

BAB 2

TINJAUAN PUSTAKA

2.1 Tinjauan Perusahaan

Tinjauan perusahaan ini merupakan informasi terkait tempat penelitian yang penulis lakukan di Pusat Penelitian Dan Pengembangan Teknologi Minyak Dan Gas Bumi (PPPTMGB) “LEMIGAS”, hal yang akan dibahas yaitu mengenai profil dan struktur organisasi Pusat Penelitian Dan Pengembangan Teknologi Minyak Dan Gas Bumi (PPPTMGB) “LEMIGAS”.

2.1.1 Profil PPPTMGB “LEMIGAS”

Pusat Penelitian dan Pengembangan Teknologi Minyak dan Gas Bumi (PPPTMGB) “LEMIGAS” didirikan pada 1 Juni 1965. Ini merupakan bagian dari Badan Litbang Kementerian Energi dan Sumber Daya Mineral (Balitbang ESDM-Kementerian ESDM). LEMIGAS adalah satu-satunya unit penelitian dan pengembangan yang fokus dalam teknologi minyak dan gas bumi di Indonesia. LEMIGAS melakukan penelitian, perekayasa, dan pengembangan teknologi di bidang hulu dan hilir migas agar sumber daya alam dapat dikelola dengan sebaik-baiknya untuk kesejahteraan rakyat.

Di usianya yang lebih dari 50 tahun, telah banyak proses dan momentum yang dilalui oleh LEMIGAS. Pada masa awa berdirinya, LEMIGAS menjadi motor untuk memberikan pelayanan berupa litbang serta Pendidikan untuk tenaga ahli kepada industry migas. Hingga akhirnya pada 1 April 2015, LEMIGAS resmi berstatus Badan Layanan Umum (BLU) secara penuh.

Dengan berstatus sebagai BLU secara penuh, LEMIGAS memberikan layanan kepada indsutri dengan fleksibilitas yang lebih baik, dengan tetap melaksanakan fungsi-fungsinya sebagai Lembaga dibawah Kementerian Energi

dan Sumber Daya Mineral (ESDM) Republik Indonesia yang melakukan litbang sesuai arahan pemerintah maupun strateginya sendiri.

Berbekal pengalaman panjang dan fasilitas yang memadai, LEMIGAS terus memperkuat layanan-layanan yang diberikan, yaitu jasa penggunaan laboratorium, jasa konsultasi, jasa survei lapangan, dan jasa litbang komersial. Untuk jasa penggunaan laboratorium, LEMIGAS didukung oleh 60 laboratorium serta 1 laboratorium kalibrasi dan pemeliharaan.

2.1.2 Visi Dan Misi PPPTMGB “LEMIGAS”

Berikut merupakan visi dan misi dari Pusat Penelitian dan Pengembangan Teknologi Minyak dan Gas Bumi (PPPTMGB) “LEMIGAS” :

1. Visi

“Menjadi Lembaga Penelitian dan Pengembangan yang unggul, Profesional, dan Bertaraf Internasional dibidang Minyak dan Gas”.

2. Misi

Mengacu pada visi Pusat Penelitian dan Pengembangan Teknologi Minyak dan Gas Bumi “LEMIGAS” diatas, maka PPTMGB “LEMIGAS” menetapkan misinya, sebagai berikut :

1. Meningkatkan peran “LEMIGAS” untuk memberikan masukan bagi kebijakan pemerintah guna meningkatkan iklim yang kondusif bagi industry migas
2. Meningkatkan Kualitas Pelayanan untuk memberikan penilaian tambahan bagi klien
3. Menciptakan produk unggulan dan mengembangkan produk andalan
4. Meningkatkan iklim kerja yang kondusif melalui sinergi, koordinasi, serta penerapan system manajemen secara konsisten

Wiranatanagara, S.T,M.T. Kelompok Geofisika dan Stratigrafi tugasnya yaitu melakukan analisis dan melihat pergerakan tanah dari hasil pengeboran minyak dan gas bumi.

2.2 Landasan Teori

Landasan teori ini berisikan teori-teori pendukung yang digunakan dalam proses analisis dan implementasi pada permasalahan yang diangkat dalam Visualisasi Data Hulumigas Dan Cekungan Sumatera Tengah Di Pusat Penelitian dan Pengembangan Teknologi Minyak dan Gas Bumi (PPPTMGB) “LEMIGAS”.

2.2.1 Visualisasi Data

Visualisasi data adalah representasi grafis dari data dan informasi. Dengan menggunakan elemen-elemen visual seperti diagram, grafik, dan peta, mengkomunikasikan data dan informasi menjadi lebih mudah dimengerti[4].

Visualisasi data bertujuan untuk agar dapat mempermudah dalam melakukan penggalian informasi dengan cepat, memahami data dalam jumlah besar. Pilihan diagram yang ditampilkan harus sesuai dengan jenis data yang terdiri dari data distribusi atau komposisi yang dapat menjelaskan kandungan dalam data. Dalam menentukan pilihan atau merubah data kedalam bentuk visual terdapat 4 kategori yang harus diperhatikan menurut *Chart Suggestion – A Thought-Starter* oleh A. Abela yaitu keterhubungan (relationship), perbandingan (comparison), distribusi atau persebaran (distribution) dan juga komposisi (composition) .

2.2.2 Tipe-Tipe Visualisasi Data

Tujuan visualisasi adalah membantu pemahaman manusia terhadap data dengan memaksimalkan sebuah sistem penglihatan manusia yang bisa membedakan *pattern*, *spot the trends*, dan indentifikasi *outliers*. Tantangan dari memvisualisasikan data adalah bagaimana membuat visualisasi yang efektif, menarik, dan tepat terhadap data yang dipakai. Terdapat tujuh hal yang harus dipenuhi dalam melakukan abstraksi tingkat tinggi (high-level abstraction),

semakin banyak hal yang disembunyikan, semakin banyak juga langkah-langkah yang harus dipenuhi, langkah tersebut yaitu :

1. Overview : melihat gambaran dari keseluruhan data,
2. Zoom : memperbesar item yang terlihat menarik
3. Filter : melakukan penyaringan terhadap item yang dirasa kurang menarik.
4. Details-on-demand : pilih satu item dari grup tertentu dan dapat melihat detail kapan saja.
5. Relate : lihat relasi dari setiap item
6. History : dapat mengulang Kembali ke aksi sebelumnya
7. Extract : dapat melakukan ekstraksi dari parameter yang diberikan

Berdasarkan taksonominya, grafik visual dibedakan menjadi : ID/Linear, 2D/Planar, Temporal, Multidimensional

2.2.2.1.1 1D / Linear

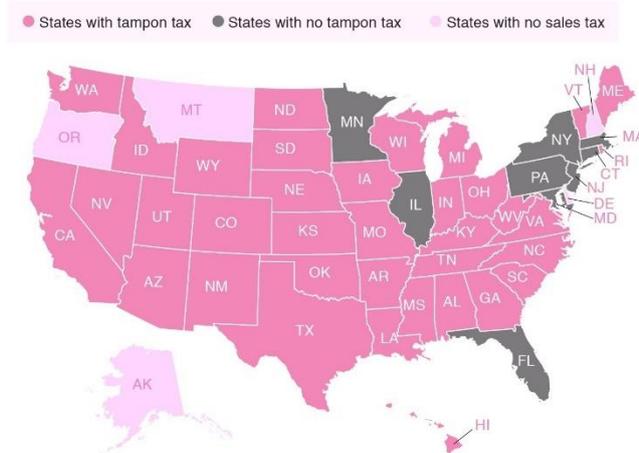
Grafik 1-dimensi termasuk di dalamnya adalah tipe data tekstual, kode sumberprogram, dan huruf alfabet. Setiap item yang digambarkan memiliki elemen garis . Contoh dari grafik 1D seperti kode-kode DNA, perbedaan kode sumber, urutan alfabet dan lain-lain. (tidak divisualisasikan secara umum).

2.2.2.1.2 2D / Planar

Grafik 2-dimensi termasuk di dalamnya peta geografis, denah rancangan, atau layout koran. Setiap item pada grafik 2-dimensi memiliki total area dan atribut (warna, ukuran, dll).

TAMPON TAX THROUGHOUT THE UNITED STATES

Only 13 states out of 50 do not tax menstrual products.



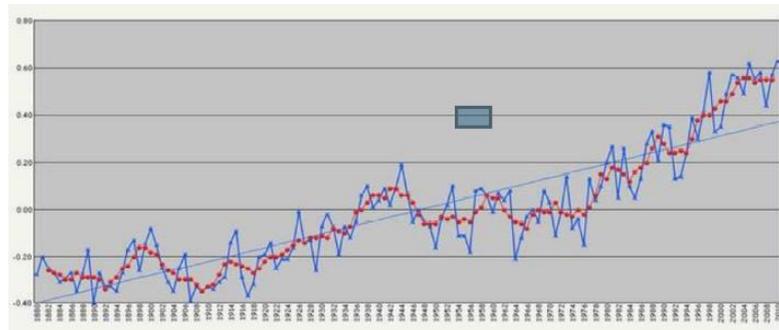
SOURCE: BBC "Why is the US 'tampon tax' so hated?"

CREDIT: Nicole Schwyn

Gambar 2.2 Contoh Grafik Planar / 2D

2.2.2.1.3 Temporal

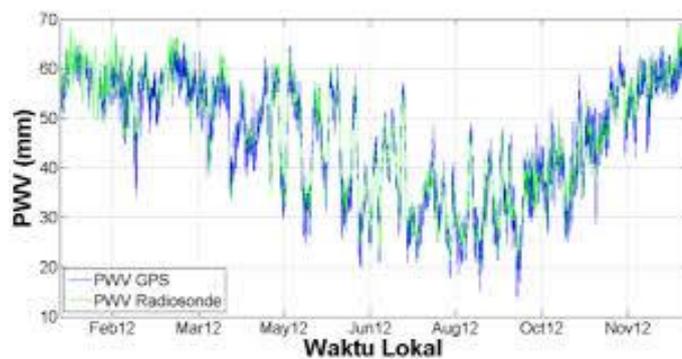
Grafik temporal adalah grafik yang berhubungan dengan waktu (time lines). Grafik ini menggambarkan persentasi historikal dari data 1-dimensi. Yang membedakan, grafik temporal memiliki item dengan waktu awal dan waktu akhir, atau periode tertentu. Contoh grafik temporal dapat dilihat pada Gambar 2.2.3.



Gambar 2.3 Contoh Grafik Temporal

2.2.2.1.4 Multidimensional

Grafik temporal didalamnya termasuk grafik-grafik yang dihasilkan dari manipulasi data dari disiplin ilmu statistika. Antarmuka representasi multidimensional adalah grafik 2-dimensi. Grafik multi-dimensi termasuk didalamnya grafik pie, histogram, tag cloud, bubble cloud, bar, tree-map, scatter plot, bubble chart, line chart, step chart, heat-map, parallel sets, spider chart, box-plot, mosaic display, waterfall, dan tabular. Contoh grafik multidimensional bisa dilihat gambar 2.4



Gambar 2.4 Contoh grafik multidimensional

2.2.3 Proses Visualisasi Data

Adapun tahap proses untuk memvisualisasi data diantaranya [5]:

- a. Acquire

Tahap acquire ini merupakan memperoleh data dari berbagai sumber, dapat dari file yang sangat banyak maupun hasil dari internet.

b. Parse

Tahap parse ini merupakan menyajikan struktur data yang kemudian digolongkan atau di kategorikan agar dapat dengan mudah diketahui jenis-jenisnya.

c. Filter

Tahap filter ini merupakan menyeleksi data yang dibutuhkan kemudian menyembunyikan data yang tidak butuh dalam pemrosesan.

d. Mine

Tahap mine ini merupakan pengaplikasian data mining maupun ilmu statistika untuk mengetahui pola dari data yang ada.

e. Represent

Tahap represent ini merupakan menampilkan sebuah data yang telah dianalisis kemudian di sampaikan kedalam bentuk model visual seperti grafik, diagram, dan lain-lain.

f. Interact

Tahap interact ini merupakan membuat data visual dapat di kontrol data apa yang ingin divisualisasikan

2.2.4 Data

Data adalah sekumpulan keterangan atau fakta yang dibuat dengan simbol, angka, kata-kata maupun kalimat [6]. Data tersebut diperoleh lewat sebuah proses pencarian serta pengamatan yang tepat berdasarkan sumber-sumber tertentu.

Dimana data berdasarkan sumber-sumber tertentu didapatkan dari hasil pengamatan yang kemudian diolah menjadi bentuk lainnya yang lebih kompleks. Baik itu berupa Database, informasi, dan lain sebagainya.

Jika ditinjau secara Bahasa latin yakni 'Datum' yang memiliki arti suatu yang diberikan. Dari istilah tersebut maka dijumpai arti data yang merupakan hasil dari pengamatan atau pengukuran sebuah variabel tertentu dalam bentuk warna, kata-kata, simbol, angka, atau keterangan lainnya.

Sesuai dengan macam atau jenis variabel, maka data juga mempunyai jenis sebanyak variabel. Data dapat dibagi dalam kelompok tertentu sesuai dengan kriteria yang menyertainya, seperti menurut susunan, sifat, waktu pengumpulan, dan sumber pengambilan.

2.2.5 Basis Data

Basis data menurut Stepens dan Plew adalah mekanisme yang digunakan untuk menyimpan informasi atau data [6]. Dengan basis data, pengguna dapat menyimpan data secara terorganisasi. Setelah data tersimpan, informasi harus mudah diambil atau ditampilkan. Cara data disimpan menentukan seberapa mudah mencari suatu informasi berdasarkan kriteria yang ada. Data pun harus mudah ditambahkan ke dalam basis data, dimodifikasi, dan dihapus.

2.2.6 NoSQL

NoSQL adalah istilah yang dikenal dalam teknologi komputasi untuk merujuk kepada kelas yang luar dalam system manajemen basis data yang diidentifikasi dengan tidak mematuhi aturan pada model system manajemen basis data relasional yang banyak digunakan [6]. NoSQL tidak dibangun terutama dengan table dan umumnya tidak menggunakan SQL untuk memanipulasi data, sehingga sering ditafsirkan sebagai 'Tidak hanya SQL'. Keunggulan dari NoSQL dibandingkan dengan basis data relasional ada pada skalabilitas. Selain skalabilitas, NoSQL memiliki keunggulan yaitu sebagai berikut.

1. *Schemaless data representation* : hampir semua model NoSQL merepresentasikan *scemaless*. Administrator basis data tidak perlu memikirkan bagaimana struktur dan model basis data yang berubah setiap waktu.
2. *Development Time: Query* yang kompleks pada basis data relasional yang sudah besar, akan membuat proses penyajian data menjadi lambat (misalnya JOIN table pada basis data tersebar dibanyak server). Pada noSQL hal demikian tidak terjadi.
3. *Plan Ahaid for scalability* : aplikasi yang memakai model NoSQL akan sangat elastis. Dapat menangani masalah lonjakan sumber daya.

2.2.7 JSON

JSON (JavaScript Object Notation) adalah format pertukaran data yang ringan. Sangat mudah untuk dibaca manusia dan diuraikan oleh computer. JSON merupakan format teks yang independent tetapi menggunakan konvensi yang akrab bagi programmer. Dengan alasan itu membuat JSON menjadi Bahasa pertukaran data yang ideal. JSON dibangun diatas dua struktur, yaitu : (Json.org, [no date])

Koleksi pasangan nama atau nilai. Dalam berbagai Bahasa pemrograman, ini direalisasikan sebagai objek, rekaman, struktur, kamus, table hash, daftar terkunci, atau associaitive array.

1. Daftar dari nilai. Dalam kebanyakan Bahasa pemrograman, ini direalisasikan sebagai array, vector, daftar, atau urutan.

Struktur-struktur data ini disebut sebagai struktur data universal. Pada dasarnya, semua Bahasa pemrograman modern mendukung struktur data ini dalam bentuk yang sama maupun berlainan. Hal ini pantas disebut demikian karena format data mudah dipertukarkan dengan Bahasa-bahasa pemrograman yang juga berdasarkan pada struktur data ini.

2.2.8 RESTful Web Service

- Web Service

Web service adalah sistem perangkat lunak yang dirancang untuk mendukung interaksi yang bisa beroperasi *machine-to-machine* diatas jaringan. Web service mempunyai alat pendukung yang diuraikan didalam format *machine-processable* (secara spesifik *Web Services Description Language*). Sistem lain yang saling berhubungan dengan Web Service didalam cara yang ditentukan oleh deskripsinya yang menggunakan pesan SOAP (*Simple Object Access Protocol*), secara khas disampaikan menggunakan HTTP (*Hypertext Transfer Protocol*) dengan XML (*Extensible Markup Language*) serialization, Bersama dengan standar lain yang terkait dengan web.

Model dari sebuah web service didasarkan pada antara 3 komputer yang berperan dalam web service, yaitu : service provider, service registry, dan service requestor atau consumer.

Interaksi yang terjadi antara ketiga komponen tersebut juga melibatkan operasi *publish*, *find* dan *bind*. *Service provider* menyediakan *service* yang dapat diakses melalui jaringan komputer, misalnya *internet*. Kemudian, *service provider* mendeskripsikan *service* yang dibangun dan mem-*publish*-kan *service description* tersebut ke *service registry* atau secara langsung ke *service consumer*. *Service requestor* atau *consumer* menggunakan operasi *find* untuk mendapatkan *service description* secara lokal maupun melalui *service registry*. *Service description* yang diperoleh itu kemudian digunakan untuk mem-*bind service provider* dan berinteraksi dengan implementasi *web-service* yang akan digunakan tersebut.

- REST

REST (Representational State Transfer) adalah salah satu jenis web service yang menerapkan konsep perpindahan antar state. State disini dapat digambarkan seperti jika browser meminta suatu halaman web, maka server akan mengirimkan state halaman web yang sekarang ke browser. Bernavigasi melalui link-link yang disediakan sama halnya dengan mengganti state dari halaman web. Begitu pula REST bekerja, dengan bernavigasi melalui link-link HTTP untuk melakukan aktivitas tertentu, seakan-akan terjadi perpindahan state satu sama lain.

Perintah HTTP yang bisa digunakan adalah fungsi GET, POST, PUT, atau DELETE. Balasan yang dikirimkan dalam bentuk XML sederhana tanpa ada protocol pemaketan data, sehingga informasi yang diterima lebih mudah dibaca dan diparsing disisi client.

Dalam pengaplikasiannya, REST lebih banyak digunakan untuk *web service* yang berorientasi pada *resource*. Maksud orientasi pada *resource* adalah orientasi yang menyediakan *resource-resource* sebagai layanannya dan bukan kumpulan-kumpulan dari aktivitas yang mengolah *resource* itu. Alasan mengapa REST tidak digunakan dalam skripsi ini karena orientasi pada *resourcenya* itu, sedangkan aplikasi lowongan pekerjaan membutuhkan pemanggilan metode yang bisa dikerjakan terhadap kumpulan *resource* data lowongan. Selain itu, karena standarnya yang kurang sehingga tidak begitu cocok diterapkan dalam aplikasi yang membutuhkan kerjasama antar aplikasi lain, dimana standar yang baik akan sangat berguna karena berbicara dalam satu bahasa yang sama. Beberapa contoh *web service* yang menggunakan REST adalah: Flickr API(*Application Program Interface*), YouTube API, Amazon API.

2.2.9 Analisis dan Desain Berorientasi Objek

Analisis dan desain berorientasi objek (Objek Oriented Analysis and Design) adalah cara baru memikirkan suatu masalah dengan menggunakan model yang dibuat menurut konsep. Dasar pembuatannya sendiri adalah objek yang merupakan kombinasi antara struktur data dan perilaku dalam satu entitas.

Alasan mengapa harus memakai metode berorientasi objek yaitu karena perangkat lunak itu sendiri yang bersifat dinamis, dimana hal ini disebabkan karena kebutuhan pengguna berubah dengan cepat. Selain itu bertujuan untuk menghilangkan kompleksitas transisi antar tahap pada pengembangan perangkat lunak, karena pada pendekatan berorientasi objek, notasi yang digunakan pada tahap analisis perancangan dan implementasi relative sama tidak seperti pendekatan konvensional yang dikarenakan notasi yang digunakan pada tahap analisisnya berbeda-beda hal ini menyebabkan transisi antar tahap pengembangan menjadi kompleks.

Di samping itu dengan pendekatan berorientasi objek membawa pengguna kepada abstraksi atau istilah yang lebih dekat dengan dunia nyata, karena di dunia nyata itu sendiri yang sering pengguna lihat adalah objeknya bukan fungsinya. Beda ceritanya dengan pendekatan terstruktur yang hanya mendukung abstraksi pada level fungsional. Adapun dalam pemrograman berorientasi objek menekankan berbagai konsep seperti: Class, Object, Abstract, Encapsulation, Polymorphism, Inheritance dan tentunya UML (Unified Modeling Language).

UML (Unified Modeling Language) sendiri merupakan salah satu alat bantu yang dapat digunakan dalam bahasa pemrograman berorientasi objek. Selain itu UML merupakan standard modeling language yang terdiri dari kumpulan-kumpulan diagram, dikembangkan untuk membantu para pengembang sistem (developer) dan software agar bisa menyelesaikan tugas-tugas seperti: Spesifikasi, Visualisasi, Desain Arsitektur, Konstruksi, Simulasi dan Testing. Dapat disimpulkan bahwa UML (Unified Modeling Language) adalah sebuah bahasa yang

berdasarkan grafik atau gambar untuk memvisualisasikan, menspesifikasikan, membangun dan pendokumentasian dari sebuah sistem pengembangan perangkat lunak berbasis Objek (Object Oriented Programming) [7].

Dokumentasi UML menyediakan ada 10 macam diagram untuk memodelkan aplikasi berorientasi objek yang 4 diantaranya adalah [8]:

1. Use Case Diagram

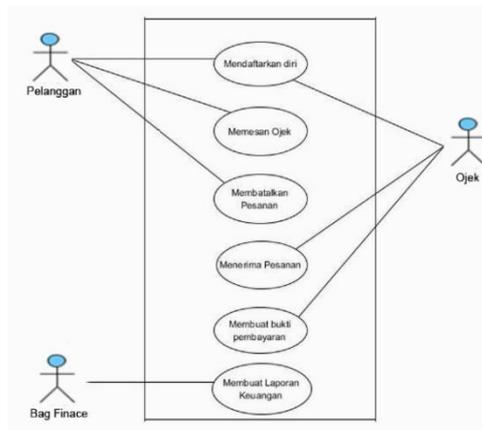
Use Case Diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Use case diagram ini lebih menekankan kepada apa yang diperbuat system dan bagaimana sebuah sistem itu bekerja. Sebuah use case merepresentasikan sebuah interaksi antara actor dengan sistem. Use case merupakan bentuk dari sebuah pekerjaan tertentu, misalnya login kedalam sistem, posting, dan sebagainya, sedangkan seorang actor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan system untuk melakukan pekerjaan-pekerjaan tertentu [9]. Adapun komponen-komponen dalam use diagram diantaranya:

- a. Aktor

Actor merupakan suatu entitas yang berkaitan dengan system tapi bukan dari bagian dalam itu sendiri. Actor berada diluar system namun berkaitan erat dengan fungsionalitas didalamnya. Actor dapat memiliki hubungan secara langsung terhadap fungsi utama baik terhadap salah satu atau semua fungsionalitas semua. Actor juga dapat dibagi berbagai jenis atau tingkatan dengan cara digeneralisasi atau dispesifikasi tergantung kebutuhan sistemnya. Actor biasanya dapat berupa pengguna atau database yang secara pandang berada dalam suatu ruang lingkup sistem tersebut.

- b. Use Case

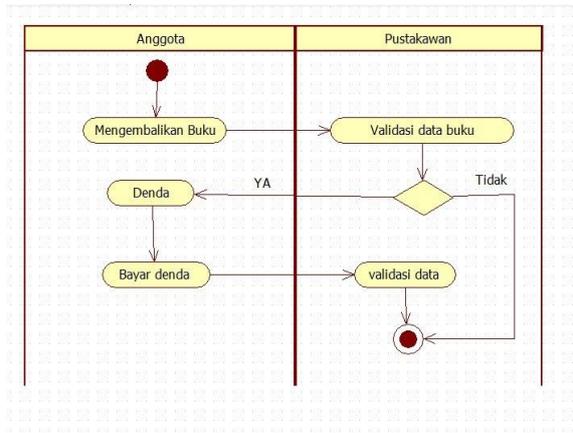
Use case merupakan gambaran umum dari fungsi atau proses utama yang menggambarkan tentang salah satu perilaku sistem. Perilaku sistem ini terdefinisi dari proses bisnis sistem yang akan dimodelkan. Tidak semua proses bisnis digambarkan secara fungsional pada use case, tetapi yang digambarkan hanya fungsionalitas utama yang berkaitan dengan sistem. Use case menitik beratkan bagaimana suatu sistem dapat berinteraksi baik antar sistem maupun diluar sistem. Contoh use case diagram dapat dilihat pada gambar berikut:



Gambar 2.5 Contoh Use Case Diagram

2. Activity Diagram

Activity Diagram adalah sebuah tahapan yang lebih fokus kepada menggambarkan proses bisnis dan urutan aktivitas dalam sebuah proses. Di mana biasanya dipakai pada business modeling untuk memperlihatkan urutan aktifitas proses bisnis. Activity Diagram ini sendiri memiliki struktur diagram yang mirip flowchart atau data flow diagram pada perancangan terstruktur. Activity Diagram dibuat berdasarkan sebuah atau beberapa use case pada use case diagram. Contoh activity diagram dapat dilihat pada gambar berikut.

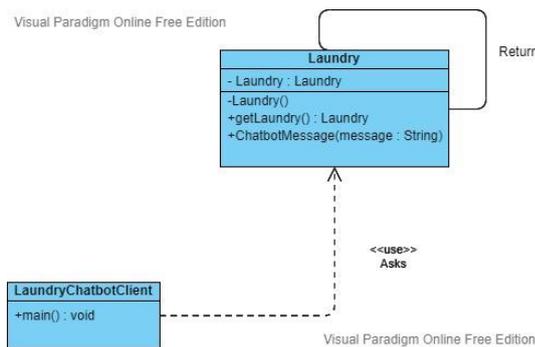


Gambar 2.6 Contoh Activity Diagram

3. Class Diagram

Class Diagram adalah sebuah class yang menggambarkan struktur dan penjelasan class, paket, dan objek serta hubungan satu sama lain.

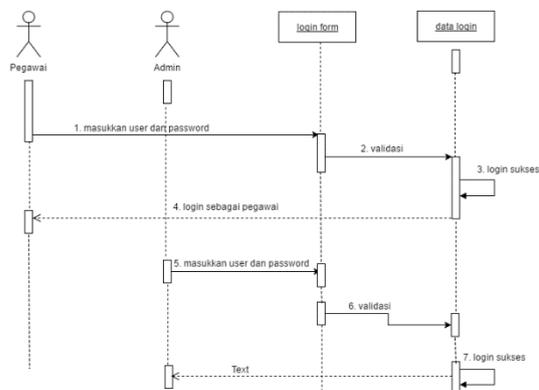
Class diagram juga menjelaskan hubungan antar class secara keseluruhan di dalam sebuah sistem yang sedang dibuat dan bagaimana caranya agar mereka saling berkolaborasi untuk mencapai sebuah tujuan [9]. Contoh dari class diagram dapat dilihat pada gambar berikut



Gambar 2.7 Contoh Class Diagram

1. Sequence Diagram

Sequence Diagram digunakan untuk menggambarkan perilaku pada sebuah skenario. Diagram jenis ini memberikan kejelasan sejumlah objek dan pesan-pesan yang diletakkan diantaranya di dalam sebuah use case. Komponen utamanya adalah objek yang digambarkan dengan kotak segi empat atau bulat, message yang digambarkan dengan garis penuh, dan waktu yang ditunjukkan dengan progress vertical. Manfaat dari sequence diagram adalah memberikan gambaran detail dari setiap use case diagram yang dibuat sebelumnya [9]. Bisa dilihat dari contoh sequence diagram berikut.



Gambar 2.8 Contoh Sequence Diagram

2.2.10 Usability Testing

Bauer (2010) memberikan definisi tentang *usability testing* (uji kegunaan) adalah mengukur efisiensi, kemudahan dalam mempelajari, dan kemampuan untuk mengingat bagaimana berinteraksi tanpa mengalami kesulitan [10]. Sejak dimulai berkembangnya *internet* para pakar dalam bidang uji kegunaan menekankan uji kegunaan dengan dua hal penting, yaitu :

- a. Ease of learning

Mengukur kegunaan dengan membandingkan waktu yang digunakan dalam mempelajari system computer. Yang belum pernah dikenalnya sama sekali, dengan waktu yang diperlukan untuk melakukan hal yang sama dengan cara lain.

b. Ease of use

Mengukur jumlah Tindakan yang digunakan untuk menyelesaikan suatu pekerjaan. Misalnya membandingkan antara halaman pertama dengan kedua pada dua desain.

Dari pendapat diatas maka dapat disimpulkan usability sangat berguna untuk keberlangsungan dari website. Jika sebuah website sulit digunakan oleh pengguna, maka pengguna akan pergi dan tidak akan Kembali untuk mengunjungi website tersebut. Jika sebuah *homepages* susah dalam memberikan penjelasan kepada pengguna maka pengguna juga tidak akan mengunjungi Kembali *homepages* tersebut. Jika sebuah website harus dirancang sebaik mungkin, sehingga memudahkan pengguna untuk menggunakan *website* dan mengunjungi Kembali situs website tersebut. Jadi sebuah website harus dirancang sebaik mungkin, sehingga memudahkan pengguna untuk menggunakan *website* dan mengunjungi Kembali situs website tersebut.

Uji kegunaan website merupakan kombinasi dari lima aspek yaitu :

- a. *Ease of learning* (mudah dipelajari)
- b. *Efficiency of use* (efisien dalam penggunaan).
- c. *Memorability* (mudah diingat).
- d. *Error frequency and severity* (frekuensi kesalahan dan kesederhanaan).
- e. *Subjective satisfaction* (kepuasaan pengguna).

2.2.10.1.1 Langkah-langkah Usability Testing

Berikut langkah – langkah yang harus dilakukan untuk menerapkan metode usability testing :

1. Komponen Usability Testing

Komponen *usability testing* menurut Sastramihardja(2006) terdiri dari:

- a. Learning , tingkat keberhasilan dalam menyelesaikan tugas untuk setiap jenis partisipan dan rasio halaman dengan hasil rata-rata kunjungan.
- b. Efficiency, kelompok penggunaan dalam mengerjakan tugas yang bermacam- macam.

2. Pemilihan Responden Usability Testing

Pemilihan responden menurut Krug (2006) jumlah pengguna yang ideal untuk setiap pengujian tiga atau empat paling banyak. Pemilihan responden yang akan memberikan isian kuisisioner dengan jumlah responden tiga orang diantaranya : satu orang pengguna aktif internet, satu orang pengguna terampil dalam menggunakan internet, dan satu orang pengguna awam (Rusidi, 2011).

3. Pengukuran Usability

Pengukuran *usability* digunakan untuk menilai apakah interaksi diantara pengguna dengan sebuah perangkat lunak berjalan dengan baik. Pengukuran dilakukan dengan mengikuti konsep *user testing*, dengan menekankan kepada pengukuran bukan kepada pengujian, sebagai berikut:

- a. Menentukan tujuan dan mengeksplorasi pertanyaan.
- b. Memilih paradigm dan Teknik pengukuran.
- c. Merancang tugas yang akan dijadikan sasaran dalam melakukan pengukuran.
- d. Memilih partisipan dari pengguna untuk mencoba website.
- e. Mempersiapkan kondisi pengukuran
- f. Merencanakan jalannya pengukuran.
- g. Melakukan evaluasi, analisis dan penyajian data.

4. Tujuan pengukuran usability

Pengukuran dilakukan untuk mengidentifikasi masalah dari *usability testing* yang dapat mempengaruhi antara interaksi sistem dengan pengguna dari hasil perancangan perangkat lunak. Pengukuran dengan melakukan uji coba perangkat lunak kepada sejumlah partisipan (bertindak sebagai pengguna) dengan melakukan pengamatan. Kemudian partisipan mengisi kuisioner agar dapat memberikan hasil dari gambaran tingkat kepuasan pengguna dalam menggunakan website. Kemudian masukan dari partisipan dapat digunakan sebagai umpan balik untuk melengkapi prasyarat fungsional atau kebutuhan terhadap pengguna.

5. Teknik pengukuran Usability

Paradigma untuk melakukan pengukuran adalah *usability testing* dengan fokus kepada pengukuran performa dari pengguna melalui sejumlah tugas yang telah dipersiapkan sebelumnya. Dalam Teknik pengukuran ini dilakukan oleh pengguna dengan meminta partisipan untuk menjalankan tugasnya.