan setiap kriteria.
- c. Membuat matriks keputusan berdasarkan kriteria (C_i), kemudian melakukan normalisasi matriks berdasarkan persamaan yang disesuaikan dengan jenis atribut (atribut keuntungan ataupun atribut biaya) sehingga diperoleh matriks normalisasi R .

d. Hasil akhir diperoleh dari proses perankingan yaitu penjumlahan dari perkalian matriks ternormalisasi R dengan vektor bobot sehingga diperoleh nilai terbesar yang dipilih sebagai alternatif terbaik (A_i) sebagai solusi.

Rumus untuk melakukan normalisasi tersebut adalah :

$$r_{ij} = \begin{cases} \frac{X_{ij}}{\text{Max}_{ij}} & \text{jika } j \text{ adalah atribut keuntungan (benefit)} \\ \frac{\text{Min}_{ij}}{X_{ij}} & \text{jika } j \text{ adalah atribut keuntungan (cost)} \end{cases} \quad (1)$$

Dimana :

r_{ij} = rating kinerja ternormalisasi

Max_{ij} = nilai maksimum dari setiap bisnis dan kolom

X_{ij} = baris dari kolom dari matriks

Dengan r_{ij} adalah rating kinerja ternormalisasi dari alternatif A_i pada atribut A_i pada atribut C_j ; $i = 1, 2, \dots, m$ dan $j = 1, 2, \dots, n$.

Nilai preferensi untuk setiap alternatif (V_i) diberikan sebagai :

$$V_i = \sum_{j=1}^n W_{ij} r_{ij} \quad (2)$$

Dimana :

V_i = Nilai akhir dari alternatif

W_j = Bobot yang telah ditentukan

r_{ij} = Normalisasi matriks

Nilai V_i yang lebih besar mengindikasikan bahwa alternatif A_i lebih terpilih.

2.2.11 Metode Pengujian Perangkat Lunak

a. Black Box Testing

Black box testing adalah metode pengujian perangkat lunak yang menguji fungsionalitas dari aplikasi yang bertentangan dengan struktur internal atau kerja. “Pada *black box testing*, cara pengujian hanya dilakukan dengan menjalankan atau mengeksekusi unit atau modul, kemudian diamati apakah hasil dari unit itu sesuai dengan proses yang diinginkan”. (Fatta, 2007)

Metode *black box testing* memfokuskan pada keperluan fungsionalitas dari *software*. Karena *black box testing* memungkinkan pengembang *software* untuk membuat himpunan kondisi *input* yang akan melatih seluruh syarat-syarat fungsionalitas suatu program. Metode *black box testing* bukan merupakan alternatif dari metode *white box testing*, tetapi merupakan pendekatan yang melengkapi untuk menemukan kesalahan lainnya, selain menggunakan *white box testing*.

Black box testing berusaha untuk menemukan kesalahan dalam beberapa kategori, diantaranya :

1. Fungsi-fungsi yang salah atau hilang.
2. Kesalahan *interface*.
3. Kesalahan dalam struktur data atau akses *database* eksternal.
4. Kesalahan insialisasi dan terminasi.

2.2.12 User Acceptance Testing

User Acceptance Testing merupakan pengujian yang dilakukan oleh *end-user* dimana *user* tersebut adalah *staff*/karyawan perusahaan yang langsung berinteraksi dengan sistem dan dilakukan verifikasi apakah fungsi yang ada telah berjalan sesuai dengan kebutuhan/fungsinya. (Perry, 2006)

Setelah dilakukan *system testing*, *acceptance testing* menyatakan bahwa sistem *software* memenuhi persyaratan. *Acceptance testing* merupakan pengujian yang dilakukan oleh pengguna yang menggunakan teknik pengujian *black box* untuk menguji sistem terhadap spesifikasinya. Pengguna akhir bertanggung jawab untuk memastikan semua fungsionalitas yang relevan telah diuji. (Lewis, 2009)

Dari definisi di atas, *user acceptance testing* adalah pengujian yang dilakukan oleh pengguna dari sistem tersebut untuk memastikan fungsi-fungsi yang ada pada sistem tersebut telah berjalan dengan baik dan sesuai dengan kebutuhan pengguna.

2.2.13 Basis Data

Basis data merupakan kumpulan data yang disatukan dalam suatu organisasi. Tujuan dari penggunaan *database* adalah untuk menyimpan semua data yang diinginkan pada suatu lokasi, sehingga penyimpanan data *redundant* di dalam organisasi tersebut dapat dieliminasi. (Suryadi, 2001)

Jika ditinjau dari segi bahasa, basis data terdiri dari dua kata, yaitu basis dan data, yang dapat didefinisikan dalam sejumlah sudut pandang seperti :

1. Himpunan kelompok data (Arsip) yang saling berhubungan dan diorganisasikan sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah.
2. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (*redundansi*) yang tidak perlu.
3. Kumpulan file/ table/ arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronik.

a. Tujuan Basis Data

Penggunaan sebuah basis data memiliki berbagai macam tujuan, adapun tujuan-tujuan tersebut meliputi :

1. Kecepatan dan kemudahan (*Speed*).
2. Efisiensi ruang penyimpanan (*Space*).
3. Keakuratan (*Accuracy*).
4. Ketersediaan (*Availability*).
5. Kelengkapan (*Completeness*).
6. Keamanan (*Security*).
7. Kebersamaan (*Shareability*).

b. Kegunaan Basis Data

Penyusunan satu basis data digunakan untuk mengatasi masalah-masalah pada penyusunan data yang dialami, yaitu :

1. Redudansi dan inkonsistensi data.
2. Kesulitan pengaksesan data.
3. Isolasi data dan standarisasi.
4. *Multiuser* (Banyak pemakai).
5. Masalah keamanan (*Security*).
6. Masalah integrasi.

2.2.14 Analisa Data Kuantitatif

Analisa data adalah proses menyederhanakan data kedalam bentuk yang lebih mudah dibaca dan diinterpretasikan. dalam proses ini seringkali digunakan statistik. Statistik disini berfungsi menyederhanakan data penelitian yang amat besar jumlahnya menjadi informasi yang lebih sederhana dan lebih mudah dipahami. (Sinarimbun, 1989)

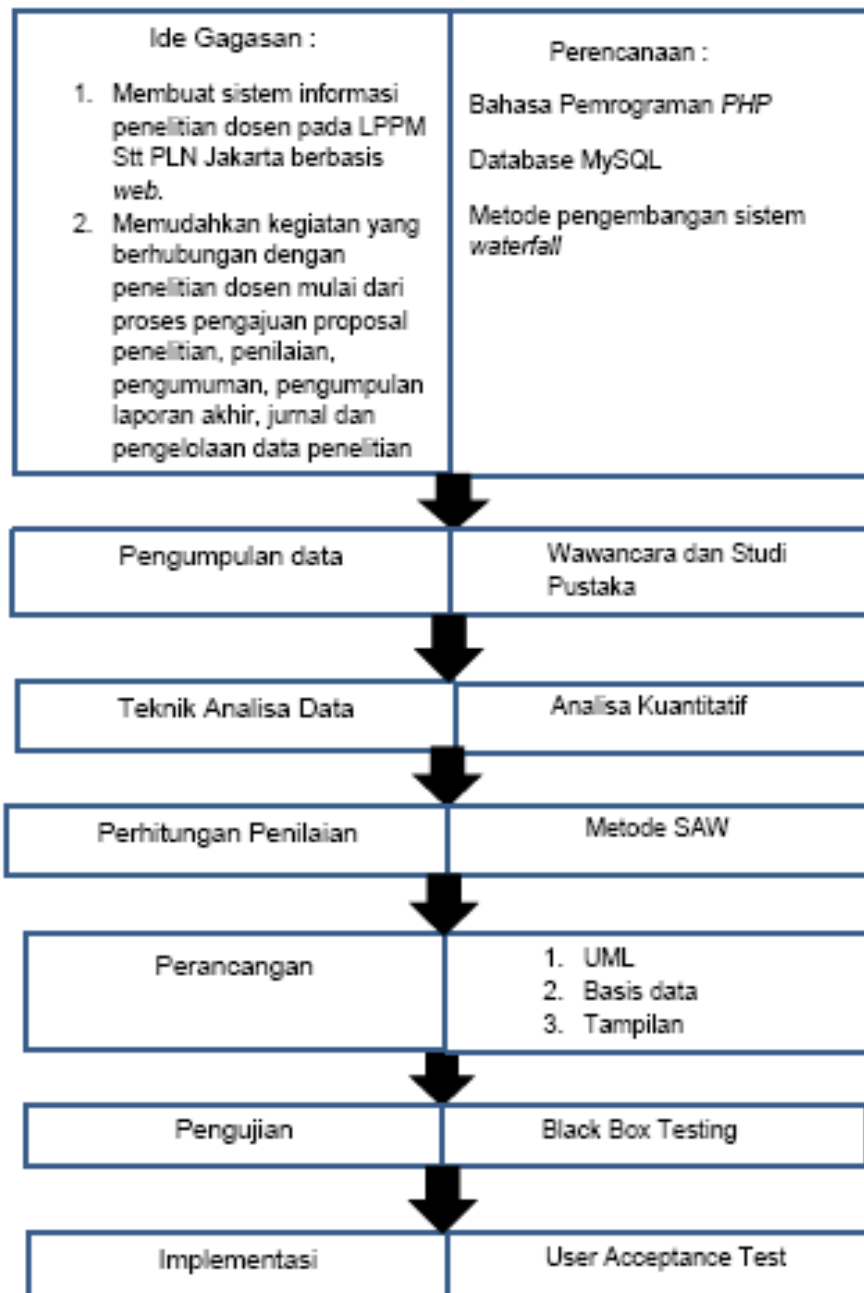
Sedangkan data kuantitatif yaitu data dalam bentuk jumlah dituangkan untuk menerangkan suatu kejelasan dari angka-angka atau memperbandingkan

dari beberapa gambaran sehingga memperoleh gambaran baru, kemudian dijelaskan kembali dalam bentuk kalimat/ uraian. (Subagyo, 2004)

Dari paparan diatas dapat disimpulkan bahwa metode analisis data merupakan cara untuk menganalisa hasil dari data yang diperoleh dalam penelitian sehingga lebih mudah untuk dibaca dan diinterpretasikan. Analisis data ini dilakukan setelah terkumpulnya semua data hasil penelitian. Adapun cara yang ditempuh dalam rangka menganalisis data kuantitatif ini dengan menggunakan metode statistik.

2.2 Kerangka Pemikiran

Berikut ini merupakan penjelasan dari kerangka pemikiran yang penulis lakukan. Kerangka pemikiran tersebut mencakup seluruh tahap dari awal pembuatan hingga akhir penyusunan.



Gambar 2.2 Kerangka Pemikiran

Penjelasan tahapan kerangka pemikiran sebagai berikut :

1. Ide Gagasan

Ide gagasan penulis yaitu membuat suatu sistem informasi penelitian dosen pada LPPM STT-PLN berbasis *web*. Bertujuan untuk memudahkan kegiatan yang

berhubungan dengan penelitian dosen mulai dari proses pengajuan proposal penelitian, penilaian, pengumuman, pengumpulan laporan akhir, pengumpulan luaran hasil penelitian yang dapat berupa HAKI, jurnal maupun *proceeding* dan pengelolaan data penelitian. Dalam perencanaan sistem informasi ini dibuat dengan menggunakan bahasa pemrograman *PHP* dengan *database MySQL*. Metode pengembangan sistem yang digunakan adalah metode *Waterfall*.

2. Pengumpulan Data

Pengumpulan data dilakukan untuk memperoleh teori-teori dasar yang dibutuhkan, sehingga dapat digunakan sebagai acuan dalam menyelesaikan penelitian ini. Teknik pengumpulan data yang digunakan oleh penulis adalah metode wawancara dan studi pustaka. Metode wawancara merupakan metode pengumpulan data melalui tatap muka secara langsung dengan pihak-pihak tertentu, dalam hal ini adalah LPPM STT-PLN. Sedangkan metode studi pustaka yang dilakukan penulis adalah dengan cara membaca dan memahami terhadap buku, artikel maupun bahan kepustakaan yang berhubungan dengan sistem informasi penelitian dosen berbasis *web*.

3. Teknik Analisa Data

Analisa data digunakan untuk mengolah data yang didapat dalam penelitian menjadi informasi. Teknik analisa data yang digunakan oleh penulis adalah analisa kuantitatif.

4. Perhitungan Penilaian

Pada tahap ini penulis menerapkan metode *SAW (Simple Additive Weighting)* dalam penetapan pemenang apabila terdapat nilai yang sama pada dua atau lebih proposal penelitian yang diajukan setelah dilakukan proses *deskevaluasi* namun kuota proposal yang diterima terbatas. Dengan menggunakan metode *SAW* ini, kriteria dan bobot penilaian harus ditetapkan. Pada penerapan perhitungan menggunakan metode ini peran admin LPPM dibutuhkan untuk

memberi skor sesuai dengan kriteria yang sudah ditetapkan. Berikut ini merupakan kriteria-kriteria beserta *prosentase* bobot masing-masing yang telah ditetapkan, yaitu :

- a. Tanggal pengusulan : 30 %
- b. Biaya yang diusulkan : 25 %
- c. Status dosen (pemula/lama) : 20 %
- d. Kemutakhiran topik yang diusulkan : 15 %
- e. Format penulisan : 10 %

5. Perancangan

Tahap perancangan meliputi perancangan *UML*, perancangan *database*, perancangan tampilan dan pengkodean. Perancangan *UML* adalah tahap mendeskripsikan dan mendesain sistem perangkat lunak dalam bentuk diagram. Setelah perancangan *UML* selesai dilakukan maka akan dilakukan tahap pembuatan *database* dengan mengacu kepada *UML* yang sudah dibuat. Perancangan tampilan adalah tahap merancang tampilan (*interface*) untuk aplikasi yang akan dibuat. Setelah rancangan tampilan telah selesai dibuat, maka akan dilakukan tahap pengkodean.

6. Pengujian

Tahap pengujian atau biasa juga disebut dengan tahap uji coba bertujuan untuk mengetahui apakah sistem sudah berkerja dengan baik dan sempurna atau tidak. Pada tahap ini, penulis menggunakan metode *black box testing*. Sehingga, uji coba difokuskan pada keperluan fungsionalitas dari *software*, apakah *output* yang keluar sudah sesuai dengan *input*.

7. Implementasi

Apabila tidak ada kesalahan dalam program aplikasi tersebut saat dilakukan pengujian, maka aplikasi dinyatakan berhasil. Sehingga aplikasi tersebut sudah dapat diterapkan. Pada tahap implementasi ini penulis menggunakan metode *User*

Acceptance Test untuk mengetahui apakah sistem yang dibuat sudah sesuai dengan kebutuhan.