

BAB II

LANDASAN TEORI

Bab ini menjelaskan dasar teori yang menunjang penulisan tugas akhir mengenai Sistem Aplikasi Report Borang Akreditasi Standar VII berbasis web menggunakan PHP dan MySQL. Bab ini akan dibahas tinjauan pustaka mengenai Sistem Penilaian Borang Akreditasi dan Aplikasi Database Berbasis Web (PHP, CSS, MySQL & Bluefish Editor) dan UML.

2.1 Sistem

Pada sub bab ini akan dijelaskan mengenai pengertian sistem, karakteristik, perancangan dan pengembangan dari suatu sistem.

2.1.1 Pengertian Sistem

Seperangkat elemen yang digabungkan satu dengan lainnya untuk suatu tujuan bersama. (Murdick dan Ross, 1993)

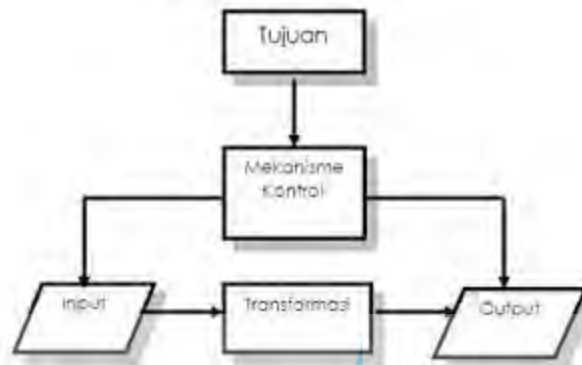
Sistem terdiri dari unsur-unsur seperti masukan (*input*), pengolahan (*proccessing*) serta keluaran (*output*). Ciri pokok sistem ada empat, yaitu sistem yang beroperasi dalam suatu lingkungan, memiliki unsur-unsur, saling berhubungan dan mempunyai satu fungsi atau tujuan utama. (Scott, 1996)

Berikut susunan model sistem dapat dilihat pada **Gambar 2.1** dibawah ini:



Gambar 2.1 Model sistem (Scott, 1996)

Sementara *Mc. Leod* (*Mc. Leod, 1995*) mendefinisikan sistem sebagai sekelompok elemen-elemen yang terintegrasi dengan maksud yang sama untuk mencapai suatu tujuan. Sumber daya mengalir dari elemen input dan untuk menjamin prosesnya berjalan dengan baik maka dihubungkan dengan mekanisme kontrol. Untuk lebih jelasnya elemen sistem tersebut dapat dilihat pada **Gambar 2.2** berikut :



Gambar 2.2 Model hubungan elemen-elemen sistem (Mc. Leod, 1995)

2.1.2 Karakteristik Sistem

Karakteristik sistem adalah sistem yang mempunyai komponen-komponen, batas sistem, lingkungan sistem, penghubung, masukan, keluaran, pengolah, dan sasaran.

1. Komponen-komponen (*components*)

Setiap sistem baik sistem dalam skala besar maupun sistem dalam skala kecil sekalipun memiliki komponen-komponen atau elemen-elemen. Komponen-komponen ini saling berhubungan dan bekerja sama sehingga tercipta satu kesatuan fungsi dari sistem. Sehingga sistem dapat mencapai tujuannya.

2. Penghubung Sistem (*System Interface*)

Penghubung sistem merupakan media perantara antara subsistem yang satu dengan subsistem yang lainnya. Melalui penghubung sistem ini, maka subsistem-subsistem dapat saling memberi dan menerima sumber daya sehingga terjalin kerja sama dan dapat membentuk satu kesatuan fungsi dari sistem.

3. Lingkungan Luar (*Environment*)

Lingkungan luar dari sistem adalah segala sesuatu yang berada di luar batas sistem. Lingkungan luar ini bisa juga berupa ekosistem dimana sistem tersebut berada. Walaupun keberadaannya diluar sistem, tapi lingkungan luar dapat mempengaruhi sistem. Adanya ketidakserasian antara lingkungan luar dengan sistem dapat menyebabkan terganggunya fungsi sistem. Oleh karena itu harus senantiasa tercipta keharmonisan antara sistem dengan lingkungan luarnya.

4. Batas Sistem (*Boundary*)

Batas sistem merupakan daerah pemisah antara satu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batas sistem ini memberikan ruang lingkup yang jelas dari suatu sistem. Dengan adanya ruang lingkup yang jelas dari sistem tersebut, maka kita dapat memisahkan dan membedakan satu sistem dengan sistem yang lainnya maupun sistem dengan lingkungan luar.

5. Masukan Sistem (*Sistem Input*)

Masukan adalah segala sesuatu yang masuk ke dalam sistem dan selanjutnya menjadi bahan untuk diproses. Masukkan ini dapat berupa hal-hal berwujud (tampak secara fisik) maupun yang tidak tampak.

6. Keluaran Sistem (*Sistem Output*)

Keluaran merupakan hasil dari pengolahan suatu sistem. Keluaran ini tentunya diharapkan dapat berguna sesuai dengan tujuan dari sistem. Selain sebagai hasil akhir, sebagian keluaran bisa juga dijadikan masukan untuk sistem lainnya.

7. Pengolah Sistem (*Sistem Processing*)

Pengolah sistem adalah mesin atau mekanisme yang digunakan untuk mengubah masukan menjadi keluaran. Pengolah memiliki peranan yang penting, karena disinilah proses perubahan dan pendayagunaan masukan terjadi sehingga menghasilkan keluaran yang sesuai dengan tujuan sistem.

8. Sasaran dan Tujuan

Suatu sistem pasti mempunyai tujuan (*goal*) atau sasara (*objective*). Tujuan merupakan hal akhir yang ingin dicapai oleh suatu sistem. Tujuan inilah yang menjadi pemotivasi yang mengarahkan sistem. Tanpa tujuan, sistem menjadi tak terarah dan tak terkendali.

2.1.3 Perancangan Sistem Perangkat Lunak

Menurut Pressman (2002), model waterfall adalah model klasik yang bersifat sistematis, berurutan dalam membangun software. Nama model ini sebenarnya adalah “Linear Sequential Model”. Model ini sering disebut dengan “*classic life cycle*” atau model waterfall. Model ini termasuk kedalam model generic pada rekayasa perangkat lunak dan pertama kali diperkenalkan oleh Winston Royce sekitar tahun 1970 sehingga sering dianggap kuno, tetapi merupakan model yang paling banyak dipakai didalam *Software Engineering* (SE). Model ini melakukan pendekatan secara sistematis dan berurutan. Disebut dengan *waterfall* karena tahap demi tahap yang dilalui harus menunggu selesainya tahap sebelumnya dan berjalan berurutan.

Waterfall adalah suatu metodologi pengembangan perangkat lunak yang mengusulkan pendekatan kepada perangkat lunak sistematis dan sekuensial yang mulai pada tingkat kemajuan sistem pada seluruh analisis, design, kode, pengujian dan pemeliharaan. Langkah-langkah yang harus dilakukan pada metodologi Waterfall adalah sebagai berikut :

A. Analisis kebutuhan perangkat lunak

Proses pengumpulan kebutuhan diintensifkan dan difokuskan, khususnya pada perangkat lunak. Untuk memahami sifat program yang dibangun, rekayasa perangkat lunak (analisis) harus memahami domain informasi, tingkah laku, unjuk kerja dan antar muka (interface) yang diperlukan. Kebutuhan baik untuk sistem maupun perangkat lunak di dokumentasikan dan dilihat dengan pelanggan.

Mengumpulkan kebutuhan secara lengkap kemudian dianalisis dan didefinisikan kebutuhan yang harus dipenuhi oleh software yang akan dibangun. Hal ini sangat penting, mengingat software harus dapat berinteraksi dengan elemen-elemen yang lain seperti hardware, database, dsb. Tahap ini sering disebut dengan *Project Definition*.

B. Desain

Desain perangkat lunak sebenarnya adalah proses multi langka yang berfokus pada empat atribut sebuah program yang berbeda; struktur data, asitektur perangkat lunak, representasi interface dan detail (algoritma) prosedural. Proses desain menerjemahkan syarat/kebutuhan kedalam sebuah representasi perangkat lunak yang dapat di perkirakan demi kualitas sebelum dimulai pemunculan kode. Sebagaimana persyaratan, desain didokumentasikan dan menjadi bagian dari konfigurasi perangkat lunak.

Proses pencarian kebutuhan diintensifkan dan difokuskan pada software. Untuk mengetahui sifat dari program yang akan dibuat, maka para software engineer harus mengerti tentang domain informasi dari software, misalnya fungsi yang dibutuhkan, user interface, dsb. Dari dua aktivitas tersebut (pencarian kebutuhan sistem dan software) harus didokumentasikan dan ditunjukkan kepada user. Proses software design untuk mengubah kebutuhan-kebutuhan di atas menjadi representasi ke dalam bentuk "blueprint" software sebelum coding dimulai. Desain harus dapat mengimplementasikan kebutuhan yang telah disebutkan pada tahap sebelumnya. seperti dua aktivitas sebelumnya, maka proses ini juga harus didokumentasikan sebagai konfigurasi dari software.

C. Generasi Kode

Desain harus diterjemahkan dalam bentuk mesin yang bisa di baca. Langkah pembuatan kode melakukan tugas ini. Jika desain dilakukan dengan cara yang lengkap, pembuatan kode dapat diselesaikan secara mekanis.

Untuk dapat dimengerti oleh mesin, dalam hal ini adalah komputer, maka desain tadi harus diubah bentuknya menjadi bentuk yang dapat dimengerti oleh mesin, yaitu ke dalam bahasa pemrograman melalui proses *coding*. Tahap ini

merupakan implementasi dari tahap design yang secara teknis nantinya dikerjakan oleh *programmer*.

D. Pengujian

Proses Pengujian dilakukan pada logika internal untuk memastikan semua pernyataan sudah diuji. Pengujian eksternal fungsional untuk menemukan kesalahan-kesalahan dan memastikan bahwa input akan memberikan hasil yang aktual sesuai yang dibutuhkan.

E. Pemeliharaan

Perangkat lunak yang sudah disampaikan kepada pelanggan pasti akan mengalami perubahan. Perubahan tersebut bisa karena mengalami kesalahan karena perangkat lunak harus menyesuaikan dengan lingkungan (peripheral atau sistem operasi baru) baru, atau karena pelanggan membutuhkan perkembangan fungsional atau unjuk kerja.

Sesuatu yang dibuat haruslah diujicobakan. demikian juga dengan *software*. Semua fungsi-fungsi *software* harus diujicobakan, agar *software* bebas dari error, dan hasilnya harus benar-benar sesuai dengan kebutuhan yang sudah didefinisikan sebelumnya.

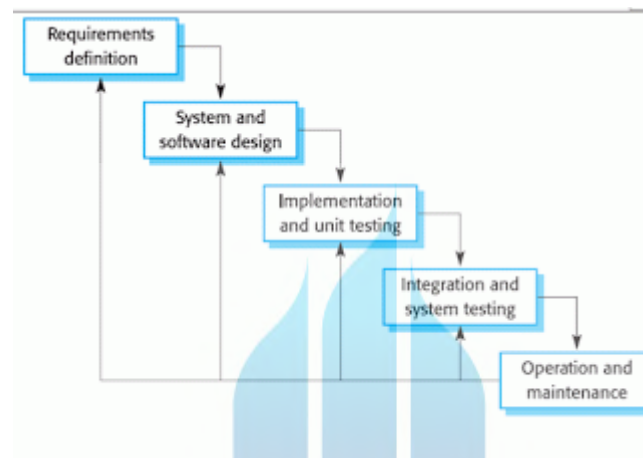
Pemeliharaan suatu *software* diperlukan, termasuk di dalamnya adalah pengembangan, karena *software* yang dibuat tidak selamanya hanya seperti itu ketika dijalankan mungkin saja masih ada error kecil yang tidak ditemukan sebelumnya atau ada penambahan fitur-fitur yang belum ada pada *software* tersebut.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

Kelebihan dari model ini adalah selain karena pengaplikasian menggunakan model ini mudah, kelebihan model ini adalah ketika semua kebutuhan sistem dapat didefinisikan secara utuh, eksplisit, dan benar di awal proyek, maka *Software Engineering*(SE) dapat berjalan dengan baik dan tanpa masalah meskipun seringkali kebutuhan sistem tidak dapat didefinisikan se-eksplisit yang diinginkan, tetapi paling tidak, problem pada kebutuhan sistem di awal proyek lebih ekonomis dalam hal uang (lebih murah), usaha, dan waktu

yang terbuang lebih sedikit jika dibandingkan problem yang muncul pada tahap-tahap selanjutnya.

Kekurangan yang utama dari model ini adalah kesulitan dalam mengakomodasi perubahan setelah proses dijalani. Fase sebelumnya harus lengkap dan selesai sebelum mengerjakan fase berikutnya.

Fase-fase dalam model *waterfall* menurut referensi Sommerfille :



Gambar 2.3 Metode Waterfall Sommerfille

2.2 Penilaian Borang Akreditasi

Pada sub bab ini akan dijelaskan mengenai akreditasi, standar akreditasi dan borang penilaian akreditasi.

2.2.1 Akreditasi

Akreditasi merupakan salah satu bentuk penilaian mutu dan kelayakan institusi yang dilakukan oleh organisasi atau badan mandiri di luar institusi itu. Bentuk penilaian mutu eksternal yang lain adalah penilaian yang berkaitan dengan akuntabilitas, pemberian izin, pemberian lisensi oleh badan tertentu. Ada juga pengumpulan data oleh badan pemerintah bagi tujuan tertentu, dan survei untuk menentukan peringkat (*ranking*) institusi itu.

BAN-PT adalah lembaga yang memiliki kewenangan untuk mengevaluasi dan menilai, serta menetapkan status dan peringkat mutu institusi berdasarkan standar mutu yang telah ditetapkan.

2.2.2 Akreditasi Program Studi Sarjana

Akreditasi program studi sarjana adalah proses evaluasi dan penilaian secara komprehensif atas komitmen program studi terhadap mutu dan kapasitas penyelenggaraan program tridarma perguruan tinggi, untuk menentukan kelayakan program akademiknya. Evaluasi dan penilaian dalam rangka akreditasi program studi dilakukan oleh tim asesor yang terdiri atas pakar sejawat dan/atau pakar yang memahami penyelenggaraan program akademik program studi. Keputusan mengenai mutu didasarkan pada evaluasi dan penilaian terhadap berbagai bukti yang terkait dengan standar yang ditetapkan dan berdasarkan nalar dan pertimbangan para pakar sejawat. Bukti-bukti yang diperlukan termasuk laporan tertulis yang disiapkan oleh program studi yang diakreditasi, diverifikasi dan divalidasi melalui kunjungan atau asesmen lapangan tim asesor ke lokasi program studi.

Dengan demikian, tujuan dan manfaat akreditasi program studi adalah sebagai berikut :

1. Memberikan jaminan bahwa program studi yang terakreditasi telah memenuhi standar mutu yang ditetapkan oleh BAN-PT dengan merujuk pada standar nasional pendidikan yang termaktub dalam Peraturan Pemerintah No. 19 tahun 2005 tentang Standar Nasional Pendidikan, sehingga mampu memberikan perlindungan bagi masyarakat dari penyelenggaraan program studi yang tidak memenuhi standar yang ditetapkan itu.
2. Mendorong program studi untuk terus menerus melakukan perbaikan dan mempertahankan mutu yang tinggi.
3. Hasil akreditasi dapat dimanfaatkan sebagai dasar pertimbangan dalam transfer kredit perguruan tinggi, pemberian bantuan dan alokasi dana, serta pengakuan dari badan atau instansi yang lain.

Mutu program studi merupakan cerminan dari totalitas keadaan dan karakteristik masukan, proses, keluaran, hasil, dan dampak, atau layanan/kinerja program studi yang diukur berdasarkan sejumlah standar yang ditetapkan itu.

Penilaian mutu dalam rangka akreditasi program studi sarjana harus dilandasi oleh standar yang lengkap dan jelas sebagai tolok ukur penilaian tersebut, dan juga memerlukan penjelasan operasional mengenai prosedur dan langkah-langkah yang ditempuh, sehingga penilaian itu dapat dilakukan secara sistemik dan sistematis.

2.2.3 Program Studi Sarjana

Program studi merupakan penataan program akademik bagi bidang studi tertentu yang didedikasikan untuk:

1. Menguasai, memanfaatkan, mendiseminasikan, mentransformasikan dan mengembangkan ilmu pengetahuan, teknologi, dan seni (Ipteks) dalam bidang studi tertentu.
2. Mempelajari, mengklarifikasikan dan melestarikan budaya yang berkaitan dengan bidang studi tertentu.
3. Meningkatkan mutu kehidupan masyarakat dalam kaitannya dengan bidang studi tertentu. Oleh karena itu program studi sebagai lembaga melaksanakan fungsi Tridarma Perguruan Tinggi, yaitu Pendidikan, Penelitian, dan Pengabdian kepada masyarakat, serta mengelola Ipteks selaras dengan bidang studi yang dikelolanya. Untuk menopang dedikasi dan fungsi tersebut, program studi harus mampu mengatur diri sendiri dalam upaya meningkatkan dan menjamin mutu secara berkelanjutan, baik yang berkenaan dengan masukan, proses maupun keluaran program akademik dan layanan yang diberikan kepada masyarakat selaras dengan bidang studi yang dikelolanya.

Dalam rangka mewujudkan akuntabilitas publik, program studi harus secara aktif membangun sistem penjaminan mutu internal. Untuk membuktikan bahwa sistem penjaminan mutu internal telah dilaksanakan dengan baik dan benar, program studi harus diakreditasi oleh lembaga penjaminan mutu eksternal

(BAN-PT). Dengan sistem penjaminan mutu yang baik dan benar, program studi akan mampu meningkatkan mutu, menegakkan otonomi, dan mengembangkan diri sebagai penyelenggara program akademik/profesional sesuai dengan bidang studi yang dikelolanya, dan turut serta dalam meningkatkan kekuatan moral masyarakat secara berkelanjutan.

Berdasarkan peraturan perundang-undangan yang berlaku dan berbagai pertimbangan tersebut di atas, BAN-PT melakukan akreditasi bagi semua program studi dari semua institusi perguruan tinggi di seluruh Indonesia. Kriteria untuk mengevaluasi dan menilai komitmen tersebut dijabarkan dalam sejumlah standar akreditasi beserta parameternya.

2.2.3.1 Aspek-Aspek Pelaksanaan Akreditasi Program Studi

Dalam melaksanakan keseluruhan proses akreditasi program studi terdapat beberapa aspek pokok yang perlu diperhatikan oleh setiap pihak yang terkait, yaitu asesor, program studi sarjana yang diakreditasi, dan BAN-PT sendiri. Aspek-aspek tersebut yaitu :

1. Standar akreditasi program studi sarjana yang digunakan sebagai tolok ukur dalam mengevaluasi dan menilai mutu kinerja, keadaan dan perangkat kependidikan program studi sarjana.
2. Prosedur akreditasi program studi sarjana yang merupakan tahap dan langkah yang harus dilakukan dalam rangka akreditasi program studi sarjana.
3. Instrumen akreditasi program studi sarjana yang digunakan untuk menyajikan data dan informasi sebagai bahan dalam mengevaluasi dan menilai mutu program studi sarjana, disusun berdasarkan standar akreditasi yang ditetapkan.
4. Kode etik akreditasi program studi sarjana yang merupakan “aturan main” untuk menjamin kelancaran dan obyektivitas proses dan hasil akreditasi program studi sarjana.

2.2.4 Standar Akreditasi Program Studi Sarjana

Standar akreditasi adalah tolok ukur yang harus dipenuhi oleh program studi sarjana. Standar akreditasi terdiri atas beberapa parameter (indikator kunci) yang dapat digunakan sebagai dasar, sebagai berikut :

1. Penyajian data dan informasi mengenai kinerja, keadaan dan perangkat kependidikan program studi sarjana, yang dituangkan dalam instrumen akreditasi
2. Evaluasi dan penilaian mutu kinerja, keadaan dan perangkat kependidikan program studi sarjana
3. Penetapan kelayakan program studi sarjana untuk menyelenggarakan program-programnya
4. Perumusan rekomendasi perbaikan dan pembinaan mutu program studi sarjana.

Standar akreditasi program studi sarjana mencakup standar tentang komitmen program studi sarjana terhadap kapasitas institusional(*institutional capacity*) dan komitmen terhadap efektivitas program pendidikan(*educational effectiveness*), yang dikemas dalam tujuh standar akreditasi, yaitu:

Standar 1. Visi, Misi, Tujuan dan Sasaran, serta Strategi Pencapaian

Standar 2. Tata pamong, Kepemimpinan, Sistem Pengelolaan, dan Penjaminan mutu

Standar 3. Mahasiswa dan Lulusan

Standar 4. Sumber daya manusia

Standar 5. Kurikulum, Pembelajaran, dan Suasana Akademik

Standar 6. Pembiayaan, Sarana dan Prasarana, serta Sistem Informasi

Standar 7. Penelitian, Pelayanan/Pengabdian kepada Masyarakat, dan Kerjasama

Asesmen kinerja program studi sarjana didasarkan pada pemenuhan tuntutan standar akreditasi. Dokumen akreditasi program studi sarjana yang dapat diproses harus telah memenuhi persyaratan awal yang ditandai dengan adanya izin yang sah dan berlaku dalam penyelenggaraan program studi sarjana dari pejabat yang berwenang, memiliki anggaran dasar dan anggaran rumah tangga dan dokumen-dokumen rencana strategis atau rencana induk pengembangan yang menunjukkan dengan jelas visi, misi, tujuan dan sasaran program studi sarjana, nilai-nilai dasar yang dianut dan berbagai aspek mengenai organisasi dan pengelolaan program studi sarjana, proses pengambilan keputusan penyelenggaraan program, dan sistem jaminan mutu.

Berikut merupakan deskripsi standar beserta rincian elemen-elemen yang ada dan dinilai pada Standar 7.

2.2.4.1 Standar 7. Penelitian, Pengabdian Masyarakat dan Kerjasama

Standar ini adalah acuan keunggulan mutu penelitian, pelayanan pengabdian kepada masyarakat, dan kerjasama yang diselenggarakan untuk dan terkait dengan pengembangan mutu program studi sarjana.

Penelitian adalah salah satu tugas pokok program studi sarjana yang memberikan kontribusi dan manfaat kepada proses pembelajaran, pengembangan ilmu pengetahuan, teknologi, dan seni, serta peningkatan mutu kehidupan masyarakat. Program studi sarjana harus memiliki sistem perencanaan pengelolaan serta implementasi program-program penelitian yang menjadi unggulan. Sistem pengelolaan ini mencakup akses dan pengadaan sumber daya dan layanan penelitian bagi pemangku kepentingan, memiliki peta-jalan (*road-map*), melaksanakan penelitian serta mengelola dan meningkatkan mutu hasilnya dalam rangka mewujudkan visi, melaksanakan/ menyelenggarakan misi, dan mencapai tujuan yang dicita-citakan program studi sarjana.

Program studi sarjana menciptakan iklim yang kondusif agar dosen dan mahasiswa secara kreatif dan inovatif menjalankan peran dan fungsinya sebagai pelaku utama penelitian yang bermutu dan terencana. Program studi sarjana memfasilitasi dan melaksanakan kegiatan diseminasi hasil-hasil penelitian dalam

berbagai bentuk, antara lain penyelenggaraan forum/seminar ilmiah, presentasi ilmiah dalam forum nasional dan internasional, publikasi dalam jurnal nasional terakreditasi dan/atau internasional yang bereputasi.

Pelayanan/pengabdian kepada masyarakat dilaksanakan sebagai perwujudan kontribusi kepakaran, kegiatan pemanfaatan hasil pendidikan, dan/atau penelitian dalam bidang ilmu pengetahuan, teknologi, dan/atau seni, dalam upaya memenuhi permintaan dan/atau memprakarsai peningkatan mutu kehidupan bangsa. Program studi sarjana yang baik memiliki sistem pengelolaan kerja sama dengan pemangku kepentingan eksternal dalam rangka penyelenggaraan dan peningkatan mutu secara berkelanjutan program-program akademik. Hasil kerja sama dikelola dengan baik untuk kepentingan akademik dan sebagai perwujudan akuntabilitas program studi sarjana sebagai lembaga nirlaba. Program studi sarjana yang baik mampu merancang dan mendayagunakan program-program kerja sama yang melibatkan partisipasi aktif program studi sarjana dan memanfaatkan dan meningkatkan kepakaran dan mutu sumber daya program studi sarjana.

Akuntabilitas pelaksanaan tridarma dan kerja sama program studi sarjana diwujudkan dalam bentuk keefektifan pemanfaatannya untuk memberikan kepuasan pemangku kepentingan terutama peserta didik.

Elemen Penilaian :

- 7.1 Partisipasi aktif dalam perencanaan, implementasi, dan peningkatan mutu penelitian, PkM, dan kerjasama yang mendukung keunggulan yang diharapkan pada visi dan misi prodi dan institusi.
- 7.2 Kejelasan, transparansi, dan akuntabilitas sistem pengelolaan penelitian dan PkM, termasuk proses monitoring, evaluasi dan peninjauan ulang strategi secara periodik dalam rangka peningkatan mutu berkelanjutan.
- 7.3 *Benchmark* dan target mutu penelitian dan PkM.
- 7.4 Dukungan dan komitmen institusi pada prodi dalam pelaksanaan penelitian dan PkM dalam bentuk pendanaan secara internal dari perguruan

tingginya, upaya kerjasama, dan fasilitas yang sesuai dengan program dan kegiatan penelitian, PkM, dan kerjasama.

7.5 Partisipasi dosen dan mahasiswa dalam kegiatan penelitian, PkM, dan kerjasama.

7.6 Aktivitas penelitian, PkM, dan kerjasama yang berkontribusi dan berdampak pada proses pembelajaran.

7.7 Produktivitas dan mutu hasil penelitian dosen dan atau mahasiswa prodi (publikasi dosen pada jurnal nasional terakreditasi - kuantitas dan produktivitas; publikasi dosen pada jurnal internasional - kuantitas dan produktivitas; sitasi hasil publikasi dosen; karya inovatif (paten, karya/produk monumental).

7.8 Kegiatan PkM dosen dan mahasiswa prodi yang bermanfaat bagi pemangku kepentingan (kerjasama, karya, penelitian, dan pemanfaatan jasa/produk kepakaran).

7.9 Jumlah dan mutu kerjasama yang efektif yang mendukung pelaksanaan misi prodi dan institusi dan dampak kerjasama untuk penyelenggaraan dan pengembangan prodi.

2.2.5 Borang Akreditasi Program Studi

Borang akreditasi program studi merupakan instrumen untuk mengumpulkan informasi mengenai kinerja Program Studi pada perguruan tinggi negeri, perguruan tinggi kedinasan, perguruan tinggi keagamaan, dan perguruan tinggi swasta dalam rangka pengendalian mutu semua perguruan tinggi di Indonesia.

Borang akreditasi terdiri atas seperangkat pertanyaan yang sebagian berupa pertanyaan tertutup, dan sebagian lagi berupa pertanyaan terbuka. Sesuai dengan petunjuk pada setiap butir pertanyaan, maka sebagian dari pertanyaan-pertanyaan dapat dijawab dengan menuliskan jawabannya pada tempat yang disediakan dalam borang akreditasi, sebagian lagi memerlukan lembaran tersendiri.

Borang akreditasi ini terdiri dari dua bagian yaitu borang program studi yang diisi oleh Program Studi yang diakreditasi dan portofolio yang diisi oleh

fakultas atau sekolah tinggi yang mengelola program studiiitu. Untuk proses akreditasi program studi yang berada dalam satu fakultas/sekolah tinggi, pihak fakultas atau sekolah tinggi hanya perlu membuat satu versi isian portofolio fakultas atau sekolah tinggi.

Dalam pertanyaan tertentu diminta lampiran untuk melengkapi informasi yang berkaitan dengan jawaban terhadap pertanyaan yang bersangkutan. Lampiran diberikan dalam bentuk buku yang terpisah dari borang akreditasi. Setiap lampiran didahului oleh lembar halaman dengan warna berbeda yang berisi nomor lampiran dan judulnya. Lampiran-lampiran tersebut merupakan komponen yang dijadikan bahan kelengkapan bagi program studi dan sebagai bukti.

Informasi yang diperoleh dari jawaban terhadap pertanyaan-pertanyaan dalam borang akreditasi itu digunakan untuk dua tujuan pokok, yaitu untuk menilai kinerja akademik dan administratif program studi dan menemukan dimensi-dimensi kinerja Program studi yang memerlukan perbaikan atau pembinaan.

Pertanyaan yang dituangkan dalam borang akreditasi disusun berdasarkan sembilan dimensi mutu yang menunjukkan mutu suatu program studi. Kesembilan dimensi ini menunjukkan mutu komprehensif dari suatu penyelenggaraan program untuk menghasilkan keluaran yang bermutu tinggi, sesuai dengan bidang ilmu masing-masing. Hubungan kesembilan dimensi tersebut mewujudkan prinsip RAISE++ (*Relevance, Academic Atmosphere, Institutional Commitment, Sustainability, Efficiency, Leadershi, and Equity*), adalah sebagai berikut:

1. Kelayakan (*Appropriateness*)

Merupakan tingkat ketepatan unsur masukan, proses, keluaran, maupun tujuan program ditinjau dari ukuran ideal secara normatif.

2. Kecukupan (*Adequacy*)

Menunjukkan tingkat ketercapaian persyaratan ambang yang diperlukan untuk penyelenggaraan suatu program.

3. Relevansi/Kesesuaian (*Relevancy*)

Merupakan tingkat keterkaitan tujuan maupun hasil/keluaran program pendidikan dengan kebutuhan masyarakat di lingkungannya maupun secara global.

4. Suasana Akademik (*Academic Atmosphere*)

Merujuk pada iklim yang mendukung interaksi antara dosen dan mahasiswa, antara sesama mahasiswa, maupun antara sesama dosen untuk mengoptimalkan proses pembelajaran.

5. Efisiensi (*Efficiency*)

Merujuk pada tingkat pemanfaatan masukan (sumberdaya) yang digunakan untuk proses pembelajaran.

6. Keberlanjutan (*Sustainability*)

Menggambarkan keberlangsungan penyelenggaraan program yang mencakup ketersediaan masukan, aktivitas pembelajaran, maupun pencapaian hasil yang optimal.

7. Selektivitas (*Selectivity*)

Menunjukkan bagaimana penyelenggara program memilih unsur masukan, aktivitas proses pembelajaran, maupun penentuan prioritas hasil/keluaran berdasarkan pertimbangan kemampuan/kapasitas yang dimiliki.

8. Produktivitas (*productivity*)

Menunjukkan tingkat keberhasilan proses pembelajaran yang dilakukan dalam memanfaatkan masukan.

9. Efektivitas (*Effectiveness*)

Adalah tingkat ketercapaian tujuan program yang telah ditetapkan yang diukur dari hasil/keluaran program.

2.2.5.1 Panduan Pengisian Borang Program Studi Standar 7

**STANDAR 7. PENELITIAN, PELAYANAN/PENGABDIAN KEPADA
MASYARAKAT, DAN KERJA SAMA**

Tabel 2.1 Pengisian borang standar 7

No. Butir	No. Kolom	Panduan Pengisian
7.1.1	(2)-(4)	Berdasarkan sumber pembiayaan, tuliskan jumlah seluruh hasil penelitian dosen tetap yang sesuai dengan bidang keilmuan PS pada kolom (2) s.d. (4), selama tiga tahun terakhir.
7.1.2		Beri tanda √ pada kondisi yang sesuai tentang keterlibatan mahasiswa yang melakukan tugas akhir melalui skripsi dalam penelitian dosen selama tiga tahun terakhir. Tuliskan banyaknya mahasiswa yang terlibat dalam penelitian dosen untuk skripsinya serta jumlah total mahasiswa yang melakukan tugas akhir melalui skripsi dalam tiga tahun terakhir pada tempat yang disediakan.
7.1.3	(2)-(5)	Artikel ilmiah/karya ilmiah/karya seni/buku yang dihasilkan oleh dosen tetap PS dalam tiga tahun terakhir. Tuliskan • judul artikel ilmiah/karya ilmiah/karya seni/buku pada kolom (2) • nama penulis, dimulai dengan penulis pertama, dan (jika ada) diikuti oleh nama penulis kedua, dst. pada kolom (3) • nama seminar atau nama penerbitan pada kolom (4) • tahun kegiatan, seminar/tahun terbit pada kolom (5)
	(6)-(8)	• Beri tanda √ pada kolom (6) jika cakupannya lokal • Beri tanda √ pada kolom (7) jika cakupannya nasional • Beri tanda √ pada kolom (8) jika cakupannya internasional
7.1.4	(2)	Sebutkan karya dosen dan atau mahasiswa Program Studi yang telah memperoleh/sedang memproses perlindungan Hak atas Kekayaan Intelektual (HaKI) dari Dirjen HaKI atau <i>International Office</i> dalam tiga tahun

No. Butir	No. Kolom	Panduan Pengisian
		terakhir. Lampirkan surat paten HaKI atau keterangan sejenis.
7.2.1	(2)-(4)	Berdasarkan sumber pembiayaan, tuliskan jumlah kegiatan pelayanan/ pengabdian kepada masyarakat selama tiga tahun terakhir yang sesuai dengan bidang PS pada kolom (2)-(4). Catatan: Pelayanan/Pengabdian kepada Masyarakat adalah penerapan bidang ilmu untuk menyelesaikan masalah di masyarakat (termasuk masyarakat industri, pemerintah, dsb.)
7.2.2		Keterlibatan mahasiswa dalam kegiatan pelayanan/ pengabdian kepada masyarakat selama tiga tahun terakhir. Beri tanda √ pada kondisi yang sesuai. Uraikan tingkat partisipasi dan bentuk keterlibatan mahasiswa dalam kegiatan pelayanan/ pengabdian kepada masyarakat pada tempat yang disediakan.
7.3.1	(2)-(6)	Kerja sama/kemitraan yang dimaksud adalah bentuk kerja sama yang melibatkan PS dengan institusi lain di dalam negeri dalam pelaksanaan aspek-aspek Tridarma PT, misalnya penelitian bersama, tukar menukar dosen dan mahasiswa, serta penyelenggaraan seminar bersama. Kegiatan kerja sama dalam tiga tahun terakhir. Tuliskan • nama institusi, pada kolom (2) • jenis kegiatan kerja sama, pada kolom (3) • waktu mulai kerja sama (tanggal, bulan, dan tahun), pada kolom (4) • waktu selesai kerja sama, pada kolom (5). Jika kerja sama masih

No. Butir	No. Kolom	Panduan Pengisian
		berlangsung dan tidak ada batasan waktu berakhirnya, maka pada kolom ini ditulis “masih berlangsung”. • manfaat yang diperoleh dari kegiatan kerja sama tersebut, pada kolom (6)
7.3.2	(2)-(6)	Kerja sama/kemitraan yang dimaksud adalah bentuk kerja sama yang melibatkan PS dengan institusi lain di luar negeri dalam pelaksanaan aspek-aspek Tridarma PT, misalnya penelitian bersama, tukar menukar dosen dan mahasiswa, penyelenggaraan seminar bersama, dll. Kegiatan kerja sama dalam tiga tahun terakhir. Tuliskan • nama institusi, pada kolom (2) • jenis kegiatan kerja sama, pada kolom (3) • waktu mulai kerja sama (tanggal, bulan, dan tahun), pada kolom (4) • waktu selesai kerja sama, pada kolom (5). Jika kerja sama masih berlangsung dan tidak ada batasan waktu berakhirnya, maka pada kolom ini ditulis “masih berlangsung”. • Manfaat yang telah diperoleh dari kegiatan kerja sama tersebut, pada kolom (6)

2.3 Aplikasi Berbasis Web

Aplikasi berbasis web adalah sebuah sistem informasi yang mendukung interaksi pengguna melalui antarmuka berbasis web. Fitur-fitur aplikasi web biasanya berupa *data persistence*, mendukung transaksi dan komposisi halaman web dinamis yang dapat dipertimbangkan sebagai hibridisasi, antara hipermedia dan sistem informasi. Aplikasi web adalah bagian dari *client-side* yang dapat dijalankan oleh browser web. *Client-side* mempunyai tanggung jawab untuk pengeksekusian proses bisnis/sistem. (Janner, 2010)

Perancangan dan pembangunan aplikasi berbasis web setidaknya memiliki 3 unsur didalamnya, yaitu web server, basis data, dan bahasa pemrograman dalam pembentukannya. Terdapat beberapa jenis server web yang umumnya digunakan yaitu Apache, ISS (*Internet Information Services*), nginx (*Engine-X*) dan lain-lain. Jenis basis data yang biasa digunakan yaitu MySQL, Oracle, Microsoft Access, Postgre dan lain-lain. Dalam hal bahasa pemrograman juga terdapat bermacam-macam jenis yang bisa digunakan, di antaranya HTML (*HyperText Markup Language*), PHP (*PHP HyperText Preprocessor*), CSS (*Cascading Style Sheet*), ASP (*Active Server Page*), Javascript dan lain-lain.

Prinsip dasar kerja dari aplikasi web adalah sebagai berikut :

1. Aplikasi web disimpan dalam dokumen yang disebut dengan halaman-halaman web (*web pages*).
2. Halaman web kemudian disimpan dalam server web, dan komputer server web ini akan melayani permintaan dari *client*.
3. Komputer *client* melakukan eksekusi aplikasi melalui *web browser* yang terdapat pada komputer *client* dengan memasukkan alamat dari aplikasi web yang ditempatkan di server web.

Berikut dibawah ini merupakan penggambaran dari cara kerja aplikasi berbasis *web*.



Gambar 2.4 Cara kerja aplikasi berbasis web (Janner, 2010)

Dalam pengerjaan tugas akhir ini, penulis menggunakan PHP sebagai bahasa pemrogramannya, MySQL dan phpmyadmin dalam linux ubuntu sebagai tempat penyimpanan sekaligus pengolahan database dan juga Apache dalam linux ubuntu sebagai server webnya. Dibawah ini akan dijelaskan sedikit definisi dari PHP, MySQL dan phpmyadmin, Apache dan sistem operasi Linux.

2.3.1 PHP (HyperText Preprocessor)

PHP adalah sebuah bahasa pemrograman scripting untuk membuat halaman web yang dinamis. Walaupun dikenal sebagai bahasa untuk membuat halaman web, tapi PHP sebenarnya juga dapat digunakan untuk membuat aplikasi command-line dan juga GUI. Cara kerja PHP adalah dengan menyelipkannya di antara kode HTML).

Website yang menggunakan PHP membutuhkan aplikasi yang bernama server web sebagai tempat pemrosesan kode PHP. Server web ini yang akan memproses kode PHP dan menghasilkan output berupa halaman web. PHP ini bersifat terbuka dan multiplatform, karenanya dapat dijalankan di banyak jenis server web.

2.3.2 CSS (Cascading Style Sheet)

Cascading Style Sheet (CSS) adalah suatu kumpulan kode-kode untuk memformat, yang mengendalikan tampilan dan mempercantik isi dalam suatu

halaman web yang akan ditampilkan. Penulisan CSS ini terdapat tiga cara, yaitu inline, internal, dan eksternal. Pada penulisan kode CSS dengan metode inline ini bisa dilakukan langsung pada tag yang ingin diberi style dengan menggunakan atribut style, contohnya :

```
<h1 style="color : blue;"> Aplikasi Borang </h1>.
```

Pada penulisan kode CSS dengan metode Internal ditulis dalam tag style dan ditulis dalam tag head, contohnya :

```
<html><head><title>Aplikasi Borang</title>
<style>
  h1 {
  Color : blue ;
  }
</style>
</head>
</html>
```

Pada penulisan kode CSS dengan metode Eksternal ini membuat file CSS di panggil dalam tag head, contohnya :

```
<title>Aplikasi Borang</title>
<link rel="stylesheet" href="iklan.css" />
```

Kelebihan lain dari penggunaan CSS dibandingkan dengan menggunakan kode HTML saja yaitu lebih hemat waktu dan lebih mudah dalam mengedit tampilan web. Hal ini dikarenakan CSS fungsinya seperti master halaman. Jadi ketika ingin mengubah seluruh web hanya cukup mengubah dari file CSS-nya tanpa perlu satu per satu dari tiap halaman. Ini juga berefek pada *loading* halaman yang menjadi lebih cepat, karena ukuran file tiap halaman web jadi lebih kecil. Sedangkan kekurangannya adalah, beberapa kode CSS yang berjalan baik pada suatu web browser, belum tentu bisa aktif pada web browser yang lain.

2.3.3 Database dan MySQL

Basis data (*database*) adalah kumpulan informasi yang disimpan di dalam komputer secara sistematis sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut(<http://id.wikipedia.org/wiki/Database>). Database digunakan untuk menyimpan

informasi atau data yang terintegrasi dengan baik di dalam komputer. Untuk mengelola *database* diperlukan suatu perangkat lunak yang disebut **DBMS** (*Database Management System*). DBMS merupakan suatu system perangkat lunak yang memungkinkan user (pengguna) untuk membuat, memelihara, mengontrol, dan mengakses *database* secara praktis dan efisien. Dengan DBMS, user akan lebih mudah mengontrol dan memanipulasi data yang ada. Sedangkan **RDBMS** atau *Relationship Database Management System* merupakan salah satu jenis DBMS yang mendukung adanya *relationship* atau hubungan antar tabel. Di samping RDBMS, terdapat jenis DBMS lain, misalnya *Hierarchy DBMS*, *Object Oriented DBMS*, dsb.

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (bahasa Inggris: *database management system*) atau DBMS yang multithread, multi-user, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis di bawah lisensi GNU *General Public License* (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL.

Sedangkan phpmyadmin merupakan *web base control panel* untuk MySQL yang telah terinstall didalam komputer. Phpmyadmin merupakan front-end MySQL berbasis web. PhpMyAdmin dibuat dengan menggunakan PHP. Saat ini phpmyadmin banyak digunakan dalam hampir semua penyedia hosting yang ada di internet. Phpmyadmin mendukung berbagai fitur administrasi MySQL termasuk manipulasi database, tabel, index dan juga dapat mengekspor data ke dalam berbagai format data.

2.3.4 Apache (Linux Base)

Apache atau Server Web/WWW Apache adalah server web yang dapat dijalankan di banyak sistem operasi (Unix, BSD, Linux, Microsoft Windows dan Novell Netware serta platform lainnya) yang berguna untuk melayani dan memfungsikan situs web. Protokol yang digunakan untuk melayani fasilitas web/www ini menggunakan HTTP.

Apache memiliki fitur-fitur canggih seperti pesan kesalahan yang dapat dikonfigurasi, autentikasi berbasis basis data dan lain-lain. Apache juga didukung oleh sejumlah antarmuka pengguna berbasis grafik (GUI) yang memungkinkan penanganan server menjadi mudah.

Apache merupakan perangkat lunak sumber terbuka dikembangkan oleh komunitas terbuka yang terdiri dari pengembang-pengembang dibawah naungan Apache Software Foundation.

2.3.5 Linux

Sistem Operasi adalah perangkat lunak yang mengatur segala koordinasi kerja antar semua perlengkapan perangkat keras di dalam sebuah komputer. Linux adalah sistem operasi yang bersifat open source, yang disebar secara luas dengan bebas di bawah lisensi GNU General Public License (GPL), dimana setiap orang bisa mengembangkan aplikasi di dalamnya. Sekarang Linux adalah sistem UNIX yang lengkap dan spesial, bisa digunakan untuk jaringan (networking), pengembangan software, bahkan untuk kebutuhan sehari-hari. Linux adalah alternatif sistem operasi yang jauh lebih murah dibandingkan dengan sistem operasi yang komersial, dengan kemampuan linux yang setara atau bahkan lebih (Romi Satrio:2003).

Keunggulan linux meliputi true-multitasking, virtual memory, desain dan keamanan multiuser, stabilitas, kemudahan akses device, dan lain-lain. Linux ini sendiri memiliki banyak distro yang tersedia dimana setiap distro ini mempunyai ciri khasnya masing masing. Penulis cukup akrab dengan linux berbasis distro debian dan redhat, khususnya Ubuntu dan Centos.

2.3.6 Bluefish Editor

Pengguna linux juga mempunyai banyak pilihan editor untuk pemrograman seperti, **Emacs**, **Jed**, ataupun **gPHPEdit**. Dengan menggunakan aplikasi tersebut memang sudah bisa dikatakan cukup bagi programmer web berbasis Linux. Namun ada satu lagi aplikasi yang mungkin lebih kompleks dan powerfull dibanding editor di atas, yang dipanggil dengan panggilan **Bluefish**. Ini bukanlah aplikasi game melainkan sebuah aplikasi editor untuk para programmer maupun web desainer.

Bluefish selama ini mengalami perkembangan yang sangat pesat dan telah mendukung hampir seluruh bahasa pemrograman maupun bahasa markup, mulai dari C++ hingga CSS. Adapun kelebihan yang dimiliki aplikasi ini, diantaranya adalah:

- Selain untuk lingkungan Linux, Bluefish juga tersedia bagi pengguna Windows
- Aplikasi ini tidak membutuhkan banyak resource memori komputer.
- Mempunyai tools yang akan bermanfaat untuk para desainer web.
- Memudahkan kita memahami sintaks dengan highlight yang menarik.
- Free and Open Source.
- Auto-recovery pada dokumen atau sintaks program jika suatu saat aplikasi kita mengalami crash maupun mati mendadak.
- Auto tag closing, menutup secara otomatis tag bahasa pemrograman web tertentu, misalnya tag pada HTML.

- Untuk lebih lengkapnya baca juga di <http://bluefish.openoffice.nl/features.html>

2.4 Unified Modeling Language

UML (Unified Modeling Language) adalah sebuah bahasa yang digunakan untuk memvisualisasikan, membangun, dan mendokumentasikan sebuah sistem pengembangan software berbasis OO (Object-Oriented). UML tidak hanya merupakan sebuah bahasa pemrograman visual saja, namun juga dapat secara langsung dihubungkan ke berbagai bahasa pemrograman, seperti JAVA, C++, Visual Basic, atau bahkan dihubungkan secara langsung ke dalam sebuah object-oriented database. Tujuan Penggunaan UML :

1. Memberikan bahasa pemodelan yang bebas dari berbagai bahas pemrograman dan proses rekayasa sebuah sistem.
2. Menyatukan praktek-praktek terbaik yang terdapat dalam pemodelan.
3. Memberikan model yang siap pakai, bahasa pemodelan visual yang ekspresif untuk mengembangkan dan saling menukar model dengan mudah dan dimengerti secara umum.
4. UML bisa juga berfungsi sebagai sebuah (blue print) cetak biru karena sangat lengkap dan detail. Dengan cetak biru ini maka akan bisa diketahui informasi secara detail tentang coding program atau bahkan membaca program dan menginterpretasikan kembali ke dalam bentuk diagram.

Berikut penjelasan dari 13 jenis diagram yang ada pada UML,

1. Use Case Diagram

Use case adalah abstraksi dari interaksi antara system dan actor. Use case bekerja dengan cara mendeskripsikan tipe interaksi antara user sebuah system dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah system dipakai. Use case merupakan konstruksi untuk mendeskripsikan bagaimana system akan terlihat di mata user. Sedangkan use case diagram memfasilitasi komunikasi diantara analis dan pengguna serta antara analis dan client.

2. Activity Diagram

Pada dasarnya diagram Activity sering digunakan oleh *flowchart*. Diagram ini berhubungan dengan diagram Statechart. Diagram Statechart berfokus pada *obyek yang dalam suatu proses* (atau proses menjadi suatu obyek), diagram Activity berfokus pada *aktifitas-aktifitas yang terjadi yang terkait dalam suatu proses tunggal*. Jadi dengan kata lain, diagram ini menunjukkan bagaimana aktifitas-aktifitas tersebut bergantung satu sama lain.

3. Sequence Diagram

Diagram Class dan diagram Object merupakan suatu gambaran *model statis*. Namun ada juga yang bersifat *dinamis*, seperti **Diagram Interaction**. Diagram sequence merupakan salah satu diagram Interaction yang menjelaskan bagaimana suatu operasi itu dilakukan; *message* (pesan) apa yang dikirim dan kapan pelaksanaannya. Diagram ini diatur berdasarkan waktu. Obyek-obyek yang berkaitan dengan proses berjalannya operasi diurutkan dari kiri ke kanan berdasarkan waktu terjadinya dalam pesan yang terurut.

4. Communication Diagram (Collaboration diagram in versi 1.x)

Collaboration diagram menggambarkan interaksi antar objek seperti sequence diagram, tetapi lebih menekankan pada peran masing-masing objek. Setiap message memiliki sequence number, di mana message dari level tertinggi memiliki nomor 1. Messages dari level yang sama memiliki prefiks yang sama. Diagram Collaboration juga merupakan *diagram interaction*. Diagram membawa informasi yang sama dengan diagram Sequence, tetapi lebih memusatkan atau memfokuskan pada kegiatan obyek dari waktu pesan itu dikirimkan.

5. 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.

6. State Machine Diagram (Statechart diagram in versi 1.x)

Statechart diagram menggambarkan transisi dan perubahan keadaan (dari satu *state* ke *state* lainnya) suatu objek pada sistem sebagai akibat dari *stimuli* yang diterima. Pada umumnya *statechart diagram* menggambarkan *class* tertentu (satu *class* dapat memiliki lebih dari satu *statechart diagram*). Dalam UML, *state* digambarkan berbentuk segiempat dengan sudut membulat dan memiliki nama sesuai kondisinya saat itu. Transisi antar *state* umumnya memiliki kondisi *guard* yang merupakan syarat terjadinya transisi yang bersangkutan, dituliskan dalam kurung siku. *Action* yang dilakukan sebagai akibat dari *event* tertentu dituliskan dengan diawali garis miring. Titik awal dan akhir digambarkan berbentuk lingkaran berwarna penuh dan berwarna setengah.

7. 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.

8. 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 fisik. Sebuah *node* adalah server, *workstation*, atau piranti keras lain yang digunakan untuk men-*deploy* komponen dalam lingkungan sebenarnya. Hubungan antar *node* (misalnya TCP/IP) dan *requirement* dapat juga didefinisikan dalam diagram ini.

9. Composite Structure Diagram

Diagram struktur komposit adalah diagram yang menunjukkan struktur internal classifier, termasuk poin interaksinya ke bagian lain dari sistem. Hal ini

menunjukkan konfigurasi dan hubungan bagian, yang bersama-sama melakukan perilaku classifie. Diagram struktur komposit merupakan jenis diagram struktur statis dalam Unified Modeling Language (UML), yang menggambarkan struktur internal kelas dan kolaborasi.

10. Interaction Overview Diagram

Interaction Overview Diagram adalah pencangkakan secara bersama antara *activity diagram* dengan *sequence diagram*. *Interaction Overview Diagram* dapat dianggap sebagai *activity diagram* dimana semua aktivitas diganti dengan sedikit *sequence diagram*, atau bisa juga dianggap sebagai *sequence diagram* yang dirincikan dengan notasi *activity diagram* yang digunakan untuk menunjukkan aliran pengawasan.

11. Object Diagram

Object diagram merupakan sebuah gambaran tentang objek-objek dalam sebuah sistem pada satu titik waktu. Karena lebih menonjolkan perintah-perintah daripada *class*, *object diagram* lebih sering disebut sebagai sebuah diagram perintah.

12. Package Diagram

Diagram objek melengkapi notasi grafik untuk pemodelan objek, kelas dan relasinya dengan yang lain. Diagram objek bermanfaat untuk pemodelan abstrak dan membuat perancangan program. Untuk mengatur pengorganisasian diagram Class yang kompleks, dapat dilakukan pengelompokan kelas-kelas berupa *package* (paket-paket). *Package* adalah kumpulan elemen-elemen logika UML.

13. Timing Diagram

Timing Diagram adalah bentuk lain dari *interaction diagram*, dimana fokus utamanya lebih ke waktu. *Timing diagram* sangat berdaya guna dalam menunjukkan faktor pembatas waktu diantara perubahan *state* pada objek yang berbeda.

Dalam penulisan tugas akhir ini, penulis hanya menggunakan *Use Case Diagram*, *Activity Diagram* dan *Sequence Diagram*. Berikut pada sub bab dibawah ini akan dijelaskan mengenai definisi dari ketiga diagram tersebut.

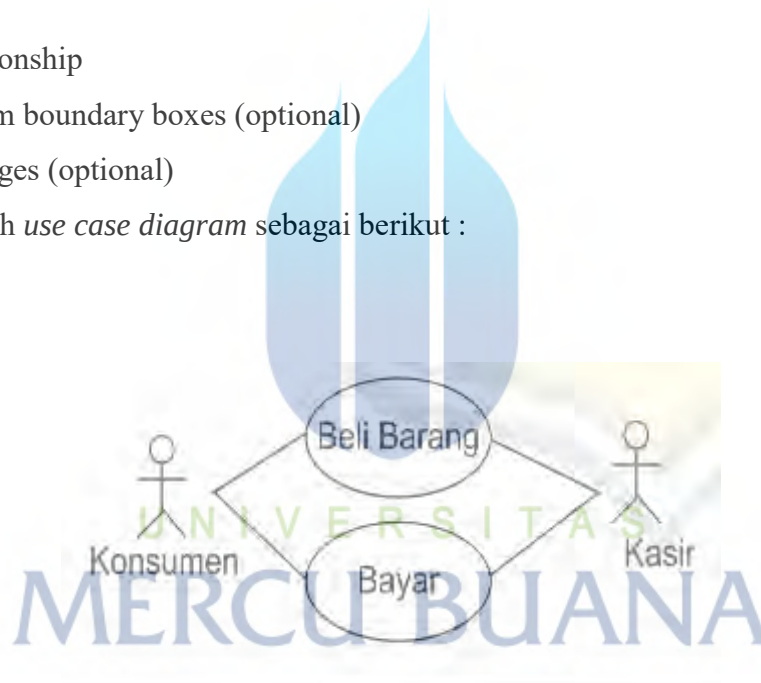
2.4.1 Use Case Diagram

Use-case diagram adalah mendeskripsikan fungsionalitas dari sebuah system dan yang ditekankan pada *use case diagram* adalah “apa” yang dikerjakan oleh sebuah system, bukan “bagaimana” system itu mengerjakan. Menggambarkan kebutuhan system pada proses komputerisasi yang menghubungkan hubungan antara *actor* dan *use case*. *Use case* menggambarkan proses system, perilaku system dan segala transaksi dalam system.

Use-case diagram terdiri dari beberapa factor :

- Use case
- Actor
- Relationship
- System boundary boxes (optional)
- Packages (optional)

Contoh *use case diagram* sebagai berikut :



Gambar 2.5 Contoh Use Case Diagram

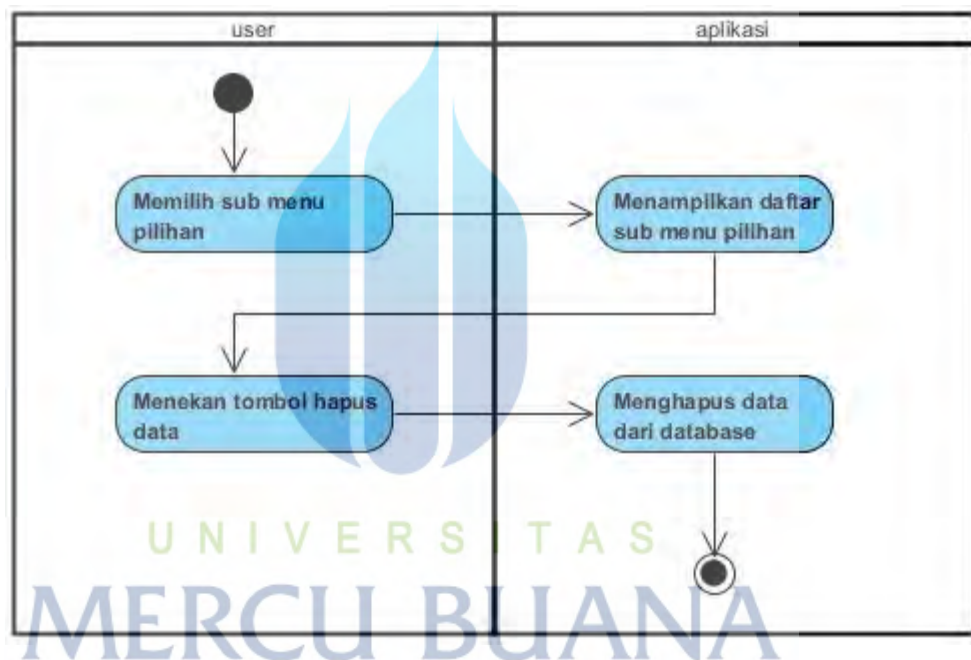
2.4.2 Activity Diagram

Activity diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. (Rosa, 2011).

Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal-hal berikut :

1. Rancangan proses bisnis di mana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan
2. Urutan atau pengelompokkan tampilan dari sistem/user interface dimana setiap aktivitas dianggap memiliki sebuah rancangan antar muka tampilan.
3. Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya.

Berikut adalah contoh dari Activity Diagram :



Gambar 2.6 Contoh Activity Diagram

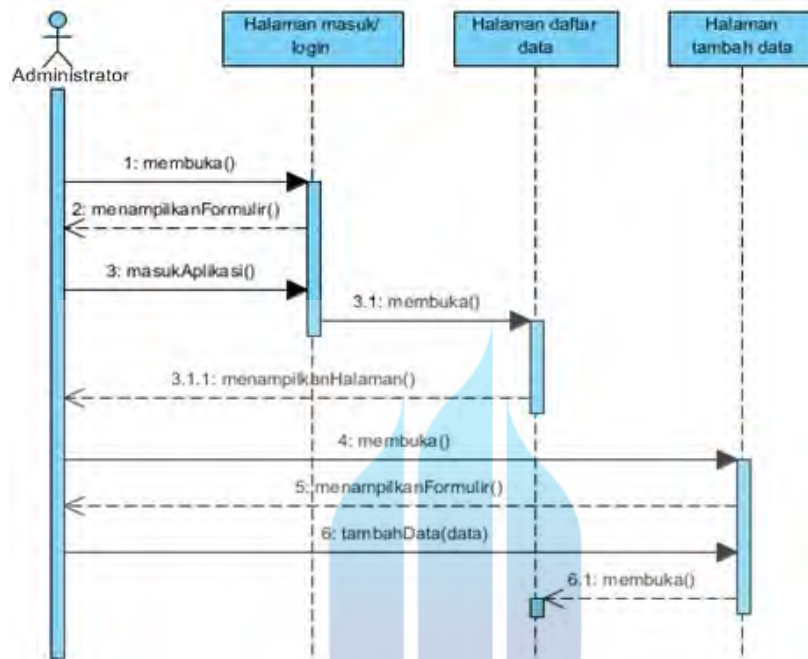
2.4.3 Sequence Diagram

Sequence diagram 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 sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. (Rosa, 2011)

Banyaknya diagram sekuen yang harus digambar adalah sebanyak pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use*

case yang telah didefinisikan interaksi jalannya pesan sudah dicakup pada diagram sekuen sehingga semakin banyak *usecase* yang didefinisikan maka diagram sekuan yang harus dibuat juga semakin banyak.

Berikut adalah contoh dari sequence diagram :



Gambar 2.7 Contoh Sequence Diagram

2.5 Metode Pengujian Kotak Hitam (*Black Box*)

Pengujian merupakan salah satu bagian yang penting dari proses pembuatan aplikasi. Pengujian dilakukan untuk menguji setiap modul menjalankan fungsinya dengan baik. Metode pengujian *black box* terfokus pada apakah unit program memenuhi kebutuhan (*requirement*) yang disebutkan dalam spesifikasi. Pada *black box testing*, cara pengujian hanya dilakukan dengan menjalankan atau mengeksekusi unit atau modul, kemudian diamati apakah hasil output sesuai dengan proses yang diinginkan. (Hanif, 2007)