

BAB II

LANDASAN TEORI

2.1. STIKES Yayasan Dr. Soetomo

Stikes Yayasan RS Dr. Soetomo adalah perguruan tinggi swasta milik Yayasan RS. Dr. soetomo yang memperoleh ijin operasional berdasar Surat keputusan Dirjen Dikti Kemendikbud Nomor 31/E/O/2012 Tanggal 26 Januari 2012 tentang Perubahan Bentuk Akademi Perkam Medik dan Informatika Kesehatan (APIKES) Pena Husada menjadi Sekolah Tinggi Ilmu Kesehatan (STIKES) Yayasan RS. Dr. Soetomo. (STIKES, 2015)

Kampus STIKES Yayasan RS. Dr. Soetomo di Jl. Karangmenjangan 12, Surabaya, dan mempunyai 2 (dua) Program Studi, yaitu program studi D-III Rekam Medik dan Informatika Kesehatan dan program studi S-1 Administrasi Rumah Sakit. Pilihan dua program studi tersebut berdasar analisis bahwa tenaga kesehatan Ahli madya Perkam Kesehatan serta Sarjana Administrasi Rumah Sakit adalah tenaga kesehatan yang mempunyai peran strategis untuk meningkatkan mutu pelayanan kesehatan dan pengelolaan rumah sakit di era persaingan global.

Undang-Undang nomor 44 Tahun 2009 tentang Rumah Sakit dan Undang-undang nomor 36 tahun 2009 tentang Kesehatan mengamanatkan bahwa rumah sakit harus memiliki tenaga tetap di bidang manajemen rumah sakit dan tenaga perekam medis. kebutuhan manajer madya yang mempunyai kemampuan dan keahlian di bidang perumahsakitian dan rekam medis akan sangat dibutuhkan dibutuhkan dimasa-masa yang akan datang. Pendidikan S-1 administrasi rumah

Sakit dan D-III Rekam Medik dan Informasi kesehatan didesain untuk menyiapkan lulusan yang memiliki kualifikasi manajerial di bidang manajemen dan rekam medis di Rumah Sakit.

VISI STIKES YAYASAN RS Dr. SOETOMO (STIKES, 2015) :

Menjadi lembaga Pendidikan Tinggi Swasta yang mandiri, mencetak lulusan yang profesional dibidangnya, beriman dan bertaqwa kepada Tuhan YME.

MISI STIKES YAYASAN RS Dr. SOETOMO (STIKES, 2015):

1. Mengelola lembaga perguruan tinggi yang profesional, transparan dan akuntabel;
2. Mendorong pelaksanaan Tri Dharma Perguruan Tinggi dimasing-masing program studi secara proporsional;
3. Mendorong penyusunan kurikulum dimasing-masing program studi berbasis kompetensi dengan berorientasi pada perkembangan ilmu pengetahuan dan teknologi;
4. Mendorong pelaksanaan proses belajar mengajar dimasing-masing program studi menggunakan pendekatan kurikulum berbasis kompetensi dan berorientasi pada keseimbangan antara *hard skill* dan *soft skill*;
5. Mendorong pelaksanaan proses belajar mengajar dimasing-masing program studi agar dapat meningkatkan kemandirian dan profesionalisme lulusan;
6. Melakukan kerjasama dengan mitra dari berbagai perguruan tinggi dan instansi terkait untuk memenuhi kebutuhan dan bursa kerja lulusan;

7. Mengevaluasi dan mengembangkan tata kelola dibidang akademik dan non akademik sesuai dengan perkembangan ilmu pengetahuan dan teknologi.

2.2. Pembelajaran

Kata Pembelajaran berasal dari kata dasar belajar, dalam arti sempit, pembelajaran merupakan suatu proses belajar agar seseorang dapat melakukan kegiatan belajar. Sedangkan belajar adalah suatu proses perubahan tingkah laku karena interaksi individu dengan lingkungan dan pengalaman (Arifin, 2012). Dalam UU RI Nomor 20 Tahun 2003 tentang Sistem Pendidikan Nasional menyebutkan bahwa pembelajaran adalah proses interaksi peserta didik dengan pendidik dan sumber belajar pada lingkungan belajar. Menurut Miarso (Siregar & Nara, 2011), pembelajaran adalah usaha pendidikan yang dilaksanakan secara sengaja, dengan tujuan yang telah ditetapkan sebelum proses dilaksanakan serta pelaksanaannya terkendali. Sedangkan menurut Gagne dan Briggs, pembelajaran merupakan suatu sistem yang bertujuan untuk membantu proses belajar anak didik, yang dirancang sedemikian rupa untuk mendukung terjadinya proses belajar anak didik yang bersifat internal (Djamarah, 2010).

Dari beberapa pendapat tersebut, maka dapat disimpulkan bahwa, pembelajaran merupakan proses yang dilakukan oleh pendidik untuk membelajarkan peserta didik pada lingkungan belajar tertentu sehingga dapat merubah tingkah laku dari peserta didik menjadi lebih baik.

Berdasarkan peraturan menteri pendidikan nasional nomor 41 tahun 2007 tentang standar proses untuk satuan pendidikan dasar dan menengah, bahwa

standar proses meliputi perencanaan proses pembelajaran, pelaksanaan proses pembelajaran, penilaian hasil pembelajaran dan pengawasan hasil pembelajaran untuk terlaksananya proses pembelajaran yang efektif dan efisien.

a. Perencanaan proses pembelajaran

(Sanjaya, 2008) Mengungkapkan bahwa perencanaan berasal dari kata rencana yang artinya pengambilan keputusan tentang apa yang harus dilakukan untuk mencapai tujuan. Oleh karena itu perencanaan harus dimulai dengan penetapan tujuan yang akan dicapai, kemudian menetapkan langkah-langkah yang harus ditempuh untuk mencapai tujuan tersebut. Perencanaan proses pembelajaran yang baik akan berdampak pada proses pembelajaran yang baik pula. (Sanjaya, 2008) perencanaan proses pembelajaran meliputi program menyusun alokasi waktu, program tahunan, program semester, silabus dan rencana pelaksanaan pembelajaran.

Hal ini sebagaimana terdapat dalam Permendiknas Nomor 41 Tahun 2007, bahwa perencanaan proses pembelajaran meliputi silabus dan Rencana Pelaksanaan pembelajaran (RPP), namun pada permendiknas tersebut perencanaan lebih ditekankan pada silabus dan RPP.

1. Silabus

Silabus sebagai acuan pengembangan RPP memuat identitas mata pelajaran atau tema pelajaran, SK, KD, materi pembelajaran, kegiatan pembelajaran, indikator pencapaian kompetensi, penilaian, alokasi waktu, dan sumber belajar. Silabus dikembangkan oleh satuan pendidikan berdasarkan Standar Isi (SI) dan Standar Kompetensi Lulusan (SKL), serta panduan penyusunan Kurikulum Tingkat Satuan Pendidikan (KTSP).

2. Rencana Pelaksanaan Pembelajaran (RPP)

RPP dijabarkan dari silabus untuk mengarahkan kegiatan belajar peserta didik dalam upaya mencapai KD. RPP disusun untuk setiap KD yang dapat dilaksanakan dalam satu kali pertemuan atau lebih. Berdasarkan Permendiknas No. 41 tahun 2007, menyebutkan bahwa komponen dalam rencana pelaksanaan pembelajaran (RPP) memuat identitas mata pelajaran, standar kompetensi (SK), kompetensi dasar (KD), indikator pencapaian kompetensi, tujuan pembelajaran, materi ajar, alokasi waktu, metode pembelajaran, kegiatan pembelajaran, penilaian hasil belajar, dan sumber belajar.

b. Pelaksanaan Proses Pembelajaran

Berdasarkan Permendiknas Nomor 41 tahun 2007 bahwa pelaksanaan proses pembelajaran meliputi kegiatan membuka sampai menutup pembelajaran yang terbagi menjadi kegiatan pendahuluan, kegiatan inti dan kegiatan penutup.

1. Kegiatan Pendahuluan

Dalam kegiatan pendahuluan guru melakukan kegiatan yang meliputi mempersiapkan peserta didik untuk mengikuti proses pembelajaran, melakukan apersepsi (mengaitkan dengan materi sebelumnya dengan materi yang akan dipelajari), menjelaskan tujuan pembelajaran, dan menjelaskan uraian materi sesuai silabus.

2. Kegiatan Inti

Pelaksanaan kegiatan inti merupakan proses pembelajaran untuk mencapai Kompetensi Dasar (KD). Kegiatan inti menggunakan metode yang

disesuaikan dengan karakteristik peserta didik dan mata pelajaran.

Kegiatan inti meliputi proses eksplorasi, elaborasi dan konfirmasi.

3. Kegiatan Penutup

Kegiatan penutup meliputi kegiatan menyimpulkan hasil pembelajaran yang telah dilakukan, kegiatan penilaian, pemberian umpan balik dan dan memberikan tugas kepada peserta didik serta menyampaikan rencana pembelajaran pada pertemuan berikutnya.

c. Penilaian Hasil Pembelajaran

Penilaian merupakan proses memberikan atau menentukan nilai kepada objek tertentu berdasarkan suatu kriteria tertentu (Sudjana, 2010). Dalam proses pembelajaran, penilaian memiliki peranan yang penting salah satunya untuk mengetahui tercapai tidaknya proses pembelajaran yang telah dilakukan. Penilaian adalah suatu proses yang sistematis dari pengumpulan analisis, dan interpretasi informasi atau data untuk menentukan sejauhmana peserta didik telah mencapai tujuan pembelajaran (Arifin, 2012).

Penilaian hasil belajar adalah proses pemberian nilai terhadap hasil-hasil belajar yang dicapai peserta didik dengan kriteria tertentu (Sudjana, 2010). Penilaian hasil belajar dapat dilakukan melalui kegiatan ulangan harian, ulangan tengah semester, ulangan akhir semester, ulangan kenaikan kelas, ujian sekolah/madrasah, dan ujian nasional.

Berdasarkan Peraturan Menteri Pendidikan Nasional (Permendiknas) Nomor 20 tahun 2007 tentang Standar Penilaian Pendidikan Penilaian hasil belajar oleh pendidik menggunakan berbagai teknik yang disesuaikan dengan karakteristik kompetensi dan tingkat perkembangan peserta didik. Teknik

tersebut meliputi: 1) Teknik Tes berupa tes tertulis, tes lisan, dan tes praktik atau tes kinerja, 2) Teknik Observasi atau pengamatan dilakukan selama pembelajaran berlangsung dan/atau diluar kegiatan pembelajaran, 3) Teknik Penugasan baik perseorangan maupun kelompok dapat berbentuk tugas dan/atau proyek.

1.3. Tahapan *Software Development Life Cycle* (SDLC)

1.3.1. SDLC

Menurut (IEEE, 2004) *Software Development Life Cycle* (SDLC) adalah suatu pola yang digunakan untuk mengembangkan penerapan secara sistematis dengan melalui cara pendekatan kuantitatif untuk melakukan proses pengembangan, operasi dan pemeliharaan sistem perangkat lunak, SDLC terdiri dari:

a. Rencana (*Planning*)

Merencanakan proyek-proyek sistem yang terdiri dari perencanaan jangka panjang dan pendek. Menentukan proyek-proyek sistem yang akan dikembangkan, dan mendefinisikan proyek-proyek sistem.

b. Analisa (*Analysis*)

Mengidentifikasi dan mengevaluasi permasalahan, kesempatan, hambatan yang terjadi dan kebutuhan yang diharapkan sehingga dapat diusulkan perbaikannya. Dilakukan pula identifikasi terhadap personil kunci baik yang langsung maupun tidak langsung menyebabkan terjadinya masalah.

c. Desain (*Design*)

Didalam desain dibagi menjadi tiga fase, yaitu desain secara umum, desain secara terinci dan selektif sistem.

d. Implementasi (*Implementation*)

Merupakan tahap untuk meletakkan sistem supaya siap dioperasikan adapun tahapan yang dikerjakan, yaitu penerapan rencana implementasi dan pelaksanaan kegiatan implementasi.

e. Uji Coba (*Testing*)

Merupakan tahap untuk memastikan bahwa sistem sudah berjalan sesuai dengan harapan.

f. Pengelolaan (*Maintenance*)

Perlu dilakukannya perawatan untuk menemukan kemungkinan kesalahan-kesalahan kecil yang tidak terdeteksi di saat pengujian.

1.3.2. *Software Requirement*

Menurut (IEEE, 2004) *software requirement* menjelaskan tentang kebutuhan perangkat lunak yang akan dibangun disesuaikan dengan kebutuhan *stakeholder*, untuk memenuhi kebutuhan perangkat lunak tersebut maka perlu empat tahapan, meliputi:

a. Elisitasi

Elisitasi merupakan tahap pertama yang dilakukan untuk membangun sistem, pada tahap ini *stakeholder* dan pihak pengembang menyatukan pemahaman mengenai sistem yang akan dibangun. Hal yang dilakukan adalah mengidentifikasi kebutuhan *stakeholder* yang terkait.

b. Analisis

Melakukan analisis dan identifikasi kebutuhan dan permasalahan yang dihadapi oleh *stakeholder*. Kebutuhan tersebut akan diterapkan pada sistem yang akan dibangun dan permasalahan yang dihadapi, diharapkan bisa diselesaikan dengan bantuan sistem.

c. Spesifikasi

Pada tahapan ini melakukan dokumentasi terhadap kebutuhan perangkat lunak dan penyelesaian terhadap masalah yang sedang dihadapi oleh *stakeholder*. Hasil dokumentasi ini akan diserahkan pada *stakeholder* untuk diperiksa.

d. Verifikasi dan Validasi

Pada tahapan ini merupakan proses pemeriksaan dokumen perencanaan kebutuhan perangkat lunak dan penyelesaian masalah yang akan digunakan dalam pembangunan sistem. Tahapan ini mempunyai tujuan untuk memastikan persyaratan yang sudah dibuat, sudah sesuai dengan kebutuhan *stakeholder*.

1.3.3. *Software Design*

Menurut (IEEE, 2004) *software design* adalah proses desain, pendefinisian dan pengolahan data yang terkait dengan fungsi, desain basis data, desain *interface*, dan penjelasan *input*, *proses* dan *output*. Pada tahap ini merupakan tahapan dalam merancang *software* yang akan dibangun melalui.
















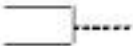

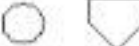

a. *System Flow*

Menurut (Jogiyanto, 2005) Bagan alir sistem (*system flowchart*) merupakan bagan yang menunjukkan arus pekerjaan secara keseluruhan

dari sistem. Bagan menjelaskan urutan-urutan dari prosedur-prosedure yang ada dalam sistem. Bagan alir sistem menunjukkan apa yang dikerjakan sistem. Pedoman dalam menggambar suatu bagan alir, analisis sistem atau pemrograman sebagai berikut:

- Bagan alir sebaiknya digambar dari atas ke bawah dan mulai dari bagian kiri dari suatu halaman.
- Kegiatan didalam bagan alir harus ditunjukkan dengan jelas.
- Harus ditunjukkan darimana kegiatan akan dimulai dan dimana akan berakhirnya.
- Masing-masing kegiatan didalam bagan alir harus didalam urutan yang semestinya.
- Kegiatan yang terpotong dan akan disambung ketempat lain harus ditunjukkan dengan jelas menggunakan symbol penghubung.

Simbol-simbol yang digunakan dalam *system flow* ditunjukkan pada Gambar 2.1.

Simbol	Nama	Keterangan	Simbol	Nama	Keterangan
	Simbol Dokumen	Menunjukkan dokumen input dan output baik untuk proses manual, mekanik atau komputer		Simbol drum magnetik	Menunjukkan input/output menggunakan drum magnetik
	Simbol Kegiatan Manual	Menunjukkan pekerjaan manual		Simbol Pita Kertas berlubang	Menunjukkan input/output menggunakan Pita Kertas berlubang
	Simbol Simpanan Offline	File non-komputer yang diarsip		Simbol Keyboard	Menunjukkan input yang menggunakan on-line keyboard
	Simbol Kartu Piong	Menunjukkan input/output yang menggunakan kartu piong (<i>punch card</i>)		Simbol display	Menunjukkan output
	Simbol Proses	Menunjukkan kegiatan proses dari operasi program komputer		Simbol pita kontrol	Menunjukkan penggunaan pita kontrol (<i>control tape</i>) dalam <i>batch control total</i> untuk pencocokan di proses <i>batch proses</i>
	Simbol operasi luar	Menunjukkan kegiatan proses di luar proses operasi komputer		Simbol hubungan komunikasi	Menunjukkan proses transmisi data melalui chanel komunikasi
	Simbol pengurutan offline	Menunjukkan proses pengurutan data di luar proses komputer		Simbol garis alir	Menunjukkan arus dari proses
	Simbol Pita Magnetik	Menunjukkan input/output menggunakan pita magnetik		Simbol penjelasan	Menunjukkan penjelasan dari suatu proses
	Simbol Hardisk	Menunjukkan input/output menggunakan hard disk		Simbol penghubung	Menunjukkan penghubung ke halaman yang masih sama atau ke halaman lain
	Simbol diskette	Menunjukkan input/output menggunakan diskette			

Gambar 2.1. Simbol-simbol pada *System Flow*

b. *Data Flow Diagram* (DFD)

DFD sering digunakan untuk menggambarkan suatu sistem yang telah ada atau sistem baru yang akan dikembangkan secara logika tanpa mempertimbangkan lingkungan fisik di tempat data tersebut mengalir. DFD merupakan alat yang digunakan pada metodologi pengembangan sistem yang terstruktur dan dapat mengembangkan arus data di dalam sistem dengan terstruktur dan jelas.

DFD fokus pada aliran data dari dan ke dalam sistem serta memproses data yang mengalir tersebut. Simbol-simbol dasar dalam DFD yaitu :

1. *External Entity*

Suatu *ExternalEntity* atau entitas merupakan orang, kelompok, departemen, atau sistem lain di luar sistem yang dibuat dapat menerima atau memberikan informasi atau data ke dalam sistem yang dibuat. Gambar 2.3 merupakan simbol entitas dalam DFD dalam model Gane dan Suatu *ExternalEntity* atau entitas merupakan orang, kelompok, departemen, atau sistem lain di luar sistem yang dibuat dapat menerima atau memberikan informasi atau data ke dalam sistem yang dibuat. Gambar 2.2 merupakan simbol entitas dalam DFD dalam model Gane dan Sarson.



Gambar 2.2 Simbol *ExternalEntity*

2. Data Flow

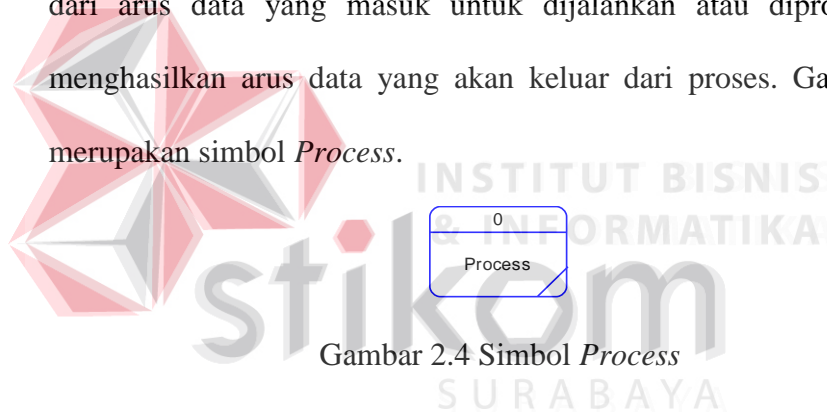
DataFlow atau aliran data disimbolkan dengan tanda panah. *DataFlow* menunjukkan arus data atau aliran data yang menghubungkan dua proses atau entitas dengan proses. Gambar 2.3 merupakan simbol *DataFlow*.



Gambar 2.3 Simbol *DataFlow*

3. Process

Suatu *Process* meliputi beberapa tindakan atau sekelompok tindakan dari arus data yang masuk untuk dijalankan atau diproses agar menghasilkan arus data yang akan keluar dari proses. Gambar 2.4 merupakan simbol *Process*.

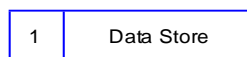


Gambar 2.4 Simbol *Process*

4. Data Store

DataStore adalah simbol yang digunakan untuk melambangkan proses penyimpanan data. Suatu nama perlu diberikan pada *DataStore* untuk menunjukkan nama dari *file*-nya. Gambar 2.5 merupakan simbol *file* penyimpanan/*Data Store* yang dapat berupa hal-hal sebagai berikut, sebagai gambaran:

1. Suatu *file* atau *Database* di sistem komputer.
2. Suatu arsip atau catatan manual.
3. Suatu tabel acuan manual.



Gambar 2.5 Simbol *Data Store*

Berikut ini adalah urutan langkah bagaimana menggambar suatu sistem pada DFD:

1. *Context Diagram*

Context Diagram merupakan langkah pertama dalam pembuatan *Data Flow Diagram*. Pada *Context Diagram* dijelaskan sistem apa yang dibuat dan *Entity* apa saja yang digunakan. Dalam *Context Diagram* harus ada arus data yang masuk dan arus data yang keluar.

2. *Data Flow Diagram Level 0*

DFD Level 0 adalah langkah selanjutnya setelah *Context Diagram*. Hal yang digambarkan dalam Diagram Level 0 ini adalah proses utama dari sistem serta hubungan *Entity*, *Process*, *Data Flow* dan *Data Store*.

3. *Data Flow Diagram Level 1*

DFD Level 1 merupakan penjelasan dari DFD Level 0. Pada proses ini dijelaskan proses apa saja yang dilakukan pada setiap proses yang terdapat di DFD Level 0.

c. *Entity Relationship Diagram (ERD)*

Entity Relationship Diagram (ERD) adalah gambaran pada sistem yang di dalamnya terdapat hubungan antara Entity beserta relasinya. Entity merupakan sesuatu yang ada dan terdefiniskan di dalam suatu organisasi, dapat abstrak dan nyata. Untuk setiap Entity biasanya mempunyai Attribute yang merupakan ciri

Entity tersebut. Menurut (Marlinda, 2004), Attribute memiliki pengertian kolom di sebuah relasi. Macam-macam Attribute yaitu :

1. *Simple Attribute*

Attribute ini merupakan *Attribute* yang unik dan tidak dimiliki oleh *Attribute* lainnya, misalnya *Entity* mahasiswa yang *Attribute*-nya NIM.

2. *Composite Attribute*

Composite Attribute adalah *Attribute* yang memiliki dua nilai harga, misalnya nama besar (nama keluarga) dan nama kecil (nama asli).

3. *Single Value Attribute*

Attribute yang hanya memiliki satu nilai harga, misalnya *Entity* mahasiswa dengan *Attribute*-nya umur (tanggal lahir).

4. *Multi Value Attribute*

Multi Value Attribute adalah *Attribute* yang banyak memiliki nilai harga, misalnya *Entity* mahasiswa dengan *Attribute*-nya pendidikan (SD, SMP, SMA).

5. *Null Value Attribute*

Null Value Attribute adalah *Attribute* yang tidak memiliki nilai harga, misalnya *Entity* tukang becak dengan *Attribute*-nya pendidikan (tanpa memiliki ijazah).

Relasi adalah hubungan antar *Entity* yang berfungsi sebagai hubungan yang mewujudkan pemetaan antar Entity. Macam-macam relasi itu sendiri antara lain :

1. *One To One (1:1)*

Relasi dari *Entity* satu dengan *Entity* dua adalah satu berbanding satu.

Contoh: Pada pelajaran privat, satu guru mengajar satu siswa dan satu siswa hanya diajar oleh satu guru.

2. *One To Many (1:m)*

Relasi antara *Entity* yang pertama dengan *Entity* yang kedua adalah satu berbanding banyak atau dapat pula dibalik, banyak berbanding satu. Contoh: Pada sekolah, satu guru mengajar banyak siswa dan banyak siswa diajar oleh satu guru.

Entity Relationship Diagram ini diperlukan agar dapat menggambarkan hubungan antar *Entity* dengan jelas, dapat menggambarkan batasan jumlah *Entity* dan partisipasi antar *Entity*, mudah dimengerti pemakai dan mudah disajikan oleh perancang Database. Untuk itu *EntityRelationshipDiagram* dibagi menjadi dua jenis model, yaitu :

1. *Conceptual Data Model (CDM)*

CDM adalah jenis model data yang menggambarkan hubungan antar tabel secara konseptual.

2. *Physical Data Model (PDM)*

PDM adalah jenis model data yang menggambarkan hubungan antar tabel secara fisikal.

d. Desain *input-output*

Merupakan tahapan untuk merancang desain antar muka input dan output sistem. Input berhubungan desain desain interface pengguna yang melakukan

inputan terhadap sistem, sedangkan output berhubungan dengan hasil dari proses sistem yang dapat berupa laporan atau informasi.

e. Desain Program (*pseudocode*)

Pseudocode adalah kode yang mirip dengan instruksi kode program yang sebenarnya. *Pseudocode* berbasis pada pemrograman sesungguhnya seperti COBOL, FORTRAN atau Pascal, sehingga lebih tepat digunakan untuk menggambarkan algoritma yang akan dikomunikasikan kepada *programmer*. *Pseudocode* akan memudahkan *programmer* untuk memahami dan menggunakannya, karena mirip dengan kode-kode program sebenarnya.

2.3.4. *Software Construction*

Menurut (IEEE, 2004) *software construction* adalah tahap pengkodean untuk membangun sistem sesuai dengan hasil desain sistem yang telah dirancang sebelumnya. Dalam proses ini digunakan beberapa *software* pendukung, seperti:

a. PHP

PHP adalah bahasa pemrograman yang didesain khusus untuk membuat halaman web. Notepad merupakan editor teks yang biasa digunakan. Fungsi-fungsi yang telah ada didalam PHP tidak *case sensitive* (membedakan huruf besar dengan huruf kecil), namun variabelnya *case sensitive*. Salah satu cara menuliskan script PHP adalah `<?php (script anda) ?>`.

b. Notepad++

Notepad++ adalah text editor yang digunakan oleh sebagian *programmer* untuk membuat program. Notepad++ merupakan salah satu software dengan free lisensi, sehingga penggunaanya bebas dari biaya. Salah satu bahasa pemrograman yang menggunakan editor notepad++ adalah PHP.

c. MYSQL

MYSQL merupakan *software* RDBMS (atau server database) yang mengelola database dengan sangat cepat, dapat menampung data dalam jumlah sangat besar, dapat diakses oleh banyak user (*multi-user*), dan dapat melakukan suatu proses secara sinkron atau berbarengan (*multi-threaded*). Beberapa alasan mengapa menggunakan MYSQL sebagai server database:

1. Fleksibel

MYSQL dapat digunakan untuk mengembangkan aplikasi desktop maupun aplikasi web dengan menggunakan teknologi yang bervariasi.

2. Performa Tinggi

MYSQL memiliki mesin query dengan performa tinggi, dengan demikian proses transaksional dapat dilakukan dengan sangat cepat. Hal ini terbukti dengan digunakannya MYSQL sebagai database dari beberapa aplikasi web yang memiliki *traffic* (lalu lintas) sangat tinggi.

3. Lintas Platform

MYSQL dapat digunakan pada *platform* atau lingkungan (dalam hal ini sistem operasi) yang beragam, bisa Microsoft Windows, Linux, atau UNIX.

4. Gratis

MYSQL dapat digunakan secara gratis, meskipun ada software MYSQL yang bersifat komersial.

5. Proteksi Data yang Handal

MYSQL menyediakan mekanisme yang *powerfull* untuk menangani hal tersebut, yaitu dengan menyediakan fasilitas manajemen user, enkripsi data, dan lain sebagainya.

6. Komunitas Luas

Karena banyaknya pengguna MYSQL, maka kita dapat mengikuti salah satu komunitas MYSQL. Hal ini dapat membantu sesama pengguna MYSQL dalam memecahkan masalah.

2.3.5. *Software Testing*

Menurut (IEEE, 2004) *software testing* adalah tahapan yang dilakukan untuk melakukan uji coba *functional* dan *non-functional*. Uji coba *functional* adalah cara yang digunakan untuk menemukan kesalahan fungsional *software* dioperasikan sesuai dengan prosedur yang ditetapkan. Sedangkan uji coba *non-functional* merupakan proses pengujian terhadap kode-kode program yang tidak bisa berjalan sesuai dengan harapan.

Software testing yang digunakan dalam penelitian ini menggunakan *black box testing*. *black box testing* adalah tipe testing yang menguji perangkat lunak dengan tidak mengetahui kinerja internalnya, sehingga para *tester* memandang perangkat lunak melalui sisi spesifikasi dan kebutuhan yang telah didefinisikan pada saat awal perancangan. Pada *black box testing* perangkat lunak yang *ditesting* akan diperiksa apakah telah memenuhi kebutuhan pengguna yang didefinisikan pada saat awal perancangan tanpa harus membongkar program kembali.