

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Sekilas Tentang Manajemen Proyek Perangkat Lunak**

##### **2.1.1 Definisi Manajemen Proyek Perangkat Lunak**

Proyek adalah urutan kegiatan yang unik, kompleks dan saling terkait, memiliki satu tujuan, dan tujuan harus diselesaikan dalam waktu tertentu, sesuai anggaran, dan memenuhi spesifikasi. Manajemen proyek perangkat lunak bertujuan agar perangkat lunak yang dibuat sampai ke tangan pelanggan (customer) tepat waktu dan sesuai dengan harapan pelanggan (customer).

Manajemen proyek perangkat lunak sangat dibutuhkan karena permasalahan yang sering timbul pada proyek perangkat lunak tanpa pengelolaan yang baik adalah pembengkakan biaya proyek dan waktu pengerjaan tidak sesuai rencana (molor waktu). Padahal waktu dan biaya biasanya sudah dianggarkan oleh pelanggan (customer) dan developer perangkat lunak harus mampu mengelolanya agar target perangkat lunak yang akan dibuat dapat dicapai. (Rosa A.S-M.Shalahuddin,2011)

##### **2.1.2 Metode Pengembangan Perangkat Lunak**

Pengembangan perangkat lunak dapat diartikan sebagai proses membuat suatu perangkat lunak baru untuk menggantikan perangkat lunak lama secara keseluruhan atau memperbaiki perangkat lunak yang telah ada. Agar lebih cepat dan tepat dalam mendeskripsikan solusi dan mengembangkan perangkat lunak,

juga hasilnya mudah dikembangkan dan dipelihara, maka pengembangan perangkat lunak memerlukan suatu metodologi khusus.

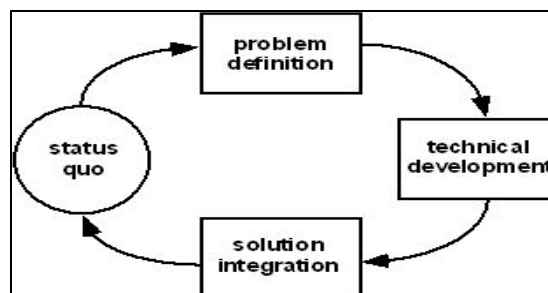
Metodologi pengembangan perangkat lunak adalah suatu proses pengorganisasian kumpulan metode dan konvensi notasi yang telah didefinisikan untuk mengembangkan perangkat lunak. Secara prinsip bertujuan untuk membantu menghasilkan perangkat lunak yang berkualitas.

### 2.1.3 Beberapa Metode Dasar

Berikut Metode-Metode Pengembangan Perangkat Lunak (Model Proses Pengembangan Perangkat Lunak) yaitu metode sekuensial linier, prototype, RAD, Fourth Generation Techniques (4GT) dan spiral. Selain metode-metode tersebut Roger S. Pressman (2002:45), mengungkapkan dua metode lagi yaitu metode Proses Perangkat Lunak Evolusioner (pertambahan, spiral, konkruen), dan metode formal. Berikut Metode-metode tersebut .:

#### 1) Model Sekuensial Linier

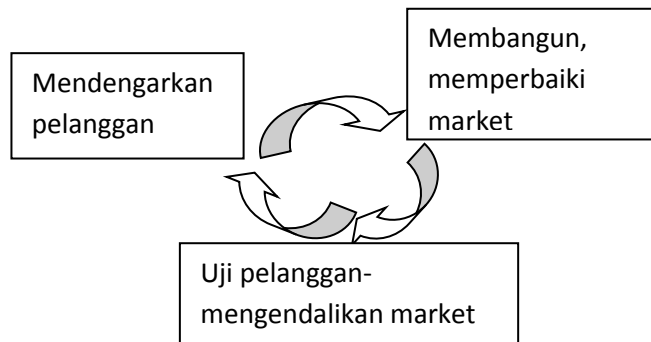
Sering juga disebut dengan “siklus kehidupan klasik” atau “model air terjun.” Penggambaran model ini :



Gambar Fase lingkaran pemecahan masalah

Metode pengembangan perangkat lunak dengan pendekatan pada perkembangan perangkat lunak yang sistematis dan sekuensial yang mulai pada tingkat dan kemajuan sistem pada seluruh analisis, desain, kode, pengujian, dan pemeliharaan.

## 2) Model Prototyping

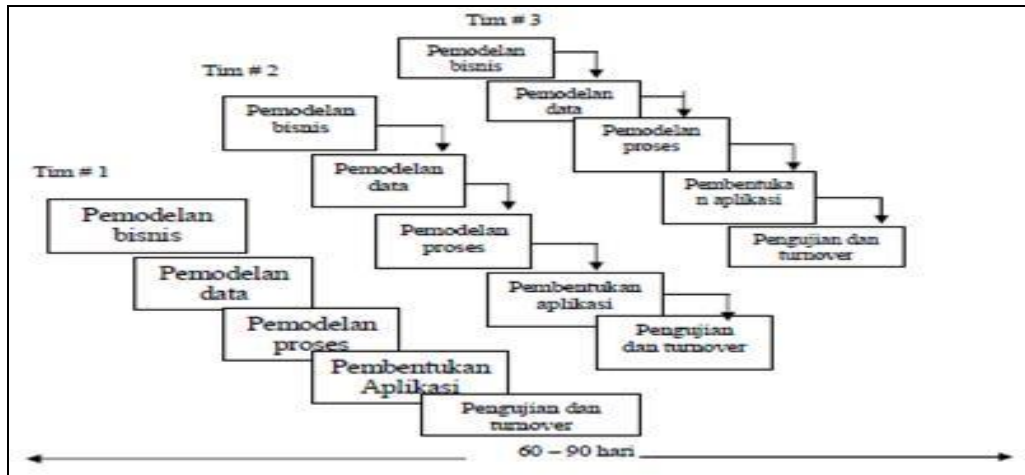


Gambar Prototype paradigma

Model ini dimulai dengan pengumpulan kebutuhan. Pendekatan *prototyping* model digunakan jika pemakai hanya mendefinisikan objektif umum dari perangkat lunak tanpa merinci kebutuhan *input*, pemrosesan dan *output*nya, sementara pengembang tidak begitu yakin akan efisiensi algoritma, adaptasi sistem operasi, atau bentuk antarmuka manusia-mesin yang harus diambil. Cakupan aktivitas dari *prototyping* model terdiri dari :

- a. Mendefinisikan objektif secara keseluruhan dan mengidentifikasi kebutuhan yang sudah diketahui.
- b. Melakukan perancangan secara cepat sebagai dasar untuk membuat prototype.
- c. Menguji coba dan mengevaluasi prototype dan kemudian melakukan penambahan dan perbaikan-perbaikan terhadap prototype yang sudah dibuat.

### 3) Model RAD (*Rapid Application Development*)



Gambar Model RAD

Merupakan model proses pengembangan perangkat lunak secara linear sequential yang menekankan pada siklus pengembangan yang sangat singkat/pendek. Jika kebutuhan dipahami dengan baik, proses RAD memungkinkan tim pengembangan menciptakan “sistem fungsional yang utuh” dalam periode waktu yang sangat pendek (kira-kira 60-90 hari).

### 4) Model Incremental

Model Incremental merupakan hasil kombinasi elemen-elemen dari model waterfall yang diaplikasikan secara berulang, atau bisa disebut gabungan dari Model linear sekuensial (waterfall) dengan Model Prototype. Elemen-elemen tersebut dikerjakan hingga menghasilkan produk dengan spesifikasi tertentu kemudian proses dimulai dari awal kembali hingga muncul hasil yang

spesifikasinya lebih lengkap dari sebelumnya dan tentunya memenuhi kebutuhan pemakai.

Model ini berfokus pada penyampaian produk operasional dalam Setiap pertambahannya. Pertambahan awal ada di versi stripped down dari produk akhir, tetapi memberikan kemampuan untuk melayani pemakai dan juga menyediakan platform untuk evaluasi oleh pemakai. Model ini cocok dipakai untuk proyek kecil dengan anggota tim yang sedikit dan ketersediaan waktu yang terbatas.

Pada proses Pengembangan dengan Model Incremental, perangkat lunak dibagi menjadi serangkaian increment yang dikembangkan secara bergantian.

#### **Kelebihan Model Incremental :**

- 1) Personil bekerja optimal.
- 2) Mampu mengakomodasi perubahan secara fleksibel, dengan waktu yang relatif singkat dan tidak dibutuhkan anggota/tim kerja yang banyak untuk menjalankannya.
- 3) Pihak konsumen dapat langsung menggunakan dahulu bagian-bagian yang telah selesai dibangun. Contohnya pemasangan data karyawan.
- 4) Mengurangi trauma karena perubahan sistem. Klien dibiasakan perlahan-lahan menggunakan produknya setiap bagian demi bagian.
- 5) Memaksimalkan pengembalian modal investasi konsumen.

#### **Kekurangan Model Incremental :**

- 1) Tidak cocok untuk proyek berukuran besar (lebih dari 200.000 baris coding).

- 2) Sulit untuk memetakan kebutuhan pemakai ke dalam rencana spesifikasi tiap-tiap hasil dari increment.

## **2.2 Metode Pengembangan Yang Digunakan**

### **2.2.1 Metode Agile**

Metodologi yang digunakan dalam menyelesaikan tugas akhir ini adalah menggunakan Metode Agile. Berikut penjelasannya :

Metodologi dalam perangkat lunak digunakan untuk merancang atau membangun suatu perangkat lunak, dengan perkembangan teknologi yang sangat pesat metodologi perangkat lunak juga terjadi perubahan atau penambahan requirements. Dari model waterfall sampai dengan model-model incremental.

Semua metodologi yang berkembang sebelumnya tidak mampu menangani kemungkinan perubahan atau penambahan requirements. Metode pengembangan perangkat lunak telah dilacak kembali pada tahun 1957. Pada tahun tersebut EA Edmonds telah memperkenalkan proses pengembangan perangkat lunak adaptif. “Lightweight” merupakan metode pengembangan perangkat lunak yang berkembang pada tahun 1990, sebagai reaksi terhadap apa yang disebut metode “heavyweight”. Yang ditandai dengan kritik mereka terhadap metode waterfall.

Pada tahun 90-an diperkenalkan dengan metodologi baru yang dikenal dengan nama *agile methods*. Kata Agile berarti bersifat cepat, ringan, bebas bergerak, waspada. Metodologi yang dikenal sebagai agile methods ini mengutamakan fleksibilitas terhadap perubahan-perubahan yang terjadi selama pengembangan. Bahkan perubahan ataupun penambahan pada saat fase terakhir pun teratasi apabila menggunakan metodologi ini.

Agile Methods dikembangkan karena pada metodologi tradisional terdapat banyak hal yang membuat proses pengembangan tidak dapat berhasil dengan baik sesuai tuntutan user. Konsep Agile Software Development dicetuskan oleh Kent Beck dan 16 rekannya dengan menyatakan bahwa Agile Software Development adalah cara membangun software dengan melakukannya dan membantu orang lain membangunnya sekaligus.

### **2.2.2 Prinsip Metode Agile**

Berikut adalah 12 prinsip dari metode agile menurut Agile Alliance, diperuntukan bagi mereka yang ingin berhasil dalam penerapan Agile Software Development: (Sumber : Principles behind the agile manifesto, kent beck, 2001).

1. Prioritas utama proses agile adalah memuaskan klien dengan menghasilkan perangkat lunak yang bermanfaat dengan cepat dan rutin.
2. Menerima perubahan kebutuhan, sekalipun diakhir pengembangan.
3. Penyerahan hasil/software dalam hitungan minggu bukan bulan.
4. Rekan bisnis/Pengusaha dan pengembang software harus bekerja sama selama proyek berlangsung.
5. Lingkungan proyek di sekitar harus nyaman dan memotivasi pengembang. Berikan dukungan dan lingkungan yang mereka butuhkan, dan beri kepercayaan kepada mereka untuk menyelesaikan pekerjaan dengan baik.
6. Komunikasi dengan berhadapan langsung adalah komunikasi yang efektif dan efisien.

7. Software yang berfungsi dengan baik adalah ukuran utama dari progress pengembangan software.
8. Dukungan yang stabil dari sponsor, pembangun, dan pengguna diperlukan untuk menjaga perkembangan yang berkesinambungan.
9. Memperhatikan keunggulan teknis yang berkesinambungan(terus menerus) dan design yang baik untuk meningkatkan kecerdasan software.
10. Kesederhanaan penting.
11. Arsitek. Kebutuhan dan desain yang bagus muncul dari tim yang mengatur dirinya sendiri.
12. Tim pengembang harus bekerjasama bagaimana untuk menjadi tim yang lebih efektif dalam mengembangkan software, kemudian menyesuaikan dan menelaraskan kebiasaan bekerja masing-masing tim.

### **2.2.3 Kelebihan dan Kekurangan Metode Agile**

#### **2.2.3.1 Kelebihan Metode Agile**

1. Meningkatkan kepuasan kepada klien.
2. Pembangunan system dibuat lebih cepat.
3. Mengurangi resiko kegagalan implementasi software dari segi non-teknis.
4. Jika pada saat pembangunan system terjadi kegagalan,kerugian dari segi materi relative kecil.



### **2.2.3.2 Kekurangan Metode Agile**

1. Developer harus selalu siap dengan perubahan, karena perubahan akan selalu diterima.
2. Agile tidak akan berjalan dengan baik jika komitmen tim kurang.
3. Tidak cocok dalam skala tim yang besar (>20 orang).
4. Perkiraan waktu release dan harga perangkat lunak sulit ditentukan.

## **2.3 Contoh Beberapa Aplikasi Manajemen Proyek**

Berikut diulas beberapa contoh aplikasi manajemen proyek yang ada, berikut dengan fitur-fitur yang melengkapinya sebagai bahan pembandingan untuk aplikasi yang akan dibangun.

### **2.3.1 Microsoft Project**

Microsoft Project adalah perangkat lunak manajemen proyek program yang dikembangkan dan dijual oleh Microsoft, yang dirancang untuk membantu seorang manager proyek dalam mengembangkan rencana, menetapkan sumber daya untuk tugas-tugas, pelacakan kemajuan, mengelola anggaran, dan menganalisis beban kerja.

Microsoft Project adalah aplikasi perusahaan ketiga berbasis Windows Microsoft, dan dalam beberapa tahun diperkenalkan menjadi perangkat lunak manajemen proyek berbasis PC yang dominan.

**Fitur :**

Proyek menciptakan anggaran berdasarkan pada pekerjaan tugas dan tingkat sumber daya. Sebagai sumber daya yang ditugaskan untuk tugas dan pekerjaan tugas diperkirakan, program menghitung biaya, sama dengan waktu kerja tingkat, yang gulungan sampai ke tingkat tugas dan kemudian untuk setiap tugas ringkasan dan akhirnya ke tingkat proyek. Definisi sumber daya (orang, peralatan dan bahan) dapat dibagi antara proyek menggunakan kolam sumber daya bersama.

Setiap sumber daya dapat memiliki kalender sendiri, yang mendefinisikan apa hari dan menggeser sumber daya yang tersedia. Tarif sumber daya yang digunakan untuk menghitung biaya tugas sumber daya yang digulung dan diringkaskan di tingkat sumber daya. Setiap sumber daya dapat ditugaskan untuk beberapa tugas dalam beberapa rencana dan tugas masing-masing dapat diberikan beberapa sumber, dan jadwal kerja aplikasi tugas berdasarkan ketersediaan sumber daya seperti yang didefinisikan dalam kalender sumber daya. Semua sumber daya dapat didefinisikan dalam label tanpa batas. Oleh karena itu tidak dapat menentukan berapa banyak produk jadi dapat diproduksi dengan jumlah yang diberikan bahan baku.

Hal ini membuat proyek Microsoft tidak cocok untuk memecahkan masalah bahan yang tersedia terbatas produksi. Software tambahan yang diperlukan untuk mengelola fasilitas kompleks yang memproduksi barang fisik.

Aplikasi membuat jalur kritis jadwal, dan rantai kritis dan acara metodologi rantai pihak ketiga add-ons juga tersedia. Jadwal dapat sumber daya diratakan, dan rantai yang divisualisasikan dalam grafik Gantt. Selain itu,

Microsoft Project dapat mengenali kelas yang berbeda dari pengguna. Kelas-kelas ini berbeda dari pengguna dapat memiliki berbeda tingkat akses ke proyek, pandangan, dan data lainnya. Objek kustom seperti kalender, pandangan, tabel, filter, dan bidang disimpan dalam suatu perusahaan global yang yang dimiliki oleh semua pengguna.

### **2.3.2 Colabtive**

Collabtive adalah web based project management software yang menggunakan lisensi open source. Collabtive dibuat oleh Philip Kiszka dan Eva Kiszka pada tahun 2007. Software ini menggunakan proprietary tools seperti Basecamp atau ActiveCollab. Collabtive sendiri ditulis menggunakan PHP5 dan menggunakan user interface berbasis Ajax. Selain itu juga mendukung hingga 30 bahasa berbeda dan terintegrasi dengan web services melalui XML API.

Fungsi lain yang dimiliki adalah penggunaan To do list, Milestones, Calendar, Messaging/Instant Messaging, File Management, Role based user permissions, Timetracking, Tagging, Search, Reporting (berbentuk file Microsoft Excel dan PDF), Exporting ke berbagai bentuk file seperti ZIP, XML, RSS, iCal dan vCard), Importing from Basecamp XML dan Multilanguage.

Adapun fitur-fitur utamanya adalah :

#### **Fungsi Inti :**

1. Tidak ada batasan proyek, milestone, tasklists, dan task
2. Tidak ada batasan anggota dengan profil pengguna
3. Manajemen pengguna berdasarkan hak akses (role)

4. Pesan dan pesan instan (chatting)
5. Manajemen file
6. Timetracking

**Pelaporan dan Notifikasi :**

1. Laporan Timetracker, log aktivitas dan pesan dalam format file PDF
2. Export profil pengguna sebagai vCard
3. RSS feed untuk pesan dan task
4. Sinkronisasi kalender untuk task melalui iCal
5. Laporan log aktivitas dan laporan Timetracker dalam format file excel
6. Notifikasi e-mail

**Lain-lain :**

1. Pencarian, tag assignment
2. Impor dari file BaseCamp
3. Tersedia lebih dari 30 bahasa

**2.3.3 Ganttter Project Management**

Ganttter adalah manajemen proyek berbasis web gratis. Anda dapat menganggap itu sebagai proyek berbasis web Microsoft. Ganttter membantu Anda untuk merencanakan dan memperkirakan proyek Anda (durasi, pekerjaan, biaya) dengan memisahkan proyek menjadi beberapa tugas; meninjau Anda berencana menggunakan Gantt chart; mengelola sumber daya proyek (pekerjaan atau materi) dan menetapkan tugas untuk mereka(to do list), mengelola kalender proyek (set

kerja kali, tambahkan libur dll), mengelola sumber daya kalender (waktu kerja, liburan dll); mengontrol kemajuan proyek Anda dengan menetapkan % nilai lengkap untuk setiap tugas; tinjauan kritis tugas.

## **2.4 Tool Untuk Membangun Aplikasi Web**

### **2.4.1 Web Server : Apache**

1. Merupakan webserver. Tempat php engine/processor berada. Tempat meletakkan file-file php dan database. Ketika user melakukan request http:// membuka suatu halaman, disinilah apache bekerja. Menjawab request tersebut dengan menampilkan halaman yang diminta.
2. Apache sama seperti PHP dan MySQL, Gratis.
3. Cross Platform Perbedaan fungsi antara PHP, MySQL dan Apache adalah, PHP merupakan bahasanya, MySQL adalah databasanya, dan Apache merupakan webserver yang dapat mengeksekusi script php dan menampilkannya kepada user, dan melalui apache lah php dapat mengolah data dan menyimpan data tersebut ke dalam database. (Loka Dwiartara, 2012)

### **2.4.2 Script Pemrograman:**

#### **1) HTML**

HTML merupakan singkatan dari HyperText Markup Language, adalah suatu bahasa yang digunakan untuk menulis halaman web. HTML dirancang untuk digunakan tanpa ketergantungan pada suatu platform tertentu. Dokumen HTML adalah suatu dokumen teks biasa, dan disebut markup language karena

mengandung tag tertentu yang digunakan untuk menentukan tampilan suatu teks dan tingkat kepentingan dari teks tersebut dalam suatu dokumen.

Pada dokumen HTML yang termasuk system hypertext, kita tidak harus membaca dokumen tersebut secara urut dari atas ke bawah atau sebaliknya, tetapi kita dapat menuju topik tertentu secara langsung dengan menggunakan teks penghubung yang akan membawa anda ke suatu topik atau dokumen lain secara langsung. (Setiawan,2012)

## 2) CSS

*Cascading Style Sheet* (CSS) merupakan salah satu bahasa pemrograman *web* untuk mengendalikan beberapa komponen dalam sebuah *web* sehingga akan lebih terstruktur dan seragam. Sama halnya styles dalam aplikasi pengolahan kata seperti Microsoft Word yang dapat mengatur beberapa style, misalnya heading, subbab, bodytext, footer, images, dan style lainnya untuk dapat digunakan bersama-sama dalam beberapa berkas (file). Pada umumnya CSS dipakai untuk memformat tampilan halaman *web* yang dibuat dengan bahasa HTML dan XHTML.

CSS dapat mengendalikan ukuran gambar, warna bagian tubuh pada teks, warna tabel, ukuran border, warna border, warna *hyperlink*, warna *mouse over*, spasi antar paragraf, spasi antar teks, margin kiri, kanan, atas, bawah, dan parameter lainnya. (Abdul Kadir, 2011).

### 3) Java Script

JavaScript adalah kode untuk menyusun halaman web yang memungkinkan dijalankan di sisi klien (pada browser yang digunakan pemakai). Karena dijalankan di sisi klien, maka JavaScript digunakan untuk membuat tampilan lebih dinamis. (Abdul Kadir, 2011).

### 4) PHP

PHP merupakan bahasa berbentuk skrip yang ditempatkan dalam server dan diproses di server. Hasilnyalah yang dikirimkan ke klien, tempat pemakai menggunakan browser.

Secara khusus, PHP dirancang untuk membentuk aplikasi web dinamis. Artinya, ia dapat membentuk suatu tampilan berdasarkan permintaan terkini. (Abdul Kadir, 2008)

### 5) Database Server : MySQL

MYSQL adalah database. Database sendiri merupakan suatu jalan untuk dapat menyimpan berbagai informasi dengan membaginya berdasarkan kategori-kategori tertentu. Dimana informasi-informasi tersebut saling berkaitan satu dengan yang lainnya.

MYSQL bersifat RDBMS (Relational Database Management System), yang memungkinkan seorang admin dapat menyimpan banyak informasi ke dalam tabel-tabel, dimana tabel-tabel tersebut saling berkaitan satu sama lain. Keuntungan RDBMS sendiri adalah kita dapat memecah database kedalam tabel-tabel yang berbeda. Setiap tabel memiliki informasi yang saling berkaitan dengan tabel yang lainnya.

Sama dengan PHP, MYSQL bersifat opensource, yang artinya semua orang dapat menggunakannya dengan gratis. MYSQL juga bersifat *Cross Platform*, yang artinya dapat digunakan under Windows ataupun under Linux. (Loka Dwiartara, 2012).

## **2.5 Pemodelan Sistem dengan UML**

UML (*Unified Modelling Language*) merupakan standar bahasa yang banyak digunakan didunia industri untuk membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek.

UML diaplikasikan untuk maksud tertentu, biasanya antara lain untuk :

1. Merancang perangkat lunak.
2. Sarana komunikasi antara perangkat lunak dengan proses bisnis.
3. Menjabarkan sistem secara rinci untuk analisa dan mencari apa yang diperlukan sistem.
4. Mendokumentasikan system yang ada, proses-proses dan organisasinya.

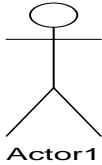

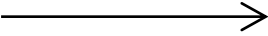
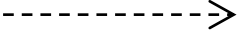
(Rosa A.S-M. Shalahuddin, 2011).



### 2.5.1 Usecase Diagram

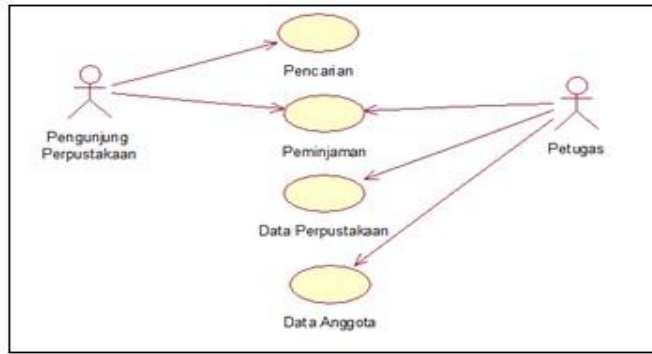
*Use case* diagram merupakan pemodelan untuk sistem informasi yang akan dibuat. *Use Case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. *Use Case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi – fungsi itu. Berikut adalah elemen dalam *use case* :

Tabel 2.1 Notasi *Usecase* Diagram

<i>Penjelasan</i>	<i>Notasi UML</i>
<i>Aktor : Mewakili peran orang, sistem yang lain atau alat ketika berkomunikasi dengan use case.</i>	
<i>Use Case : Abstraksi dari interaksi antara sistem dan actor</i>	
<i>Association : adalah abstraksi dari penghubung antara actor dan use case</i>	
<i>Generalisasi : menunjukkan spesialisasi actor untuk dapat berpartisipasi dalam use case</i>	

(Rosa A.S-M. Shalahuddin,2011)

Penamaan pada *Use Case* adalah nama didefinisikan sesimpel mungkin dan mudah dipahami. Berikut contoh *Use Case* pada gambar 3.1



Gambar 2.1 Contoh *Use case*

(Modul Pembelajaran Rekayasa Perangkat Lunak, Rosa A.S-M. Shalahuddin, 2011)

### 2.5.2 Activity Diagram

*Activity diagram* menggambarkan 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.

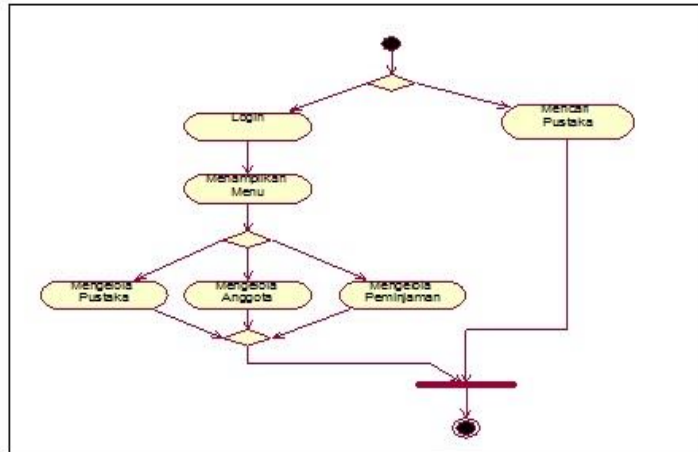
Berikut adalah elemen dalam *activity diagram* :

Simbol	Keterangan
●	Titik Awal
●	Titik Akhir
□	Activity
◇	Pilihan untuk pengambilan keputusan
—	Fork; Untuk menunjukkan kegiatan yang dilakukan secara paralel
⌋	Rake; menunjukkan adanya dekomposisi
⌋	Tanda Waktu
⌋	Tanda Penerimaan
⊗	Aliran Akhir (Flow Final)

Gambar 2.2 Notasi Activity Diagram

(Modul Pembelajaran Rekayasa Perangkat Lunak, Asril Pangroril,2011)

Berikut contoh *activity diagram* pada gambar 2.3

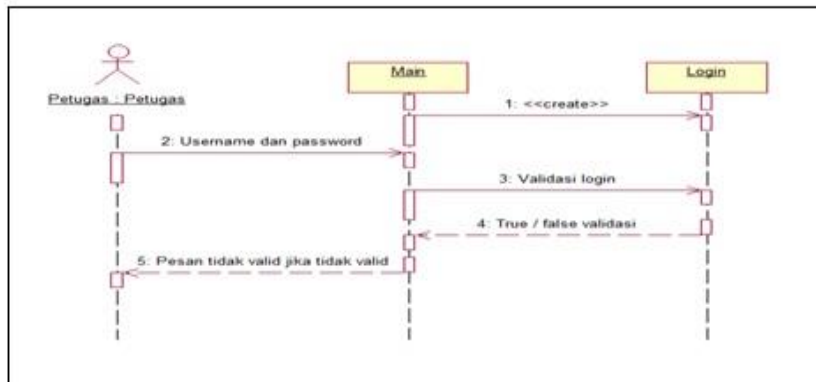


Gambar 2.3 Contoh *Activity Diagram*

(Modul Pembelajaran Rekayasa Perangkat Lunak, Asril Pangroril,2011)

### 2.5.3 *Sequence Diagram*

*Sequence diagram* menggambarkan kelakuan objek pada *Use Case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Berikut contoh *Sequence diagram* pada gambar 2.4

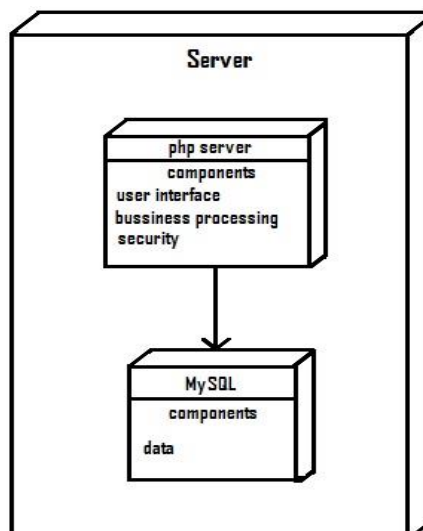


Gambar 2.4 Contoh *Sequence* Diagram

(Modul Pembelajaran Rekayasa Perangkat Lunak, Ratno Kustiawan,2011)

#### 2.5.4 *Deployment* Diagram

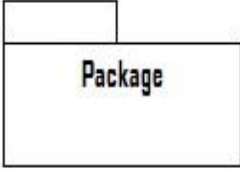
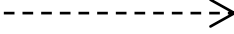
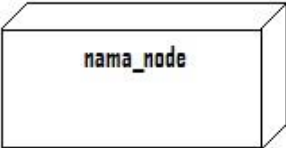

*Deployment* diagram menunjukkan konfigurasi komponen dalam proses eksekusi aplikasi.



Gambar 2.5 Contoh *Deployment* Diagram

Berikut adalah simbol-simbol yang ada pada *deployment* diagram:

Tabel 2.2 Notasi *Deployment Diagram*

<b><i>Penjelasan</i></b>	<b><i>Notasi UML</i></b>
<i>Package merupakan sebuah bungkus dari satu atau lebih node</i>	Package 
<i>Kebergantungan antar node, arah panah mengarah pada node yang dipakai</i>	dependency 
<i>Biasanya mengacu pada perangkat keras(hardware), perangkat lunak yang tidak dibuat sendiri(software), jika di dalam node disertakan komponen untuk mengkonsistensikan rancangan maka komponen yang diikutsertakan harus sesuai dengan komponen yang telah didefinisikan sebelumnya pada diagram komponen.</i>	Node 
<i>Relasi antar node</i>	Link 

(Rekayasa Perangkat Lunak, Rosa A.S-M.Shalahuddin, 2011)