

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Perijinan Disperindag kota Palangkaraya**

Sesuai dengan Peraturan Daerah nomor 17 tahun 2002, ditetapkan pelayanan perijinan Dinas Perindustrian dan Perdagangan adalah meliputi bidang industri dan perdagangan sebagaimana disebutkan dalam bab III pasal 4 ayat 1 dan 2. Adapun sistem aplikasi yang akan dibangun adalah aplikasi manajemen perijinan yang meliputi seluruh perijinan di Disperindag kota Palangkaraya saja.

Adapun pos-pos petugas yang berkaitan dengan jalannya pelayanan perijinan di Disperindag kota Palangka Raya adalah sebagai berikut :

##### **1. Bagian Informasi**

Merupakan tahap pertama yang harus dilakukan pemohon terkait informasi tentang apa saja persyaratan yang harus pemohon penuhi untuk memenuhi proses pembuatan ijin tersebut.

##### **2. Bagian Pendaftaran**

Merupakan tahap untuk pengambilan formulir pendaftaran pembuatan ijin usaha, agar pemohon dapat melengkapi proses pembuatan ijin usaha nya.

##### **3. Bagian Penetapan**

Merupakan tahap pengecekan kelengkapan dokumen persyaratan apakah sesuai atau tidak, selanjutnya akan di berikan kepada kepala Disperindag kota Palangka Raya untuk disahkan dan akan diberikan kembali ke pemohon.

Macam-macam dan jenis ijin yang diterbitkan oleh Diperindag kota Palangkaraya yaitu :

1. SIUP ( Surat Ijin Usaha Perdagangan) dan persyaratan dapat dilihat di lampiran 1
2. SITU ( Surat Ijin Tempat Usaha ) dan persyaratan dapat dilihat di lampiran 2, dan

3. Reklame dan persyaratan dapat dilihat di lampiran 3

Berdasarkan peraturan daerah nomor 17 tahun 2002 tentang retribusi perijinan tertentu di Disperindag kota Palangkaraya yaitu :

1. Ijin usaha bertarif yaitu merupakan ijin usaha yang dikenakan pajak tempat usahanya, sehingga untuk jenis ijin ini dikenakan biaya sesuai dengan besar pajak tempatnya. Tarif pajak sendiri merupakan iuran kepada negara yang dipungut berdasarkan undang-undang perpajakan yaitu pasal 23A UUD 1945 amandemen ke-4 “ Pajak dan pungutan lain yang bersifat memaksa untuk keperluan Negara diatur oleh undang-undang “, jenis ijin yang dikenakan pajak tempat yaitu seperti surat ijin mendirikan bangunan (IMB), surat ijin gangguan atau HO, surat ijin usaha jasa konstruksi (SUJK), dan ijin reklame, dan
2. Ijin non tarif merupakan jenis ijin yang tidak dikenakan biaya pajak, yaitu seperti surat ijin usaha perdagangan (SIUP), tanda daftar perusahaan (TDP), dan surat ijin tempat usaha (SITU) Konsep pelayanan satu atap adalah adalah pola pelayanan umum yang dilakukan secara terpadu pada suatu tempat oleh beberapa instansi pemerintah yang bersangkutan sesuai dengan kewenangan masing-masing. Penggunaan konsep satu atap ini digunakan untuk menghubungkan kantor atau bagian-bagian yang posisinya berjauhan. Pelayanan satu atap ini juga dinilai dapat meminimalkan atau bahkan menghilangkan pungutan-pungutan liar yang ada termasuk korupsi dan biaya-biaya yang tidak resmi dari calo-calo. Pola pelayanan satu atap dapat juga dikembangkan di pemerintah daerah dalam rangka mempercepat proses pemberian layanan publik kepada masyarakat.

## 2.2 Sistem

Sistem dapat didefinisikan sebagai seperangkat elemen yang digabungkan satu dengan lainnya untuk suatu tujuan bersama. Kumpulan elemen terdiri dari manusia, mesin, prosedur, dokumen,

data, atau elemen lain yang terorganisir dari elemen-elemen tersebut. Elemen sistem disamping berhubungan satu sama lain, juga berhubungan dengan lingkungannya untuk mencapai tujuan yang telah ditentukan sebelumnya. Sebuah sistem terdiri dari bagian-bagian yang saling berkaitan yang beroperasi bersama untuk mencapai beberapa sasaran atau maksud. Sistem adalah himpunan dari unsur-unsur yang saling berkaitan sehingga membentuk suatu kesatuan yang utuh dan terpadu menurut McLeod (2001), Karakteristik di dalam sistem terdiri dari :

- a. Organisasi : Mencakup struktur dan fungsi organisasi.
- b. Interaksi : Saling keterhubungan antara bagian yang satu dengan lainnya.
- c. Interdependensi : Bagian yang satu mempunyai ketergantungan dengan bagian yang lainnya.
- d. Integritas : Suatu keterpaduan antara subsistem-subsistem untuk mencapai tujuan.
- e. *Main Objective* (Tujuan Utama) : Pemusatan tujuan yang sama dari masing-masing subsistem.

Implementasi sistem informasi dibutuhkan perangkat lunak (Software) dan perangkat keras (Hardware) yang disusun sedemikian rupa sehingga menghasilkan informasi yang dibutuhkan oleh pengguna.

### **2.3 Metodologi Pengembangan Sistem**

Metodologi pengembangan sistem yang digunakan adalah pendekatan SDLC. SDLC sendiri adalah *System development life cycle*, yaitu suatu proses standar yang diikuti untuk melaksanakan seluruh langkah yang diperlukan untuk menganalisa, merancang, dan mengimplementasikan sistem informasi Jogiyanto (2010).

SDLC memiliki empat fase dasar yaitu *planning*, *analysis*, *design* dan *implementation*. Setiap fase itu sendiri terdiri dari serangkaian langkah yang menggunakan cara tertentu dalam

menghasilkan goal yang dicapai. Pada poin berikut akan dijelaskan secara singkat dari keempat fase tersebut.

### 1. *Analysis*

Fase *analysis* adalah penelitian atas sistem yang telah ada dengan tujuan untuk merancang sistem yang baru atau diperbaharui.

### 2. *Design*

Fase *design* yaitu kegiatan untuk memberikan gambaran secara umum kepada user tentang sistem yang baru seperti, *storyboard* yang merupakan uraian visual atau gambaran dan audio yang memberi penjelasan dari masing-masing alur dalam sistem yang dibuat.

### 3. *Implementation*

Fase final pada SDLC ini adalah fase *implementation*, yaitu pada saat sistem telah selesai dibuat. Implementasi merupakan kegiatan memperoleh dan mengintegrasikan sumber daya fisik dan konseptual yang menghasilkan suatu sistem yang bekerja. Tahap implementasi adalah tahap dimana semua elemen dan aktivitas sistem disatukan menjadi sebuah aplikasi.

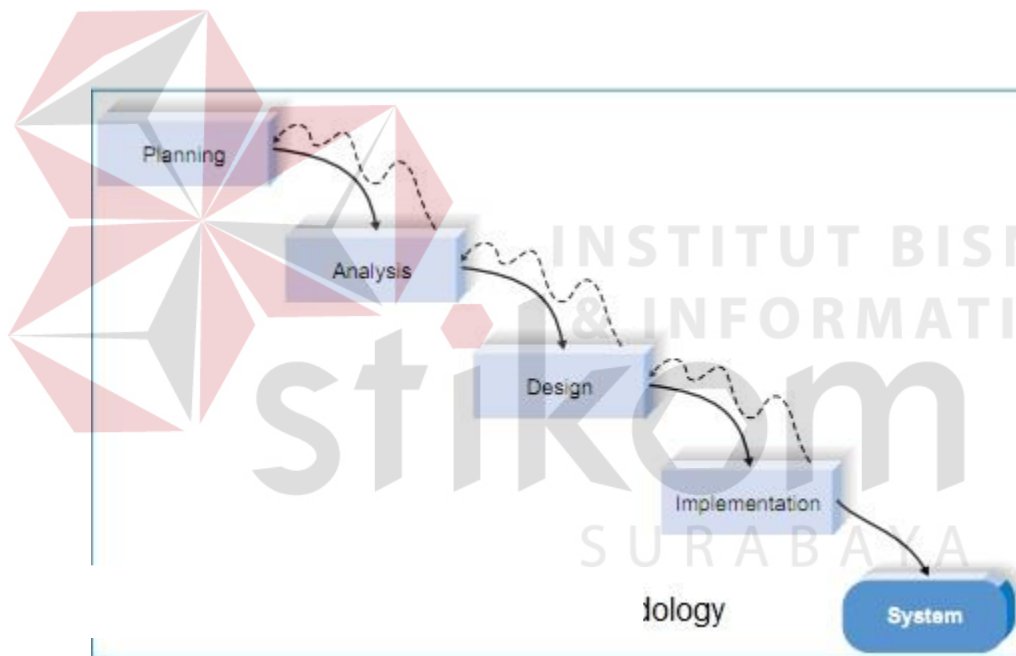
Terdapat banyak cara untuk mengkategorikan sebuah metodologi tergantung dengan fokus pada proses bisnis atau data yang mendukung bisnis. Namun dalam pengerjaan Tugas Akhir ini menggunakan metodologi berorientasi pada objek untuk mencoba menyeimbangkan fokus antara proses dan data dengan menggabungkan keduanya menjadi satu model. Penggunaan berbasis objek ini dapat mengkombinasikan data dan fungsionalitas dan proses pendekatan berbasis objek pembangun utama sistem perangkat lunak adalah objek atau kelas.

### 2.3.1 Desain Terstruktur

Metode desain terstruktur mengadopsi langkah-langkah pendekatan terhadap SDLC yang bergerak dari satu tahap ke tahap selanjutnya. Berikut pengembangan dari desain terstruktur Jogiyanto (2010):

#### 2.3.1.1 Waterfall

Struktur asli dari metode desain terstruktur adalah metode pengembangan *waterfall*. Dengan metode pengembangan *waterfall*, analis dan pengguna memproses melewati tahap demi tahap. Untuk lebih jelasnya dapat dilihat pada gambar 2.1.

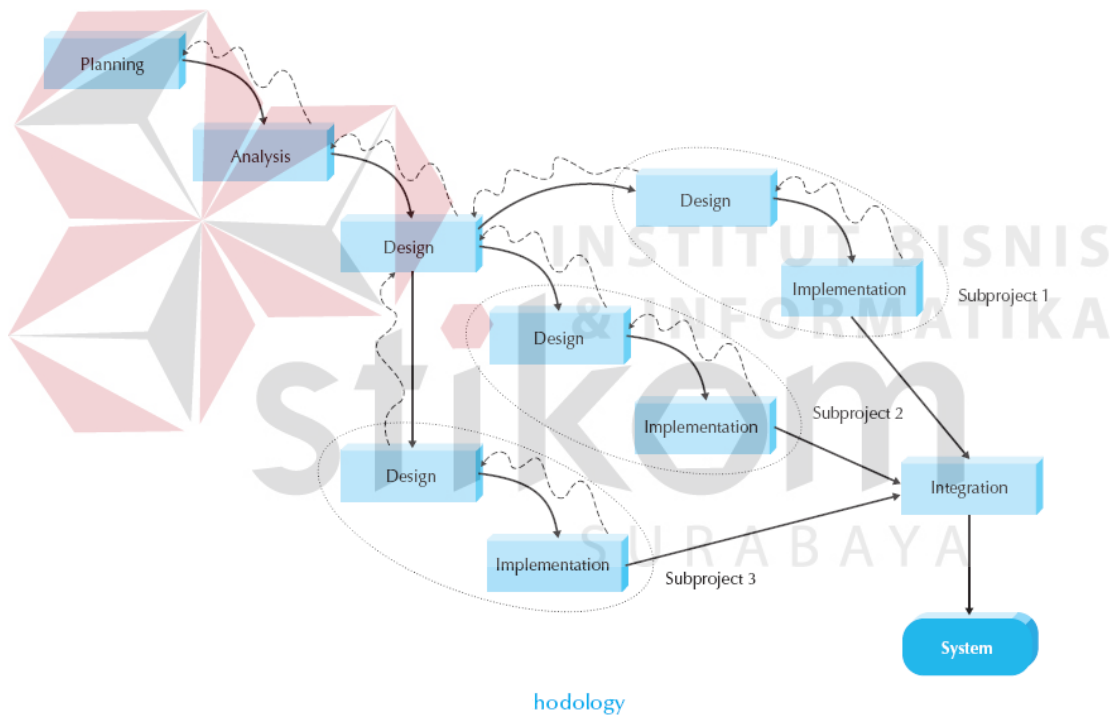


Gambar 2.1 *Waterfall Development*

Kelebihan dari menggunakan metode *waterfall* yaitu karena sistem ini mengidentifikasi kebutuhan sistem jauh sebelum melakukan pemrograman dan berguna untuk meminimalisir terjadinya perubahan pada kebutuhan sistem yang akan dibuat. Namun kekurangan dari sistem ini yaitu desain dari sistem yang akan dibuat harus selesai secara keseluruhan sebelum *programming* aplikasi berjalan dan hal tersebut membutuhkan waktu lama dalam tahap analisis sistem.

### 2.3.1.2 *Parallel*

Metode ini digunakan untuk mengatasi masalah penundaan yang lama antara analisis dan pengiriman pada sistem. Sebagai ganti dari desain sistem dan implementasi yang berjalan secara berurutan, metode ini melakukan desain umum untuk sistem secara keseluruhan kemudian membagi *project* menjadi serangkaian *sub project* kecil yang dapat dirancang dan dilaksanakan secara *parallel*. Ketika semua *sub project* telah selesai maka diintegrasikan menjadi sehingga menjadi kesatuan sistem *project* yang utuh. Untuk lebih jelasnya dapat dilihat pada gambar 2.2.



Gambar 2.2 *Parallel Development*

Kelebihan dari metode ini yaitu dapat mengurangi waktu dalam pembuatan sistem serta lebih sedikit kemungkinan terjadinya perubahan dalam lingkungan bisnis yang dibuat. Namun metode ini juga memiliki kekurangan yaitu terkadang *sub project* yang dikerjakan tidak

sepenuhnya independen tetapi masih terhubung dengan *sub project* yang lainnya sehingga pada akhir *project* membutuhkan integrasi sistem yang besar.

### **2.3.2 Desain Berbasis Objek**

Metode desain berbasis objek yang melakukan pendekatan terhadap SDLC yang berorientasi terhadap objek untuk melanjutkan ke tahap selanjutnya. Berikut pengembangan dari desain berbasis objek:

#### **2.3.2.1 Unified Modelling Language (UML)**

Menurut Nugroho (2005:16), pemodelan *visual* adalah proses penggambaran informasi-informasi secara grafis dengan notasi-notasi baku yang telah disepakati sebelumnya. Notasi-notasi baku sangat penting demi suatu alasan komunikasi. Dengan notasi-notasi pemodelan yang bersifat baku komunikasi yang baik akan terjalin dengan mudah antar anggota tim pengembang sistem/perangkat lunak dan antara anggota tim pengembang dengan para pengguna. UML dapat digambarkan sebagai bahasa pemodelan visual umum untuk memvisualisasikan, menentukan dan membangun perangkat lunak dalam sebuah dokumen.

#### **2.3.2.2 Use Case Diagram**

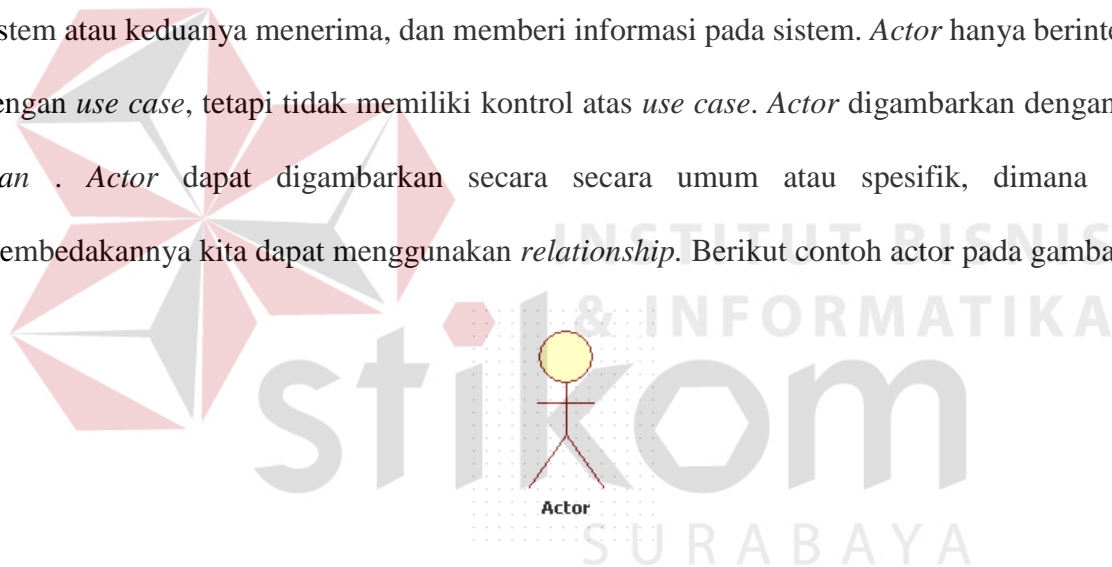
*Use-case diagram* merupakan model diagram UML yang digunakan untuk menggambarkan *requirement* fungsional yang diharapkan dari sebuah sistem. *Use-case diagram* menekankan pada “siapa” melakukan “apa” dalam lingkungan sistem perangkat lunak akan dibangun. *Use-case diagram* sebenarnya terdiri dari dua bagian besar; yang pertama adalah *use case diagram* (termasuk gambar *use case dependencies*) dan *use case description*.

*Use-case diagram* adalah gambaran *graphical* dari beberapa atau semua *actor*, *use-case*, dan interaksi diantara komponen-komponen tersebut yang memperkenalkan suatu sistem yang akan dibangun. *Use-case diagram* menjelaskan manfaat suatu sistem jika dilihat menurut

pandangan orang yang berada di luar sistem. Diagram ini menunjukkan fungsionalitas suatu sistem atau kelas dan bagaimana sistem tersebut berinteraksi dengan dunia luar. Komponen pembentuk Use Case Diagram:

## 1. Actor

Pada dasarnya *actor* bukanlah bagian dari *use case diagram*, namun untuk dapat terciptanya suatu *use case diagram* diperlukan beberapa *actor*. *Actor* tersebut mempresentasikan seseorang atau sesuatu (seperti perangkat, sistem lain) yang berinteraksi dengan sistem. Sebuah *actor* mungkin hanya memberikan informasi inputan pada sistem, hanya menerima informasi dari sistem atau keduanya menerima, dan memberi informasi pada sistem. *Actor* hanya berinteraksi dengan *use case*, tetapi tidak memiliki kontrol atas *use case*. *Actor* digambarkan dengan *stick man*. *Actor* dapat digambarkan secara umum atau spesifik, dimana untuk membedakannya kita dapat menggunakan *relationship*. Berikut contoh actor pada gambar 2.3



Gambar 2.3 Actor pada Use Case Diagram

## 2. Use Case

*Use case* adalah gambaran fungsionalitas dari suatu sistem, sehingga *customer* atau pengguna sistem paham dan mengerti mengenai kegunaan sistem yang akan dibangun.

Catatan : *Use case diagram* adalah penggambaran sistem dari sudut pandang pengguna sistem tersebut (*user*), sehingga pembuatan *use case* lebih dititikberatkan pada



fungsionalitas yang ada pada sistem, bukan berdasarkan alur atau urutan kejadian. Berikut contoh *use case* pada gambar 2.4



Gambar 2.4 Use Case pada Use Case Diagram

### 2.3.2.3 Activity Diagram

*Activity diagram* memiliki pengertian yaitu lebih fokus kepada menggambarkan proses bisnis dan urutan aktivitas dalam sebuah proses. Dipakai pada *business modeling* untuk memperlihatkan urutan aktifitas proses bisnis. Memiliki struktur diagram yang mirip *flowchart* atau *data flow diagram* pada perancangan terstruktur. Memiliki pula manfaat yaitu apabila kita membuat diagram ini terlebih dahulu dalam memodelkan sebuah proses untuk membantu memahami proses secara keseluruhan. Dan *activity* dibuat berdasarkan sebuah atau beberapa *use case* pada *use case diagram* (Felici, 2004). Terdapat beberapa hal penting yang harus diketahui, yaitu ;

- 1 Activity menggambarkan sebuah pekerjaan atau tugas dalam workflow, pada UML activity digambarkan dengan simbol kotak. Berikut contoh *symbol use case* pada gambar 2.5.



- 2 Start state dengan tegas menunjukkan dimulainya suatu workflow pada sebuah activity diagram. Hanya terdapat satu start state dalam sebuah workflow serta pada UML, start state digambarkan dengan simbol lingkaran yang solid

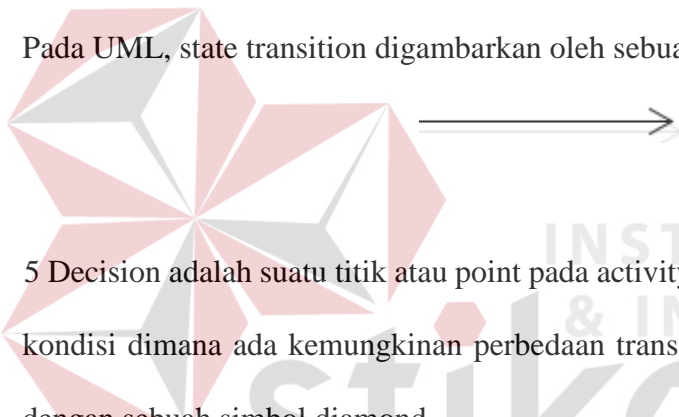


3 End state menggambarkan akhir atau terminal dari pada sebuah activity diagram. Bisa terdapat lebih dari satu end state pada sebuah activity diagram. Pada UML, end state digambarkan dengan simbol sebuah bull's eye

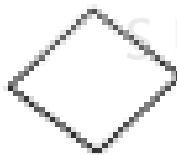


4 State transition menunjukkan kegiatan apa berikutnya setelah suatu kegiatan sebelumnya.

Pada UML, state transition digambarkan oleh sebuah solid line dengan panah



5 Decision adalah suatu titik atau point pada activity diagram yang mengindikasikan suatu kondisi dimana ada kemungkinan perbedaan transisi. Pada UML, decision digambarkan dengan sebuah simbol diamond



6 Obyek swimlane untuk menggambarkan objek mana yang bertanggung jawab untuk aktivitas tertentu atau biasa disebut entitas pada *work flow*.

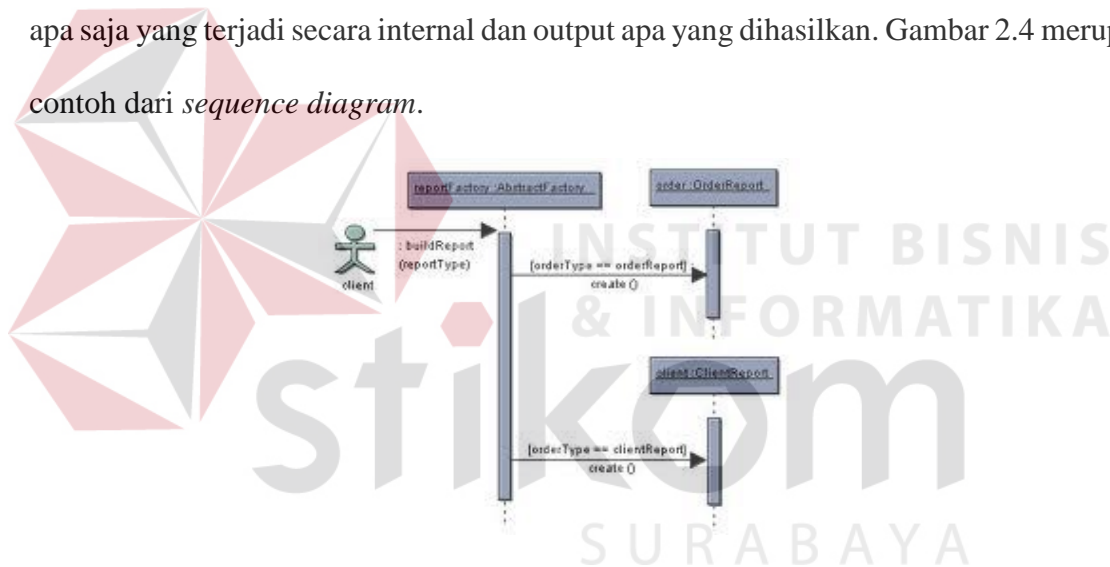


### 2.3.2.4 Sequence Diagram

*Sequence diagram* adalah interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu.

*Sequence diagram* terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait).

*Sequence diagram* biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah *event* untuk menghasilkan *output* tertentu. Diawali dari apa yang men-*trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan output apa yang dihasilkan. Gambar 2.4 merupakan contoh dari *sequence diagram*.



Gambar 2.6 Sequence Diagram

Sequence diagram biasanya digunakan untuk tujuan analisa dan desain, memfokuskan pada identifikasi *method* didalam sebuah *system*.

### 2.3.2.5 Class Diagram

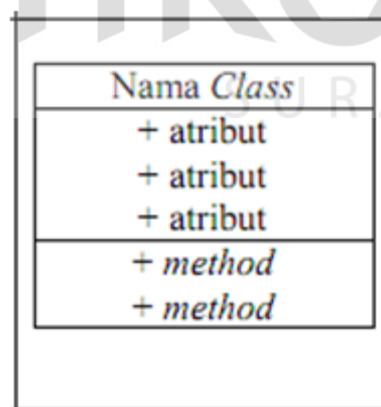
kumpulan objek-objek dengan dan yang mempunyai struktur umum, behavior umum, relasi umum, dan semantic/kata yang umum. Class-class ditentukan/ditemukan dengan cara

memeriksa objek-objek dalam sequence diagram dan collaboration diagram. Sebuah class digambarkan seperti sebuah bujur sangkar dengan tiga bagian ruangan. Class sebaiknya diberi nama menggunakan kata benda sesuai dengan domain/bagian/kelompoknya (Whitten L. Jeffery et al, 2004).

Elemen-elemen class diagram dalam pemodelan UML terdiri dari: Class-class, struktur class, sifat class (*class behavior*), perkumpulan/gabungan (*association*), pengumpulan/kesatuan (*agregation*), ketergantungan (*dependency*), relasi-relasi turunannya, keberagaman dan indikator navigasi, dan *role name* (peranan/tugas nama).

Simbol-simbol pada class diagram:

1. *Class* adalah blok-blok pembangun pada pemrograman berorientasi obyek. Sebuah *class* digambarkan sebagai sebuah kotak yang terbagi atas 3 bagian. Bagian atas adalah bagian nama dari *class*. Bagian tengah mendefinisikan *property*/atribut *class*. Bagian akhir mendefinisikan *method* dari sebuah *class*.



.Gambar 2.7 Class yang berisi atribut dan *method*

2. Association : Sebuah asosiasi merupakan sebuah relationship paling umum antara 2 class dan dilambangkan oleh sebuah garis yang menghubungkan antara 2 class. Garis ini bisa

melambangkan tipe-tipe relationship dan juga dapat menampilkan hukum-hukum multiplisitas pada sebuah relationship.(Contoh: One-to-one, one-to-many,many-to-many).

1..n Owned by 1

3. composition: Jika sebuah class tidak bisa berdiri sendiri dan harus merupakan bagian dari class yang lain, maka class tersebut memiliki relasi Composition terhadap class tempat dia bergantung tersebut. Sebuah relationship composition digambarkan sebagai garis dengan ujung berbentuk jajaran genjang berisi/solid.



4. Dependency : Kadangkala sebuah class menggunakan class yang lain. Hal ini disebut dependency. Umumnya penggunaan dependency digunakan untuk menunjukkan operasi pada suatu class yang menggunakan class yang lain. Sebuah dependency dilambangkan sebagai sebuah panah bertitik-titik.



5. Aggregation : Aggregation mengindikasikan keseluruhan bagian relationship dan biasanya disebut sebagai relasi.



Desain yang diambil dalam pengerjaan Tugas Akhir ini yaitu menggunakan desain berbasis objek. Pada penelitian ini dilakukan mulai dari tahap perencanaan, analisa, desain dan implementasi berurutan dan diimplementasikan kedalam sistem. Apabila terdapat kekurangan

maka akan diulang kembali hingga terpenuhi kebutuhan dari perusahaan. Keuntungan menggunakan desain berbasis objek ini yaitu: program lebih mudah dibaca dan dipahami, dan pemrograman berorientasi objek dapat mengontrol kerumitan program, pemodelan dapat digunakan sebagai dasar untuk pengukuran, penjadwalan atau validasi, mudah dimengerti oleh pebisnis, jadi merupakan sebuah jembatan penghubung bagi pengembang software dengan pemakai.

## **2.4 Pengembangan Aplikasi**

### **2.4.1 Aplikasi Berbasis Desktop**

Desktop based adalah suatu aplikasi yang dapat berjalan sendiri atau independen tanpa menggunakan browser atau koneksi internet disuatu komputer otonom dengan *operating system* atau *platform* tertentu seperti: Microsoft office, media player. Dan lain-lain. Keunggulan desktop base yaitu berjalan dengan independen, tidak memerlukan koneksi internet karena semua file yang diperlukan untuk menjalankan aplikasinya sudah terinstall sebelumnya, dapat dengan mudah memodifikasi settingannya, dan proses nya cepat seperti aplikasi SPSS, *Browser*, Winamp, dan lain-lain. Bahasa pemrograman yang digunakan seperti *Visual Basic* dan *Visual Net*, yang merupakan suatu himpunan dari aturan sintaks dan semantic yang dipakai untuk mendefinisikan program komputer.

### **2.4.2 Aplikasi Berbasis Web**

Web based adalah suatu aplikasi yang dapat berjalan dengan menggunakan basis teknologi web atau browser. Aplikasi ini dapat diakses dimana saja asalkan ada koneksi internet yang mendukung, tanpa perlu melakukan penginstallan di komputer masing-masing seperti pada aplikasi desktop, cukup dengan membuka browser dan menuju ke tempat server aplikasi tersebut

dipasang. Bahasa pemrograman web yang digunakan saat ini seperti PHP atau *Hypertext*, menurut Prasetyo (2004) PHP atau *Hypertext Preprocessor* merupakan Bahasa pemrograman yang memungkinkan para *web developer* membuat aplikasi web dinamis dengan cepat. Berbagai macam Bahasa pemrograman web saat ini seperti HTML, PHP, Java, VB.Net. Website dinamis yang ada saat ini Contohnya seperti: google word, webmessenger, game flash, dan lain-lain. Untuk pembuatan aplikasi di Disperindag kota Palangkaraya digunakan berbasis web dan menggunakan bahasa pemrograman PHP. Keuntungan menggunakan bahasa pemrograman ini yaitu :

1. Cross Platfrom, yang bisa digunakan di berbagai sistem operasi seperti windows, linux, mac os, dll.
2. Codeless, bahasa pemrograman yang digunakan cukup mudah.
3. Mendukung terhadap berbagai jenis database seperti MySql, Oracle, PostgrSQL, dll.
4. Freeware, yaitu PHP ini bersifat gratis atau open source.

Kekurangan menggunakan bahasa pemrograman ini yaitu :

1. PHP tidak mengenal Package atau plugin.
2. PHP memiliki kelemahan keamanan sehingga harus berhati-hati dalam melakukan pemrograman dan konfigurasi PHP.

Berdasarkan pengoperasiannya, ada 2 jenis website secara umum, yaitu website statis (*static Website*) dan website dinamis (*Dynamic Website*) menurut Abdul Kadir, (2005).

#### 1. *Static Website* (Website Statis)

Website statis hanya memiliki frond end, yaitu halaman-halaman situs yang bisa diakses oleh pengunjung. Biasanya website jenis ini hanya dibuat dengan kode HTML (*hypertext markup language*) dan biasanya dilengkapi dengan *script* dan *flash* untuk memperindah tampilan. Konten website statis ini tidak dapat diupdate, jika ingin merubah isi website statis harus

merubah kode-kode HTMLnya dan hanya dilakukan oleh orang yang bisa menggunakan Bahasa HTML. Website statis hanya bisa mengirimkan data ke dalam database yang berbentuk *xml*, sedangkan untuk memanggil data masih belum bisa. Website statis biasanya berupa *company profile*, brosur online, atau situs-situs yang berisi informasi yang tidak perlu diubah.

## 2. *Dynamic Website* (Website Dinamis)

Website dinamis adalah web yang content atau isinya dapat berubah-ubah setiap saat dan tanpa harus merubah kode-kode atau coding. Website dinamis pada umumnya menggunakan *Compact Management System (CMS)* tertentu dengan penyimpanan data di database.

Website dinamis memiliki berbagai tipe dan terus berkembang berdasarkan fungsi, fitur, dan tujuan pembuatannya, antara lain:

### a. *News Site* (Situs Berita)

Website ini biasanya berisi artikel-artikel atau berita-berita yang diupdate secara rutin. Pada beberapa situs, pengunjung atau pengguna bisa meninggalkan komentar. Seperti : Detiknews, Antaranews dan BBC.

### b. *Social Network Site* (Situs Jejaring Sosial)

Website ini digunakan untuk berinteraksi social, seperti Facebook, Instagram, Twitter, dan lain-lain.

### c. *Forum*

Website ini dibuat khusus agar para member dapat berdiskusi sesuai dengan topik-topik yang telah ditetapkan. Seperti : Kaskus, Indowebster, dan lain-lain.

### d. *E-Commerce* (Toko Online)

Website ini dibuat khusus untuk menjual produk secara online yang pada umumnya dilengkapi dengan *shopping cart* (keranjang belanja untuk memudahkan pengguna atau pengunjung



berbelanja. Tapi sebagian juga hanya berupa catalog online yang lengkap dengan detail dan harga produk untuk melakukan pembelian dapat dilakukan melalui email atau telepon.

e. *Blog*

Website yang bisa juga disebut diari online karena pemilik (*individu atau grup*) dapat mengupdate artikel, baik tulisan, gambar, ataupun file multimedia lain secara rutin, dimana semua entri tersusun berurutan dan memiliki fasilitas komentar buat pengunjung. Secara standar blog juga telah dilengkapi dengan sidebar yang full custom, fasilitas *RSS* dan *permalink*.

f. *Corporate/Company Website*

Website yang berisi informasi umum dan kegiatan suatu perusahaan. Diantaranya mengenai bisnis, visi dan misi, *our service* atau *product*, *our client*, dan lain-lain. Website ini digunakan sebagai pengganti company profile konvensional bentuk cetakan.

Beberapa tipe web untuk pengembangan web yaitu:

- Web 1.0

Web 1.0 secara umum dikembangkan untuk pengaksesan informasi dan memiliki sifat yang sedikit interaktif. Sifat dari web 1.0 adalah *read*. Ciri-ciri umum yang mencolok yaitu *consulf*, *surf*, dan *search*. Jadi web 1.0 ini hanya digunakan untuk browsing atau mencari informasi tertentu. Beberapa ciri khas dari web1.0 antara lain: halaman statis, penggunaan *framesets*, dan online *guestbook*. Situs web yang dikategorikan ke jenis ini seperti *cnn.com* dan *bhinneka.com*.

- Web 2.0

Web 2.0 pertama kali diperkenalkan oleh O'Reilly pada tahun 2004 sebagai teknologi web generasi kedua yang mengedepankan kolaborasi dan *sharing* informasi secara online. Menurut O'Reilly, web 2.0 adalah revolusi bisnis di industry komputer yang disebabkan oleh

penggunaan internet sebagai platform, dan merupakan suatu percobaan untuk memahami berbagai aturan untuk mencapai keberhasilan pada platform baru tersebut.

- Web 3.0

Web 3.0 menurut John Markoff, web 3.0 adalah sekumpulan teknologi yang menawarkan cara baru yang efisien dalam membantu komputer mengorganisasi dan menarik kesimpulan secara online dalam bentuk mobile.

## 2.5 Database

Database atau basis data adalah kumpulan data yang disimpan secara sistematis didalam komputer yang dapat diolah atau dimanipulasi menggunakan perangkat lunak untuk menghasilkan informasi menurut Gordon C. Everest (2002:17). Database yang digunakan untuk pembuatan web yaitu database MySQL.

Database MySQL adalah sistem manajemen database SQL yang bersifat *Open Source*. Prasetyo (2004), MySQL adalah suatu perangkat lunak database relasi (*Relational Database Management Sistem* atau RDBMS), antara lain ORACLE, Microsoft Access, MS SQL, dan sebagainya. Sistem database MySQL mendukung beberapa fitur seperti multithreaded, multi-user, dan SQL database management sistem (DBMS). Dari beberapa jenis database yang ada maka ditentukanlah database MySQL untuk mengembangkan dokumen manual ke dokumen digital melalui media Website. Kelebihan penggunaan sistem manajemen database ini yaitu :

- Source MySQL dapat diperoleh dengan mudah dan gratis.
- Pengaksesan database dapat dilakukan dengan mudah.
- MySQL merupakan program multithreaded, sehingga dapat dipasang pada server yang memiliki multiCPU.

- Memiliki jenis kolom yang cukup banyak sehingga memudahkan konfigurasi sistem database.
- Memiliki sistem sekuriti yang cukup baik dengan verifikasi host.
- Mendukung Open Database Connectivity (ODBC) untuk sistem operasi windows.
- Mendukung record yang memiliki kolom dengan panjang tetap atau panjang bervariasi.

## 2.6 Testing dan Implementasi

Testing adalah proses yang dibuat sedemikian rupa untuk mengidentifikasi ketidaksesuaian hasil sebuah aplikasi dengan hasil yang diharapkan menurut Romeo (2003).

- Testing dilakukan untuk memastikan kualitas yaitu menguji apakah sistem aplikasi yang dihasilkan sesuai dengan Testing dilakukan untuk memastikan kualitas seperti :
  1. Verifikasi berlaku sebagaimana mestinya
  2. Mendeteksi error, dan Validasi, apakah sudah memenuhi keinginan user
  3. Verifikasi : pengecekan/pengetesan entitas-entitas, termasuk software, untuk pemenuhan dan konsistensi dengan melakukan evaluasi hasil terhadap kebutuhan yang telah ditetapkan
  4. Validasi : melihat kebenaran sistem, apakah proses yang telah ditulis dalam spesifikasi adalah apa yang sebenarnya.
  5. Deteksi error : testing untuk mendeteksi kesalahan secara insentif, yaitu menentukan apakah suatu hal tersebut terjadi bilamana tidak seharusnya terjadi .

Alhasil melalui pengetesan terhadap sistem akan dapat :

- Meningkatkan kepercayaan, tingkat resiko yang dapat diterima
- Menyediakan informasi yang dapat mencegah terulangnya error
- Menyediakan informasi untuk mendeteksi error secara dini

- Mencari error dan kelemahan/keterbatasan sistem.
- Mencari sejauh apa kemampuan dari sistem
- Menyediakan informasi untuk kualitas produk software

Tahap dari proses implementasi sistem merupakan bagian dari pengembangan sistem informasi hanya saja implementasi sistem (sistem implementation) merupakan kegiatan memperoleh dan mengintegrasikan sumber daya fisik dan konseptual yang menghasilkan suatu sistem yang bekerja. Tahap implementasi sistem (sistem implementation) adalah tahap meletakkan sistem supaya siap dioperasikan. Untuk pengetesan program dilakukan dengan cara menggunakan metode Black Box, metode ini menggunakan tidak melihat ke struktur program perangkat lunak, yang bersifat melihat kesesuaian perangkat lunak dengan kebutuhan pengguna dan spesifikasi sistem.

Black-Box yaitu test case program berdasarkan pada spesifikasi sistem, input dari data testing diharapkan bisa menemukan output yang salah, perencanaan tes dapat dimulai pada awal proses perangkat lunak. Pengetesan Sistem, dilakukan secara bertahap dengan melihat berbagai keberhasilan dan kegagalan apa saja yang dihasilkan oleh sistem. Pengetesan sistem biasanya dilakukan setelah selesai pengetesan program. Pengetesan sistem dilakukan untuk mengecek ulang dan memeriksa kekompakan antar komponen sistem yang diimplementasi agar sesuai dengan apa yang diharapkan.