

BAB 2

TINJAUAN PUSTAKA

1.1 Tinjauan Pustaka

Pada penelitian ini digunakan beberapa referensi yang berhubungan dengan materi penelitian. Referensi diambil dari contoh-contoh hasil penelitian sebelumnya. Hasil penelitian yang dipakai antara lain :

1. Pada penelitian yang dilakukan oleh Bouchiha Djelloul, 2009 yang berjudul "Towards Reengineering Web Application to Web Services" yang membahas mengenai reengineering web dengan menerapkan arsitektur yang berorientasi pada sistem layanan yang ada dengan menggunakan kerangka kerja yang disebut WA2WS. Kerangka kerja tersebut terdiri dari dua tahap ,tahap pertama adalah fase abstraksi yang terdiri dari penggalan UML sebagai sekema konseptual dari aplikasi web dan fase kedua adalah fase implementasi yang terdiri dari menganalisis kode layanan pada web ke konseptual UML. Pada penelitian ini juga membahas tentang teknik reverse engineering untuk pemulihan dari web dengan bentuk modul use case.

Tabel 0.1 Penelitian Terkait

No	Nama Peneliti dan Tahun	Masalah	Metode	Hasil
1.	Bouchiha Djelloul, Malik Mimoun, dan Mostefai Abd El Kader, 2009	Reengineering web dengan Arcitektur yang berorientasi pada sistem layanan.	Kerangka kerja WA2WS dan teknik reverse engineering.	Reengineering website dengan membuat representasi web baru untuk perbaikan website menggunakan kerangka WA2WS

1.2 Reengineering

Reengineering adalah proses menganalisis subjek sistem untuk mengidentifikasi komponen sistem yang berkaitan dan menciptakan representasi dari sistem dalam meningkatkan atau membuat bentuk baru [3]. Kelengkapan proses reengineering mengacu pada detail yang diberikan suatu abstraksi. Kelengkapan meningkat berbanding lurus dengan jumlah analisis yang dilakukan. Tingkat abstraksi suatu proses reengineering harus melewati hal-hal berikut ini :

1. Represetasi Procedural

Menggambarkan Prosedural website yang sedang berjalan.

2. Program dan informasi struktur data

Digunakan untuk mengumpulkan suatu informasi kebutuhan pada sistem melalui konsultasi dengan pengguna sistem.

3. Data dan model system control

Pengumpulan data yang telah di analisis dan pemodelan sistem.

4. Model hubungan entitas

Pendesainan ulang website yang sudah dianalisis dan diidentifikasi dengan cara mendesain ulang suatu proses dengan mempelajari peta proses, mempermudah untuk melihat bagian-bagian tertentu yang bisa diotomatisasikan, disederhanakan dan lebih ditekankan (diutamakan) dengan menggunakan UML.

Proses dari reengineering sebuah website pertama dimulai dengan memahami dan mengekstrak abstraksi procedural yang direpresentasikan oleh kode sumber. Fungsi keseluruhan dari sistem harus dipahami sebelum melakukan ketahap berikutnya. Pada tingkat sistem, struktural data global sering direkayasa untuk mengakomodasi paradigma manajemen database baru seperti sistem database relasional atau OOP.

Pada tingkat struktural sistem dibagi menjadi dua yaitu:

1. Struktur data Internal

Merupakan pendekatan untuk rekayasa balik atau reengineering website.

2. Struktur database

Panduan untuk mendefinisikan model data yang ada ke dalam model database yang baru adalah sebagai berikut :

- a. Membangun model objek
- b. Menentukan Kunci calon
- c. Definisikan generaliasi

Pada dasarnya inti dari reengineering website adalah teknik reverse engineering data yang digunakan untuk membuat representasi di tingkat yang lebih tinggi dari abstraksi.

1.3 Kerangka Kerja WA2WS

Pada kerangka kerja WA2WS menggunakan teknik reverse engineering dalam memulihkan logika aplikasi web yang ada dalam bentuk modul use case [3]. Berdasarkan studi kasus penggunaan kerangka kerja WA2WS, pengguna dapat membangun layanan web yang diinginkan untuk memperbaiki layanan web. Dengan kerangka kerja ini juga dapat digunakan sebagai mediator untuk berkomunikasi antara pemohon layanan web dengan web yang diinginkan sehingga web sesuai dengan keinginan stakeholder. Output yang dihasilkan dari wrapper layanan dapat di terjemahkan ke dalam format XML. Oleh karena itu saran yang dihasilkan dari metode WA2WS dapat mendukung komunikasi program to program yang dihasilkan untuk layanan web. WA2WS terdiri dari empat komponen utama:

1. WA atau Komponen Ekstraktor

Komponen ekstraktor digunakan untuk mengekstraksi komponen dari aplikasi web URLnya.

2. Use Case Diagram Abstractor

Komponen untuk memulihkan dan memvisualisasikan model kasus penggunaan dari aplikasi website.

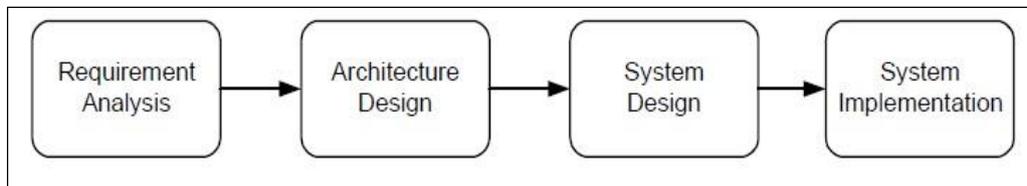
3. Layanan Wrapper Generator

Wrapper Generator adalah komponen untuk menghasilkan pembungkus layanan.

4. WSDL Generator adalah komponen untuk menciptakan Deskripsi Web Service Bahasa (WSDL) menurut skema XML standar dokumen WSDL

1.4 Reverse Engineering

Reverse engineering atau rekayasa balik, reverse engineering sendiri dapat di definisikan sebagai proses menganalisa suatu sistem melalui identifikasi komponen-komponennya dan keterkaitan antar komponen, serta mengekstraksi dan membuat abstraksi dan informasi perancangan dari sistem yang dianalisa tersebut. Pada web, teknik reverse engineering sendiri adalah seni membedah suatu aplikasi dengan tidak tersedianya source code [3]. Untuk lebih memahami tentang reverse engineering, pertama kita harus memahami tentang model waterfall dalam perkembangan perangkat lunak. Dalam metode waterfall terdiri dari beberapa tahapan yang pertama adalah proses analisis kebutuhan, arsitektur desain, desain sistem, dan implementasi sistem.



Gambar 2.4 Model Waterfall

Selama proses pengembangan sistem terutama pada tahapan desain, pemberian informasi desain tidak sepenuhnya di dukung oleh bahasa pemrograman. Tujuan utama dari pemodelan waterfall untuk rekayasa balik atau reverse engineering adalah untuk memelihara dan memperbaiki sistem serta memulihkan informasi yang hilang. Dalam proses pengembangan sistem ini, peran penting reverse engineering dapat diringkas sebagai berikut :

1. Pemahaman sistem atau System understanding

Informasi desain yang diekstraksi yang membentang dari arsitektur system digunakan untuk spesifikasi, dan untuk desain logika.

2. Dasar untuk pemeliharaan dan pembangunan kembali

Pada tahapan desain, desain tersebut diekstrak, dispesifikasikan, dan pada tahap arsitektur dapat digunakan sebagai titik awal untuk perbaikan sistem, termasuk desain ulang, re-spesifikasi, re-arsitektur dan pembangunan kembali.

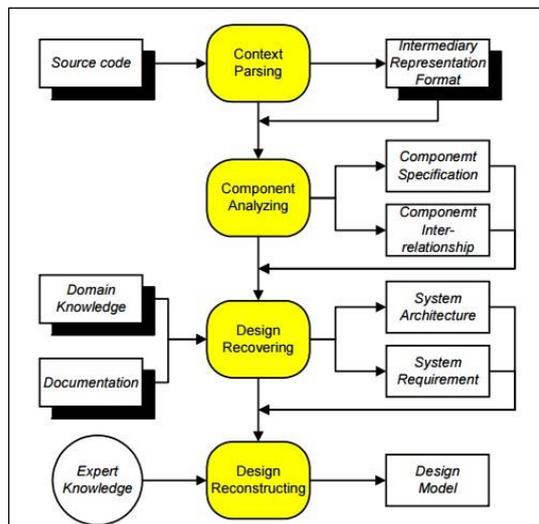
3. Penggunaan kembali komponen atau Reuse

Reuse dapat terjadi pada berbagai bentuk abstraksi, termasuk pola desain, arsitektur sistem, spesifikasi dan desain logis atau fisik.

Oleh karena itu reverse engineering adalah cara yang efektif dari kedua identifikasi dan pemulihan komponen reuse. Inti reverse engineering adalah untuk memahami kode. Berbeda dengan proses pengembangan dan penyempurnaan, reverse engineering mencoba untuk menelusuri rute dan menemukan data asli yang hilang selama pelaksanaan. Namun, karena karakter bawaan dan keterbatasan bahasa pemrograman tertentu, sebaliknya teknik dan pemahaman Program pendekatan sering dibatasi oleh satu pertimbangan.

1.4.1 Proses Reverse Engineering

Reverse engineering memungkinkan kita untuk mengidentifikasi komponen sistem dan membuat representasi tingkat yang lebih tinggi dari sistem. Kita bisa melihat ini lebih baik dengan menggambarkan proses yang digunakan dalam reverse engineering. Berikut ini adalah proses dari reverse engineering.



Gambar 2.4.1 Proses Reverse Engineering

1. Context parsing

Context parsing adalah tahap pertama untuk menganalisis source code dan analisis sintaks. Pada fase ini umumnya mengurangi penggunaan source code dan mengubahnya ke bentuk yang lebih terstruktur seperti ,Abstract Syntax Tree (AST), UML. Untuk mempermudah representasi source code.

2. Component analyzing phase

Pada fase ini akan menampilkan komponen pada website yang akan di analisis antarlain yaitu struktur grafik, atribut dan fungsi variabel. Inti dari fase ini adalah menganalisis

semua komponen yang terdapat pada website untuk mengetahui diagram aliran data supaya mempermudah dalam pembuatan UML.

3. Design Recovering Phase

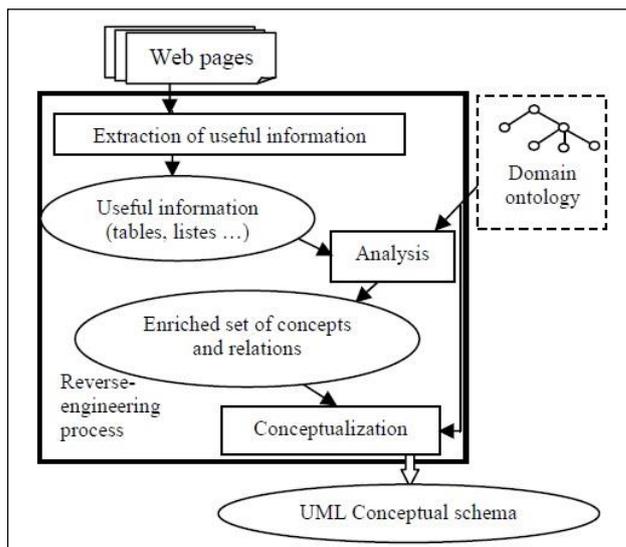
Design Recovering Phase adalah desain dari pada source code web yang telah dianalisis dengan menggabungkan representasi struktural. Kegunaan dari desain recovering adalah untuk menggambarkan keadaan nyata pada web dengan menggunakan UML atau dapat digunakan untuk mengembangkan software.

4. Design Reconstructing Phase

Pada fase ini digunakan untuk menyelesaikan fase-fase sebelumnya, pemodelan sistem dan spesifikasi diperoleh dari fase terakhir untuk merekonstruksi pandangan yang tepat dari model desain. Fase ini meliputi rekayasa keempat fase dan rekonstruksi desain model.

1.4.2 Ontologi Reverse Engineering

Merupakan pendekatan baru untuk teknik reverse engineering yang diterapkan pada website. Pendekatan ini bertujuan untuk menghasilkan skema konseptual berupa pemodelan UML pada aplikasi web [3]. Inti utama dari pendekatan ontologi reverse engineering adalah pada penggunaan ontologi yang diabstraksi pada proses reverse engineering.



Gambar 2.4.2 Proses Ontologi pada Reverse Engineering

Pada gambar 2.4.2 proses ontologi pada reverse engineering terdiri dari tiga tahap yaitu :

1. Ekstraksi Informasi

Digunakan untuk ekstraksi pada halaman HTML.

2. Analisis Informasi

Informasi yang di dapat kemudian di ekstraksi menggunakan domain ontologi.

3. Fase Konseptual UML

Menggunakan UML untuk menggambarkan konsep dari website.

1.5 Web Services

Web service adalah suatu sistem perangkat lunak yang dirancang untuk mendukung interoperabilitas dan interaksi antar sistem pada suatu jaringan. Web service digunakan sebagai suatu fasilitas yang disediakan oleh suatu web site untuk menyediakan layanan (dalam bentuk informasi) kepada sistem lain, sehingga sistem lain dapat berinteraksi dengan sistem tersebut melalui layanan-layanan (service) yang disediakan oleh suatu sistem yang menyediakan web service. Web service menyimpan data informasi dalam format XML, sehingga data ini dapat diakses oleh sistem lain walaupun berbeda platform, sistem operasi, maupun bahasa compiler [7].

Web service bertujuan untuk meningkatkan kolaborasi antar pemrogram dan perusahaan, yang memungkinkan sebuah fungsi di dalam Web Service dapat dipinjam oleh aplikasi lain tanpa perlu mengetahui detail pemrograman yang terdapat di dalamnya.

Beberapa alasan mengapa digunakannya web service adalah sebagai berikut:

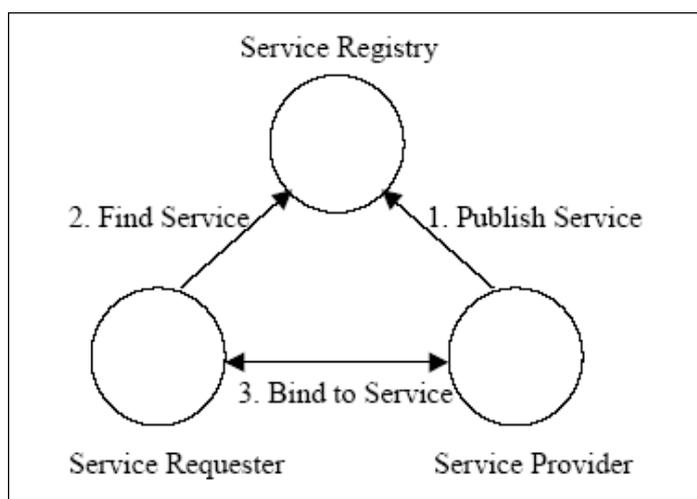
1. Web service dapat digunakan untuk mentransformasikan satu atau beberapa bisnis logic atau class dan objek yang terpisah dalam satu ruang lingkup yang menjadi satu, sehingga tingkat keamanan dapat ditangani dengan baik.
2. Web service memiliki kemudahan dalam proses deployment-nya, karena tidak memerlukan registrasi khusus ke dalam suatu sistem operasi. Web service cukup di-upload ke web server dan siap diakses oleh pihak-pihak yang telah diberikan otorisasi.
3. Web service berjalan di port 80 yang merupakan protokol standar HTTP, dengan demikian web service tidak memerlukan konfigurasi khusus di sisi firewall.

Web services atau Service-oriented Architecture (SOA) telah menjadi sebuah pilihan untuk pengembangan aplikasi Web. SOA menggunakan layanan web sebagai konstruksi dasar untuk mendukung kecepatan dan kemudahan dalam komposisi aplikasi. Dalam SOA, layanan yang didefinisikan oleh Web antarmuka yang mendukung operasi interoperable antara aplikasi perangkat lunak yang berbeda menggunakan protokol standar. Pada awal tahun sembilan puluhan, SOA menawarkan janji ketahanan dan kelincahan pada perusahaan untuk melakukan bisnis yang efisien dengan melakukan reengineering perangkat lunak, aplikasi to-aplikasi interoperabilitas, fleksibilitas desain, dan perbaikan arsitektur. Web services adalah implementasi yang paling populer yang di terapkan SOA. Sebuah web service adalah entitas software yang mampu mencapai tujuan pengguna. Jaringan layanan memungkinkan aplikasi yang ditulis dalam pemrograman yang berbeda bahasa untuk berinteraksi mulus melalui protokol standar. Web service memiliki lingkup yang luas dari aplikasi mulai dari aplikasi seperti internet banking, ramalan cuaca dan peta.

Secara umum, web service memiliki tiga operasi yang terlibat di dalamnya, yaitu:

1. Publish atau Unpublish: Menerbitkan/menghapus layanan ke dalam atau dari registry.
2. Find: Service requestor mencari dan menemukan layanan yang dibutuhkan.
3. Bind: Service requestor setelah menemukan layanan yang dicarinya, kemudian melakukan binding ke service provider untuk melakukan interaksi dan mengakses layanan/service yang disediakan oleh service provider.

1.5.1 Arsitektur Web Service



Gambar 2.5.1 Arsitektur Web Service

Web service memiliki tiga entitas dalam arsitekturnya, yaitu:

1. Service Requester (peminta layanan)

Peminta layanan yang mencari dan menemukan layanan yang dibutuhkan serta menggunakan layanan tersebut.

2. Service Provider (penyedia layanan)

Berfungsi sebagai lokasi central yang mendeskripsikan semua layanan/service yang telah di-register.

3. Service Registry (daftar layanan)

Berfungsi untuk menyediakan layanan service dan mengolah sebuah registry agar layanan-layanan tersebut dapat tersedia.

Layanan web memungkinkan perangkat lunak sebagai layanan untuk memberikan software layanan melalui jaringan menggunakan teknologi seperti XML. Layanan web yang sesuai dengan arsitektur SOA dan menggunakan protokol SOAP untuk berkomunikasi antara klien antarmuka dan penyedia disebut berbasis SOAP layanan Web. Pada tahun 2000, Fielding mengusulkan gaya arsitektur baru untuk aplikasi berbasis jaringan yang disebut Representasional Transfer (REST). SISA bertujuan untuk generalisasi interface, skalabilitas dari interaksi, dan penyebaran independen komponen perangkat lunak. Layanan web dibangun di atas SISA prinsip yang disebut web service. Dua berikutnya subbagian menjelaskan dua pendekatan ini arsitektur dengan perbandingan yang komprehensif antara keduanya.

1.5.2 Komponen-Komponen Web Service

Service Publication and Discovery (UDDI)
Service Description (WSDL)
XML Based Messaging (SOAP)
Common Internet Protocols (HTTP, TCP/IP)

Web service secara keseluruhan memiliki empat layer komponen seperti pada gambar di atas, yaitu:

1. Layer 1: Protokol internet standar seperti HTTP, TCP/IP
2. Layer 2: Simple Object Access Protocol (SOAP), merupakan protokol akses objek berbasis XML yang digunakan untuk proses pertukaran data/informasi antar layanan.
3. Layer 3: Web Service Definition Language (WSDL), merupakan suatu standar bahasa dalam format XML yang berfungsi untuk mendeskripsikan seluruh layanan yang tersedia.

1.5.3 Web Service Desain Berbasis SOAP

SOAP adalah layanan Web yang dirancang untuk memungkinkan interaksi RPC-seperti sistem remote yang didalam terdapat gaya desain antara penyedia layanan dan konsumen yang keduanya perlu membangun pemahaman sama supaya menghasilkan layanan web yang bagus. Setiap Web yang berbasis SOAP, layanan ini memiliki antarmuka yang unik dijelaskan dengan Web Services Description Language (WSDL) [7].

Dalam Universal publik Deskripsi Discovery dan Integrasi (UUDI). UDDI mempunyai lingkup mengelola dan memelihara entri layanan Web yang dapat dibuat untuk referensi WSDL dokumen. XML digunakan untuk membangun blok dasar dari layanan Web melalui beberapa bentuk XML seperti SOAP (Simple Object Access Protocol) dan XML-RPC (XML Remote Procedure Call).

Layanan SOAP telah banyak diadopsi oleh industri dan didukung oleh hampir semua alat pengembangan, pendekatan berbasis SOAP memiliki keterbatasan sebagai berikut :

1. Kompleksitas

Layanan yang berbasis SOAP membutuhkan banyak pengalaman karena kompleksitas layanan Web stack protokol. Selain itu, serialisasi dan deserializing permintaan tertulis dalam bahasa asli ke dalam pesan SOAP adalah memakan banyak waktu, yang bertentangan dengan keterbatasan website.

2. Aksesibilitas dan antarmuka

Layanan yang digunakan menggunakan point API tunggal. Oleh karena itu, semua fungsi layanan dan akses informasi menggunakan metode POST.

3. Interoperabilitas

Setiap layanan Web memiliki layanan antarmuka sendiri. Informasi deskripsi unik untuk setiap layanan menggunakan WSDL. Setelah klien menemukan layanan, informasi yang ditemukan bersifat mengikat dan tertutup.

4. Kinerja

Kinerja yang tinggi akan menyebabkan overhead pada layanan web berbasis SOAP karena penggunaan XML dan pesan SOAP yang panjang. File WSDL dan Pesan SOAP biasanya, termasuk informasi yang berlebihan yang pada gilirannya meningkatkan lalu lintas jaringan dan akan memakan lebih banyak sumber daya.

5. Data Model

Pendekatan berbasis SOAP menyembunyikan data model belakang antarmuka layanan web. Fitur ini mendikte bahwa konsumen layanan dan penyedia harus berbagi model umum untuk berkomunikasi. Namun, para pendukung SOAP berpendapat bahwa menjaga model data jauh dari klien lebih aman dan kurang berisiko.

6. Skalabilitas

Pesan SOAP diterjemahkan hanya di luar web. Sejak konsumen dan penyedia membangun layanan. Skalabilitas merupakan masalah karena untuk mencapai integrasi yang tepat dengan web, maka kedua belah pihak perlu adanya komunikasi yang baik.

1.5.4 Web Service Desain Berbasis REST

Penyedia layanan web seperti Google, Amazon, Yahoo, dan eBay mengadopsi layanan web tentang pendekatan REST [8]. Dalam konsep REST penyedia layanan dapat menghindari degradasi kinerja yang dihasilkan dari penggunaan SOAP dan XML. Layanan yang dirancang dengan pendekatan tentang mengekspos fungsi layanan sebagai sumber daya Web, setiap sumber daya ditujukan dengan URL yang unik. Keuntungan dalam pendekatan REST jika diterapkan pada website adalah sebagai berikut :

1. Skalabilitas

Pendekatan tentang dasar skalabilitas HTTP.

2. Addressability

Masing-masing sumber daya memiliki URL sendiri, yang dapat diambil sementara untuk menavigasi pengguna, meskipun koneksi link yang ada hanya antara sumber daya.

3. Link dan Koneksi

Sumber daya dapat menghubungkan satu sama lain menggunakan hyperlink dan pengalihan negara dapat dikelola melalui referral link.

4. Stateless

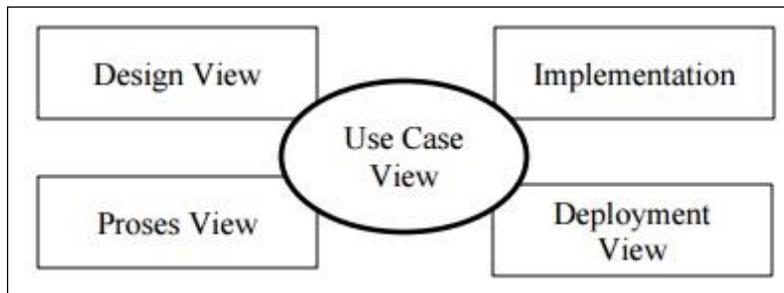
Permintaan dalam pendekatan ini memungkinkan kemampuan untuk menghapus informasi terkait dengan permintaan yang sudah selesai. Prinsip SISA mendikte bahwa pesan HTTP harus "Self-deskriptif", yang mengisaratkan bahwa setiap perantara node dapat sepenuhnya menginterpretasikan pesan, memahaminya, dan mengambil tindakan pada isinya.

5. Terpadu Interface

Semua sumber daya yang dinamis ditangani oleh metode HTTP standar, yaitu, GET, PUT, DELETE, dan POST. Setiap klien HTTP bisa berkomunikasi langsung dengan server HTTP tanpa konfigurasi khusus lebih lanjut. Alasan utama untuk ini adalah karena SOAP adalah kerangka protokol, sedangkan HTTP adalah protokol aplikasi.

1.6 Unified Modelling Language (UML)

UML (Unified Modeling Language) adalah metode pemodelan secara visual sebagai sarana untuk merancang dan atau membuat software berorientasi objek [10]. Karena UML ini merupakan bahasa visual untuk pemodelan bahasa berorientasi objek, maka semua elemen dan diagram berbasiskan pada paradigma object oriented. UML adalah salah satu tool atau model untuk merancang pengembangan software yang berbasis object oriented. UML sendiri juga memberikan standar penulisan sebuah sistem blue print, yang meliputi konsep bisnis proses, penulisan kelas-kelas dalam bahasa program yang spesifik, skema database, dan komponenkomponen yang diperlukan dalam sistem software. Dengan menggunakan UML kita dapat membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun.



Gambar 2.6 Model 4+1 View

Penjelasan model 4+1 view adalah sebagai berikut :

1. Use Case View
Mendefinisikan perilaku eksternal sistem
2. Design View
Mendefinisikan struktur logika yang mendukung fungsi-fungsi yang dibutuhkan use case (computer program, kelas-kelas, interkasi)
3. Implementation View
Menjelaskan komponen-komponen fisik dari sistem (file exe, library, database)
4. Process View
Berkaitan dengan concurrency di dalam sistem
5. Deployment View
Menjelaskan bagaimana komponen-komponen fisik didistribusikan.

1.6.1 Diagram UML

UML sendiri terdiri atas pengelompokan diagram-diagram sistem menurut aspek atau sudut pandang tertentu. Diagram adalah yang menggambarkan permasalahan maupun solusi dari permasalahan suatu model [10].

Diagram UML terdiri dari :

1. Diagram Use Case

Use case diagram digunakan untuk memperlihatkan hubungan antar actor dan use case. Use case digunakan untuk menggambarkan keadaan website dengan pengguna dan admin.

2. Diagram Class

Untuk menggambarkan sebuah sistem yang berhubungan dengan atribut dan operasi pada sebuah website.

3. Behavior Diagram

Diagram yang menunjukkan kegiatan yang ada pada sistem. Behavior diagram dibagi menjadi dua yaitu:

a. Statechart Diagram

Diagram ini menunjukkan kemungkinan dari keadaan objek dan proses yang menyebabkan perubahan pada keadaannya.

b. Activity Diagram

Diagram aktifitas digunakan untuk menggambarkan kegiatan yang ada pada sistem yang berkaitan dengan web, pengguna dan admin.

4. Interaction Diagram

Diagram yang menggambarkan interaksi antar objek. Diagram interaksi dibagi menjadi dua yaitu

a. Sequence Diagram

Digunakan untuk menggambarkan interaksi antar web, admin dan pengguna.

b. Collaboration Diagram

Diagram membawa informasi yang sama dengan diagram Sequence, tetapi lebih memusatkan atau memfokuskan pada kegiatan objek dari waktu pesan itu dikirimkan.

5. Component Diagram

Menampilkan komponen yang ada pada sistem dan hubungannya.

6. Deployment Diagram

Menggambarkan tentang proses yang berjalan pada suatu sistem dan relasi yang ada didalam website.

UML adalah sebuah “bahasa” yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem. Dengan menggunakan UML dapat dibuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. Tetapi karena UML juga menggunakan class dan operation dalam konsep dasarnya, maka lebih cocok untuk penulisan piranti lunak dalam bahasa berorientasi objek seperti C++, Java, atau php.

1.7 Black Box Testing

Black-box testing adalah metode pengujian perangkat lunak yang tes fungsionalitas dari aplikasi yang bertentangan dengan struktur internal atau kerja (lihat pengujian white-box) [11]. Pada Reengineering website evaluasi dibutuhkan mengetahui apakah perancangan yang di sarankan berfungsi dengan baik atau tidak. Oleh karena itu pengujian balck box dilakukan untuk mengetahui pengetahuan khusus dari kode aplikasi dan pengetahuan pemrograman pada umumnya tidak diperlukan.

Uji kasus dibangun di sekitar spesifikasi dan persyaratan, yakni, aplikasi apa yang seharusnya dilakukan. Menggunakan deskripsi eksternal perangkat lunak, termasuk spesifikasi, persyaratan, dan desain untuk menurunkan uji kasus. Tes ini dapat menjadi fungsional atau non-fungsional, meskipun biasanya fungsional. Perancang uji memilih input yang valid dan tidak valid dan menentukan output yang benar. Tidak ada pengetahuan tentang struktur internal benda uji itu. Metode uji dapat diterapkan pada semua tingkat pengujian perangkat lunak: unit, integrasi, fungsional, sistem dan penerimaan. Ini biasanya terdiri dari kebanyakan jika tidak semua pengujian pada tingkat yang lebih tinggi, tetapi juga bisa mendominasi unit testing juga.

Pengujian black box digunakan untuk menemukan kesalahan dalam beberapa kategori, diantaranya adalah :

1. Fungsi-fungsi yang salah atau hilang pada website

2. Kesalahan interface
3. Kesalahan dalam struktur data atau akses database eksternal
4. Kesalahan performa
5. kesalahan inisialisasi dan terminasi

1.8 Astah

Astah merupakan tool pemodelan secara visual untuk pengembangan system berbasis objek yang sangat handal untuk digunakan sebagai bantuan bagi para pengembang dalam melakukan analisis dan perancangan sistem. Astah digunakan untuk melakukan pemodelan sistem sebelum pengembang menulis kode – kode dalam bahasa pemrograman tertentu [12]. Astah mendukung pemodelan bisnis, yang membantu para pengembang untuk memahami system secara komprehensif. Astah juga membantu analisis sistem dengan cara pengembang membuat diagram use case untuk melihat fungsionalitas sistem secara keseluruhan sesuai dengan harapan dan keinginan pengguna.

Astah juga menuntut pengembang untuk membangun Interaction Diagram untuk melihat bagaimana objek – objek saling bekerja sama dalam menyediakan fungsionalitas yang diperlukan. Diagram Class juga dapat dibuat untuk melihat kelas – kelas yang terlibat dalam suatu sistem dan bagaimana mereka saling berhubungan satu sama lain. Diagram komponen dapat dikembangkan untuk menggambarkan bagaimana kelas – kelas dipetakan menjadi komponen implementasi. Terakhir deployment diagram dapat juga dikembangkan untuk melihat penyebaran komponen – komponen dalam jaringan komputer yang ada untuk sistem atau perangkat lunak yang sedang dikembangkan. Dalam Astah, pemodelan adalah cara melihat sistem dari berbagai sudut pandang. Astah mencakup semua diagram yang dikenal dalam UML, aktor – aktor yang terlibat dalam sistem, use case – use case, objek – objek, kelas – kelas, komponen – komponen deployment node. Model juga mendeskripsikan rincian yang diperlukan sistem dan bagaimana sistem bekerja, sehingga para pengembang dapat menggunakan model itu sebagai blue print untuk sistem yang akan dikembangkan. Beberapa keunggulan Astah dalam pemodelan, diantaranya :

1. Bahasa yang digunakan adalah bahasa pemodelan standar yaitu UML, dimana notasi standar dalam UML akan dapat meningkatkan komunikasi antar tim.
2. Astah mendukung round trip engineering, sehingga dapat di-generate model kedalam kode (Java, C ++, Visual Basic, dan sebagainya) dan melakukan reverse engineering

untuk menampilkan arsitektur software dari kode yang ada. Hal ini dapat dilakukan secara bolak – balik sebagai proses iteratif selama proses rekayasa software.

3. Membangun software menggunakan Astah memudahkan dalam memperbaiki software tersebut karena apabila suatu saat ditemukan requirement baru, dapat kembali digambarkan lagi software tersebut dalam UML.

