



**APLIKASI MANAJEMEN INSIDEN  
PADA PENGEMBANGAN PERANGKAT LUNAK  
STUDI KASUS DI PT. FUJITSU INDONESIA**

**NUROCHIM**

**41816110210**

**PROGRAM STUDI SISTEM INFORMATIKA  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS MERCU BUANA**

**2017**

## ABSTRAK

Dalam mengembangkan sebuah software, diperlukan manajemen insiden agar insiden yang masuk bisa tertata dengan baik dan terdistribusi secara merata kepada anggota tim. Sehingga kepuasan klien terhadap software yang mereka pesan tetap terjaga. PT. Fujitsu Indonesia adalah perusahaan yang bergerak dibidang informasi dan teknologi yang salah satu layanannya adalah menerima pengembangan software untuk perusahaan tertentu. PT. Fujitsu Indonesia dalam mengerjakan sebuah proyek pengembangan software, dokumentasi insiden yang dilaporkan oleh klien masih menggunakan microsoft Excel.

Penelitian ini bertujuan untuk mengembangkan aplikasi manajemen insiden yang akan digunakan untuk membantu PT. Fujitsu Indonesia dalam mengelola insiden yang dilaporkan oleh klien. Untuk itu dilakukan proses pengembangan perangkat lunak yang didasarkan pada rekayasa perangkat lunak yang benar. Model pengembangan perangkat lunak yang digunakan dalam penelitian ini adalah model *waterfall*. Laporan skripsi ini akan menguraikan aktifitas - aktifitas dan produk - produk yang dihasilkan pada masing - masing tahap pengembangan.

Desain dan implementasi untuk Aplikasi Manajemen Insiden ini dibatasi pada empat modul, yaitu Modul User, Modul Pencatatan Insiden, Modul Analisis Insiden dan Modul Monitoring Insiden. Pada tahap akhir pengembangan perangkat lunak, dilakukan evaluasi terhadap proses dan produk pengembangan perangkat lunak. Hal - hal apa yang telah dilakukan dan apa yang belum dilakukan pada pengembangan perangkat lunak ini akan diulas pada bagian akhir skripsi ini.

*Kata kunci: Aplikasi Manajemen Insiden,*

*Rekayasa Perangkat Lunak*

## **ABSTRAK**

In developing a software, incident management is required to ensure that the incidents that enter can be well organized and distributed equally to team members. So that the client's satisfaction with the software that they order is maintained. PT. Fujitsu Indonesia is a company engaged in information and technology that one of its services is to receive software development for certain companies. PT. Fujitsu Indonesia is working on a software development project, incident documentation reported by clients still using Microsoft Excel.

The goal of this study is to develop Incident Management Applications that will be used to assist PT. Fujitsu Indonesia in managing incidents reported by clients. To achieve it, software development process is done based on discipline software engineering. Software development model used in this study is waterfall model. This thesis will explain the activities and results produced in each development phases.

Scope of design and implementation of Incident Management Applications are limited to four module, that is Modul User, Modul Input Incident, Modul Analyzing Incident, and Modul Monitoring Incident. At the last phase, software development process and results are evaluated. What has been done and what has not been done through this software development will be discussed in the last part of this thesis.

**Keywords:** Management Incident, Management Issue, Management Bugs,  
Software Engineering

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Pada proyek pengembangan software skala besar, sering terjadi penanganan insiden yang tidak di manajemen dengan baik. Insiden yang tidak di manajemen dengan baik tersebut bisa menjadi bumerang yang mengakibatkan perusahaan pengembang mengalami kerugian besar. Setiap insiden yang ditemukan oleh klien adalah ibarat bola salju yang menggelinding, semakin cepat berputarnya semakin besar bola saljunya. Jika terus berlanjut, pihak pengembang akan kehilangan kepercayaan dari klien, sehingga klien merasa mendapatkan layanan yang tidak baik. Pihak pengembang bisa dituntut secara hukum untuk membayar denda sesuai tuntutan. Kerugian lain yang mengikuti adalah tercemarnya nama baik perusahaan pengembang, sehingga tidak lagi dipercaya untuk menerima proyek lain.

PT. Fujitsu Indonesia adalah perusahaan yang bergerak dibidang informasi dan teknologi. Produk yang ditawarkan adalah berbagai produk teknologi, solusi dan layanan service. Berdiri sejak tahun 1995 dengan kantor pusat berada di Jakarta. Salah satu layanan yang diberikan PT. Fujitsu Indonesia adalah mengembangkan software untuk kebutuhan perusahaan tertentu.

PT. Fujitsu Indonesia dalam mengerjakan sebuah proyek pengembangan software, dokumentasi insiden yang dilaporkan oleh klien masih menggunakan microsoft Excel. Dokumen tersebut kemudian di distribusikan lewat email dan disimpan di komputer masing-masing anggota tim. Saat ada perbaikan isi dokumen, kendala utamanya adalah lamanya proses *synchronize* dari dokumen yang terpisah-pisah tadi, sehingga informasi yang sampai ke anggota tim menjadi tidak *update*, dan respon tim terhadap insiden menjadi lambat.

Penelitian yang berhubungan dengan penelitian ini adalah penelitan yang dilakukan oleh Khurotul Aeni, Bambang Soedjino WA, Henderi dari STMIK AMIKOM Yogyakarta pada tahun 2016 tentang “Membuat Dokumen Tata Laksana Manajemen Insiden Berdasarkan Framework IT Infrastructure Library V.3” dihasilkan sebuah strategi pembuatan dokumen tata laksana manajemen insiden pada sistem informasi akademik SIMBARU yang dimiliki Universitas Peradaban. Dokumen tersebut digunakan sebagai dasar strategi perbaikan insiden sesuai dengan framework IT Infrastruktur Library V.3. Penelitian ini membuktikan bahwa untuk menangani insiden dengan baik, diperlukan sebuah dokumen yang lengkap yang mengidentifikasi sebab terjadinya insiden.

Penelitian yang juga berhubungan dengan penelitian ini adalah penelitian yang dilakukan oleh Husni Faqih mahasiswa AMIK BSI Tegal pada tahun 2017 tentang “Implementasi Redmine dan VisualSVN Dalam Manajemen Proyek Website Toko Baju Online” menghasilkan sebuah kesimpulan bahwa aplikasi Redmine dapat digunakan untuk membantu meningkatkan kerja sama tim proyek dan memudahkan kontrol terhadap tugas dan insiden. Redmine adalah aplikasi manajemen proyek berbasis web yang fleksibel. Dibangun menggunakan kerangka Ruby on Rails. Redmine dapat digunakan dalam lintas platform dan lintas basis data. Salah satu fitur yang dimiliki oleh aplikasi redmine adalah *Flexible issue tracking system* (sistem pelacakan insiden yang flexible).

Penelitian lain yang berhubungan dengan penelitian ini adalah penelitian yang tentang Redmine dilakukan oleh Corie Mei Hellyana mahasiswa dari AMIK BSI Purwokerto pada tahun 2015 tentang “Penggunaan Sistem Aplikasi Redmine Dan Subversion Dalam Manajemen Proyek” menghasilkan kesimpulan bahwa *Redmine* dapat digabungkan dengan aplikasi *Subversion* dan *Tortoise* sehingga dapat mempercepat komunikasi dalam tim, membantu manajer proyek untuk memonitor pekerjaan tim secara *realtime*.

Dari penelitian-penelitian diatas masih sedikit peneliti yang memfokuskan penelitiannya pada pengembangan sistem informasi manajemen insiden. Dalam hal ini penulis ingin mengembangkan sistem informasi manajemen insiden berbasis web pada PT. Fujitsu Indonesia. Sehingga dengan sistem yang dikembangkan dapat membantu mengelola insiden dengan lebih baik dan kerja sama tim dalam menangani insiden menjadi lebih efektif.

## **1.2 Perumusan Masalah**

Dari berbagai latar belakang yang telah diutarakan di atas, maka dapat dirumuskan “Bagaimana membangun aplikasi Sistem Informasi Manajemen Insiden Berbasis Web pada PT. Fujitsu Indonesia”.

## **1.3 Tujuan dan Manfaat**

Tujuan penelitian yang penulis lakukan adalah untuk membuat sebuah sistem informasi manajemen insiden dengan studi kasus pada PT. Fujitsu Indonesia. Dengan sistem ini, diharapkan nantinya seluruh aktivitas saat menangani insiden tercatat dengan akurat dan terorganisir dengan baik.

## **1.4 Batasan Masalah**

Agar aplikasi yang dibuat tidak menyimpang dari permasalahan yang ada, maka perlu dibuat suatu pembatasan ruang lingkupnya. Adapun ruang lingkup dalam aplikasi ini antara lain:

1. Aplikasi yang dibangun mencakup pencatatan insiden, pembagian tugas kepada tim, mencatat akar masalah (*root cause*) dan solusi (*solution*) dari insiden.
2. Aplikasi yang akan dibangun dapat menampilkan laporan perkembangan insiden setiap hari.
3. Bahasa yang pemrograman yang akan digunakan yaitu *PHP* dan menggunakan database *MySQL* sebagai media penyimpanan.

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Aplikasi**

Pengertian Aplikasi Menurut Kamus Kamus Besar Bahasa Indonesia (1998:52) adalah penerapan dari rancang sistem untuk mengolah data yang menggunakan aturan atau ketentuan bahasa pemrograman tertentu. Aplikasi adalah suatu program komputer yang dibuat untuk mengerjakan dan melaksanakan tugas khusus dari pengguna

#### **2.2 Manajemen Insiden Perangkat Lunak**

##### **2.2.1 Manajemen**

Kata Manajemen berasal dari bahasa Perancis kuno *ménagement*, yang memiliki arti "seni melaksanakan dan mengatur." Kata manajemen mungkin berasal dari bahasa Italia (1561) *maneggiare* yang berarti "mengendalikan," terutama dalam konteks mengendalikan kuda, yang berasal dari bahasa latin *manus* yang berarti "tangan". Bahasa Prancis lalu mengadopsi kata ini dari bahasa Inggris menjadi *ménagement*, yang memiliki arti seni melaksanakan dan mengatur.

Fungsi manajemen adalah elemen-elemen dasar yang akan selalu ada dan melekat di dalam proses manajemen yang akan dijadikan acuan oleh manajer dalam melaksanakan kegiatan untuk mencapai tujuan. Fungsi manajemen pertama kali diperkenalkan oleh seorang industrialis Perancis bernama Henry Fayol pada awal abad ke-20. Ketika itu, ia menyebutkan lima fungsi manajemen, yaitu merancang, mengorganisir, memerintah, mengordinasi, dan mengendalikan. Namun saat ini, kelima fungsi tersebut telah diringkas menjadi tiga, yaitu:

1. **Perencanaan (planning)** adalah memikirkan apa yang akan dikerjakan dengan sumber yang dimiliki. Perencanaan dilakukan untuk menentukan tujuan perusahaan secara keseluruhan dan cara terbaik untuk memenuhi tujuan itu. Manajer mengevaluasi berbagai rencana alternatif sebelum mengambil tindakan dan kemudian melihat apakah rencana yang dipilih cocok dan dapat digunakan untuk memenuhi tujuan perusahaan. Perencanaan merupakan proses terpenting dari semua fungsi manajemen karena tanpa perencanaan, fungsi-fungsi lainnya tak dapat berjalan.
2. **Pengorganisasian (organizing)** dilakukan dengan tujuan membagi suatu kegiatan besar menjadi kegiatan-kegiatan yang lebih kecil. Pengorganisasian mempermudah manajer dalam melakukan pengawasan dan menentukan orang yang dibutuhkan untuk melaksanakan tugas yang telah dibagi-bagi tersebut. Pengorganisasian dapat dilakukan dengan cara menentukan tugas apa yang harus dikerjakan, siapa yang harus mengerjakannya, bagaimana tugas-tugas tersebut dikelompokkan, siapa yang bertanggung jawab atas tugas tersebut, dan pada tingkatan mana keputusan harus diambil.
3. **Pengarahan (directing)** adalah suatu tindakan untuk mengusahakan agar semua anggota kelompok berusaha untuk mencapai sasaran sesuai dengan perencanaan manajerial dan usaha.

### 2.2.2 Insiden

Insiden merupakan kejadian yang menyebabkan perubahan sistem keamanan dari perusahaan atau perubahan kebijakan ke arah yang merugikan. Insiden juga dapat dipahami sebagai kejadian yang tidak biasa bagi perusahaan, yang tidak dapat dijelaskan sebagai konsekuensi operasi perusahaan pada kondisi normal. Secara umum, insiden adalah kejadian yang menyebabkan kecelakaan, luka, kehilangan, kerugian finansial bagi para pekerja, pengunjung, pelajar,



sukarelawan, dan lain sebagainya. Sehubungan dengan adanya insiden, baik internal atau eksternal, struktur kepengurusan bukanlah hal yang sangat esensial, tetapi yang lebih penting adalah bagaimana kepengurusan yang ada mengatasi insiden yang terjadi dengan baik dan efektif sehingga tidak mengganggu operasi perusahaan yang sedang berlangsung. Jadi kemampuan untuk bereaksi cepat dan terarah terhadap insiden yang terjadi merupakan komponen penting dari keseluruhan rencana mengatasi insiden. Untuk itu, diperlukan tim yang khusus menangani insiden, yang telah dilatih agar mampu melaporkan dan menangani insiden dengan efektif. Tentu saja, hal yang sangat diinginkan adalah semua staf perusahaan memiliki kemampuan untuk mengatasi insiden dengan efektif. Hal ini dapat direalisasikan bila pimpinan atau manajer dari setiap departemen di dalam suatu perusahaan dapat mengarahkan dan memberikan training ke para pekerjanya mengenai isi kebijakan dan prosedur yang dapat dilakukan bila insiden terjadi.

Insiden yang terjadi dapat menyebabkan kerusakan atau permasalahan yang serius atau besar, baik bagi para konsumen, masyarakat luas, para pekerja, pimpinan perusahaan, dan lain sebagainya. Tim prosedur insiden perlu memastikan bahwa risiko yang harus diterima dapat diminimisasi dan semua usaha yang dilakukan berdasarkan pengalaman dan latihan, sistem kerja, kebijakan, dan prosedur yang terus dikembangkan.

### **2.2.3 Perangkat Lunak**

Perangkat lunak atau peranti lunak (bahasa Inggris: software) adalah istilah khusus untuk data yang diformat, dan disimpan secara digital, termasuk program komputer, dokumentasinya, dan berbagai informasi yang bisa dibaca, dan ditulis oleh komputer. Dengan kata lain, bagian sistem komputer yang tidak berwujud. Istilah ini menonjolkan perbedaan dengan perangkat keras komputer.

Pembuatan perangkat lunak itu sendiri memerlukan "bahasa pemrograman" yang ditulis oleh seorang pemrogram untuk selanjutnya dikompilasi dengan aplikasi kompiler sehingga menjadi kode yang bisa dikenali oleh mesin hardware.

## **2.3 Rekayasa Perangkat Lunak**

RPL (Rekayasa Perangkat Lunak) adalah suatu disiplin ilmu yang mempelajari semua aspek produksi dalam suatu proses perancangan suatu perangkat lunak / sistem dengan tujuan untuk dapat menghasilkan suatu perangkat lunak / sistem dengan cara yang efektif dan efisien dari segi biaya, waktu, dan tenaga. Pada dasarnya rekayasa perangkat lunak lebih memfokuskan pada tahapantahapan pengembangan suatu perangkat lunak yakni :*Analisis, Desain, Implementasi, Pengujian, Pemeliharaan, dan Dokumentasi.*(Sommerville, 2003)

## **2.4 Tahap Pengembangan Sistem**

Pada perancangan aplikasi ini penulis menggunakan metode *waterfall* "*waterfall modeling*". Karena metode tersebut merupakan metode yang merujuk ke arah pendekatan yang sistematis dan sekuensial melalui tahap-tahap yang ada pada SDLC untuk membangun sebuah aplikasi perangkat lunak.

### **2.4.1 Metode System Development Life Cycle (SDLC)**

Mengutip dari buku *Software Engineering : Seventh Edition* karangan Roger S. Pressman yang diterbitkan oleh McGraw-hill, pengertian dari SDLC adalah : "*System Development Life Cycle* atau siklus hidup pengembangan sistem adalah proses perancangan sistem serta metodologi yang digunakan untuk mengembangkan sistem-sistem tersebut."

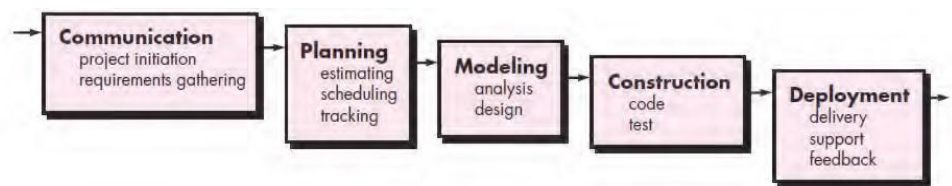
Dalam penelitian skripsi ini, penulis menggunakan proses *Waterfall Model* sebagai pola pengembangan sistem. Definisi yang diartikan oleh Roger S. Pressman dalam bukunya *Software Engineering : Seventh Edition* dijelaskan bahwa :

"*Waterfall Model* sebuah proses perancangan yang secara berurutan dan sering digunakan dalam proses pengembangan perangkat lunak."

#### 2.4.2 *Waterfall Model*

Menurut Pressman (2010, p39) *Waterfall Model* adalah model klasik yang bersifat sistematis, berurutan dalam membangun *software*. Berikut ini gambaran dari *Waterfall Model*.

Fase-fase dalam *Waterfall Model* menurut Referensi Pressman :



Gambar 2.1 *Waterfall Model*  
(Sumber : Pressman, 2010, p39)

Berikut adalah gambar pengembangan perangkat lunak berurutan/ linear :

- a. Communication Langkah ini merupakan analisis terhadap kebutuhan software, dan tahap untuk mengadakan pengumpulan data dengan melakukan pertemuan dengan customer, maupun mengumpulkan data-data tambahan baik yang ada di jurnal, artikel, maupun dari internet.
- b. Planning Proses planning merupakan lanjutan dari proses communication (analysis requirement). Tahapan ini akan menghasilkan dokumen user requirement atau bisa dikatakan sebagai data yang berhubungan dengan keinginan user dalam pembuatan software, termasuk rencana yang akan dilakukan.
- c. Modeling Proses modeling ini akan menerjemahkan syarat kebutuhan ke sebuah perancangan software yang dapat diperkirakan sebelum dibuat coding. Proses ini berfokus pada rancangan struktur data, arsitektur software, representasi

interface, dan detail (algoritma) prosedural. Tahapan ini akan menghasilkan dokumen yang disebut software requirement.

- d. Construction Construction merupakan proses membuat kode. Coding atau pengkodean merupakan penerjemahan desain dalam bahasa yang bisa dikenali oleh komputer. Programmer akan menerjemahkan transaksi yang diminta oleh user. Tahapan inilah yang merupakan tahapan secara nyata dalam mengerjakan suatu software, artinya penggunaan komputer akan dimaksimalkan dalam tahapan ini. Setelah pengkodean selesai maka akan dilakukan testing terhadap sistem yang telah dibuat tadi. Tujuan testing adalah menemukan kesalahan-kesalahan terhadap sistem tersebut untuk kemudian bisa diperbaiki.
- e. Deployment Tahapan ini bisa dikatakan final dalam pembuatan sebuah software atau sistem. Setelah melakukan analisis, desain dan pengkodean maka sistem yang sudah jadi akan digunakan oleh user. Kemudian software yang telah dibuat harus dilakukan pemeliharaan secara berkala.

Keuntungan Dan Kekurangan Metode Waterfall Keuntungan yang didapat dengan menggunakan teknik waterfall ini adalah sebagai berikut :

- a. Proses menjadi teratur.
- b. Estimasi proses menjadi lebih baik.

Selain itu, teknik waterfall juga memiliki kekurangan, kekurangannya adalah sebagai berikut :

- c. Sifatnya kaku, sehingga susah melakukan perubahan ditengah proses.
- d. Jadwal menjadi lebih tak menentu.

## **2.5 UML (Unified Modeling Language)**

### **2.5.1 Use Case Diagram**

Use Case Diagram menekankan pada ‘apa’ yang akan dikerjakan oleh sistem tanpa mempedulikan bagaimana sistem akan melakukannya secara rinci (penggambaran fungsionalitas sistem yang berinteraksi dengan aktor). Contohnya login ke sistem, mengcreate sebuah daftar belanja, dan sebagainya. Dalam diagram sebuah objek aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu. Diagram ini membantu penyusunan requirement sebuah sistem, mengkomunikasikan rancangan dengan klien, dan merancang test case untuk semua feature yang ada pada sistem. Sebuah use case dapat meng-include fungsionalitas use case lain sebagai bagian dari proses dalam dirinya. Setiap kali use case yang meng-include dieksekusi secara normal maka use case yang di-include akan dipanggil. Sebuah use case dapat di-include oleh lebih dari satu use case lain, sehingga duplikasi fungsionalitas dapat dihindari dengan cara menarik keluar fungsionalitas yang common. Sebuah use case juga dapat meng-extend use case lain dengan behaviour -nya sendiri. Sementara hubungan generalisasi antar use case menunjukkan bahwa use case yang satu merupakan spesialisasi dari yang lain.

### **2.5.2 Activity Diagram**

Activity diagrams menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, decision yang mungkin terjadi, dan bagaimana mereka berakhir. Activity diagram juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. Activity diagram merupakan state diagram khusus, di mana sebagian besar state adalah action dan sebagian besar transisi di-trigger oleh selesainya state sebelumnya (internal processing). Oleh karena itu activity diagram tidak menggambarkan behaviour internal sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih

menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum. Sebuah aktivitas dapat direalisasikan oleh satu use case atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara use case menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas. Sama seperti state, standar UML menggunakan segiempat dengan sudut membulat untuk menggambarkan aktivitas. Decision digunakan untuk menggambarkan behaviour pada kondisi tertentu. Untuk mengilustrasikan proses-proses paralel ( fork dan join ) digunakan titik sinkronisasi yang dapat berupa titik, garis horizontal atau vertikal. Activity diagram dapat dibagi menjadi beberapa object swimlane untuk menggambarkan objek mana yang bertanggung jawab untuk aktivitas tertentu.

### **2.5.3 Class Diagram**

Class merupakan inti dari pengembangan dan desain berorientasi objek. Class menggambarkan keadaan (atribut/properti) suatu sistem dan manipulasi keadaan tersebut (metoda/fungsi). Class diagram menggambarkan struktur dan deskripsi class, package dan objek beserta hubungan satu sama lain seperti containment, pewarisan, asosiasi, dan lain-lain.

Atribut dan metoda dapat memiliki salah satu sifat berikut :

- Private, tidak dapat dipanggil dari luar class yang bersangkutan
- Protected, hanya dapat dipanggil oleh class yang bersangkutan dan anak-anak yang mewarisinya
- Public, dapat dipanggil oleh siapa saja

Class dapat merupakan implementasi dari sebuah interface, yaitu class abstrak yang hanya memiliki metoda. Interface karena tidak dapat langsung diinstansiasikan, tetapi harus diimplementasikan dahulu menjadi sebuah class. Sesuai dengan

perkembangan class model, class dapat dikelompokkan menjadi package.

Hubungan Antar Class :

1. Asosiasi, yaitu hubungan statis antar class. Umumnya menggambarkan class yang memiliki atribut berupa class lain, atau class yang harus mengetahui eksistensi class lain. Panah navigability menunjukkan arah query antar class.
2. Agregasi, yaitu hubungan yang menyatakan bagian (“terdiri atas..”).
3. Pewarisan, yaitu hubungan hirarkis antar class. Class dapat diturunkan dari class lain dan mewarisi semua atribut dan metoda class asalnya dan menambahkan fungsionalitas baru, sehingga ia disebut anak dari class yang diwarisinya. Kebalikan dari pewarisan adalah generalisasi.
4. Hubungan dinamis, yaitu rangkaian pesan ( message ) yang di-passing dari satu class kepada class lain. Hubungan dinamis dapat digambarkan dengan menggunakan sequence diagram yang akan dijelaskan kemudian.

#### **2.5.4 Sequence Diagram**

Sequence diagram menggambarkan 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. Masing-masing objek, termasuk aktor, memiliki lifeline vertikal. Message digambarkan

sebagai garis berpanah dari satu objek ke objek lainnya. Pada fase desain berikutnya, message akan dipetakan menjadi operasi/metoda dari classx. Activation bar menunjukkan lamanya eksekusi sebuah proses, biasanya diawali dengan diterimanya sebuah message. Untuk objek-objek yang memiliki sifat khusus, standar UML mendefinisikan icon khusus untuk objek boundary, controller dan persistent entity .

## **2.6 Hypertext Markup Language (HTML)**

Hypertext Markup Language (HTML) adalah sebuah bahasa markah yang digunakan untuk membuat sebuah halaman web, menampilkan berbagai informasi di dalam sebuah penjelajah web Internet dan pemformatan hiperteks sederhana yang ditulis dalam berkas format ASCII agar dapat menghasilkan tampilan wujud yang terintegrasi. Dengan kata lain, berkas yang dibuat dalam perangkat lunak pengolah kata dan disimpan dalam format ASCII normal sehingga menjadi halaman web dengan perintah-perintah HTML. Bermula dari sebuah bahasa yang sebelumnya banyak digunakan di dunia penerbitan dan percetakan yang disebut dengan SGML (Standard Generalized Markup Language), HTML adalah sebuah standar yang digunakan secara luas untuk menampilkan halaman web. HTML saat ini merupakan standar Internet yang didefinisikan dan dikendalikan penggunaannya oleh World Wide Web Consortium (W3C). HTML dibuat oleh kolaborasi Caillau TIM dengan Berners-lee Robert ketika mereka bekerja di CERN pada tahun 1989 (CERN adalah lembaga penelitian fisika energi tinggi di Jenewa)

### **2.6.1 HTML 5**

HTML5 adalah sebuah bahasa markah untuk menstrukturkan dan menampilkan isi dari Waring Wera Wanua, sebuah teknologi inti dari Internet. HTML5 adalah revisi kelima dari HTML (yang pertama kali diciptakan pada tahun 1990 dan versi keempatnya, HTML4, pada tahun 1997[1]) dan hingga bulan Juni 2011 masih dalam pengembangan. Tujuan utama pengembangan HTML5 adalah untuk memperbaiki teknologi HTML agar mendukung teknologi



multimedia terbaru, mudah dibaca oleh manusia dan juga mudah dimengerti oleh mesin.

HTML5 merupakan salah satu karya Konsortium Waring Wera Wanua (World Wide Web Consortium, W3C) untuk mendefinisikan sebuah bahasa markah tunggal yang dapat ditulis dengan cara HTML ataupun XHTML. HTML5 merupakan jawaban atas pengembangan HTML 4.01 dan XHTML 1.1 yang selama ini berjalan terpisah, dan diimplementasikan secara berbeda-beda oleh banyak perangkat lunak pembuat web.

## **2.7 PHP**

Pada awalnya PHP merupakan kependekan dari Personal Home Page (Situs personal). PHP pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. Pada waktu itu PHP masih bernama Form Interpreted (FI), yang wujudnya berupa sekumpulan skrip yang digunakan untuk mengolah data formulir dari web.

Selanjutnya Rasmus merilis kode sumber tersebut untuk umum dan menamakannya PHP/FI. Dengan perilis kode sumber ini menjadi sumber terbuka, maka banyak pemrogram yang tertarik untuk ikut mengembangkan PHP.

Pada November 1997, dirilis PHP/FI 2.0. Pada rilis ini, interpreter PHP sudah diimplementasikan dalam program C. Dalam rilis ini disertakan juga modul-modul ekstensi yang meningkatkan kemampuan PHP/FI secara signifikan.

Pada tahun 1997, sebuah perusahaan bernama Zend menulis ulang interpreter PHP menjadi lebih bersih, lebih baik, dan lebih cepat. Kemudian pada Juni 1998, perusahaan tersebut merilis interpreter baru untuk PHP dan meresmikan rilis tersebut sebagai PHP 3.0 dan singkatan PHP diubah menjadi akronim berulang PHP: Hypertext Preprocessing.

Pada pertengahan tahun 1999, Zend merilis interpreter PHP baru dan rilis tersebut dikenal dengan PHP 4.0. PHP 4.0 adalah versi PHP yang paling banyak dipakai pada awal abad ke-21. Versi ini banyak dipakai

disebabkan kemampuannya untuk membangun aplikasi web kompleks tetapi tetap memiliki kecepatan dan stabilitas yang tinggi.

Pada Juni 2004, Zend merilis PHP 5.0. Dalam versi ini, inti dari interpreter PHP mengalami perubahan besar. Versi ini juga memasukkan model pemrograman berorientasi objek ke dalam PHP untuk menjawab perkembangan bahasa pemrograman ke arah paradigma berorientasi objek. Server web bawaan ditambahkan pada versi 5.4 untuk mempermudah pengembang menjalankan kode PHP tanpa menginstall software server.

Versi terbaru dan stabil dari bahasa pemrograman PHP saat ini adalah versi 7.0.16 dan 7.1.2 yang resmi dirilis pada tanggal 17 Februari 201

## **Sintaksis Dasar**

### **1. Pembatas**

PHP hanya mengeksekusi kode yang ditulis dalam pembatas sebagaimana ditentukan oleh dasar sintaks PHP. Apapun di luar pembatas tidak diproses oleh PHP (meskipun teks PHP ini masih mengendalikan struktur yang dijelaskan dalam kode PHP. Pembatas yang paling umum adalah "<?php" untuk membuka dan ">" Untuk menutup kode PHP. Tujuan dari pembatas ini adalah untuk memisahkan kode PHP dari kode di luar PHP, seperti HTML, Javascript.

### **2. Variabel**

Variabel diawali dengan simbol dolar \$. Pada versi php PHP 5 diperkenalkan jenis isyarat yang memungkinkan fungsi untuk memaksa mereka menjadi parameter objek dari class tertentu, array, atau fungsi. Namun, jenis petunjuk tidak dapat digunakan dengan jenis skalar seperti angka atau string. Contoh variabel dapat ditulis sebagai \$nama\_variabel.

Penulisan fungsi, penamaan kelas, nama variabel adalah peka akan huruf besar (Kapital) dan huruf kecil . Kedua kutip ganda "" dari string memberikan kemampuan untuk interpolasi nilai variabel

ke dalam string PHP. PHP menerjemahkan baris sebagai spasi, dan pernyataan harus diakhiri dengan titik koma ;.

### 3. Komentar

PHP memiliki 3 jenis sintaks sebagai komentar pada kode yaitu tanda blok `/* */`, komentar 2 baris `//` Serta tanda pagar `#` digunakan untuk komentar satu baris. Komentar bertujuan untuk meninggalkan catatan pada kode PHP dan tidak akan diterjemahkan ke program.

### 4. Fungsi

Ratusan fungsi yang disediakan oleh PHP serta ribuan lainnya yang tersedia melalui berbagai ekstensi tambahan. fungsi-fungsi ini didokumentasikan dalam dokumentasi PHP. Namun, dalam berbagai tingkat pengembangan, kini memiliki berbagai konvensi penamaan.

## **Kelebihan PHP Dari Bahasa Pemrograman Lain**

Beberapa kelebihan PHP dari bahasa pemrograman web, antara lain:

- Bahasa pemrograman PHP adalah sebuah bahasa script yang tidak melakukan sebuah kompilasi dalam penggunaannya.
- Web Server yang mendukung PHP dapat ditemukan di mana - mana dari mulai apache, IIS, Lighttpd, hingga Xitami dengan konfigurasi yang relatif mudah.
- Dalam sisi pengembangan lebih mudah, karena banyaknya milis - milis dan developer yang siap membantu dalam pengembangan.
- Dalam sisi pemahaman, PHP adalah bahasa scripting yang paling mudah karena memiliki referensi yang banyak.
- PHP adalah bahasa open source yang dapat digunakan di berbagai mesin (Linux, Unix, Macintosh, Windows) dan dapat dijalankan secara runtime melalui console serta juga dapat menjalankan perintah-perintah system.

## **Tipe data**

PHP memiliki 8 tipe data, yaitu:

- Boolean
- Integer
- Float/ Double
- String
- Array
- Object
- Resource
- NULL

### **2.7.1 Laravel**

Laravel adalah sebuah framework PHP yang dirilis dibawah lisensi MIT, dibangun dengan konsep MVC (model view controller). Laravel adalah pengembangan website berbasis MVP yang ditulis dalam PHP yang dirancang untuk meningkatkan kualitas perangkat lunak dengan mengurangi biaya pengembangan awal dan biaya pemeliharaan, dan untuk meningkatkan pengalaman bekerja dengan aplikasi dengan menyediakan sintaks yang ekspresif, jelas dan menghemat waktu.

MVC adalah sebuah pendekatan perangkat lunak yang memisahkan aplikasi logika dari presentasi. MVC memisahkan aplikasi berdasarkan komponen- komponen aplikasi, seperti : manipulasi data, controller, dan user interface.

1. Model, Model mewakili struktur data. Biasanya model berisi fungsi-fungsi yang membantu seseorang dalam pengelolaan basis data seperti memasukkan data ke basis data, pembaruan data dan lain-lain.
2. View, View adalah bagian yang mengatur tampilan ke pengguna. Bisa dikatakan berupa halaman web.
3. Controller, Controller merupakan bagian yang menjembatani model dan view.

Beberapa fitur yang terdapat di Laravel :

- Bundles, yaitu sebuah fitur dengan sistem pengemasan modular dan tersedia beragam di aplikasi.
- Eloquent ORM, merupakan penerapan PHP lanjutan menyediakan metode internal dari pola “active record” yang mengatasi masalah pada hubungan objek database.
- Application Logic, merupakan bagian dari aplikasi, menggunakan controller atau bagian Route.
- Reverse Routing, mendefinisikan relasi atau hubungan antara Link dan Route.
- Restful controllers, memisahkan logika dalam melayani HTTP GET and POST.
- Class Auto Loading, menyediakan loading otomatis untuk class PHP.
- View Composer, adalah kode unit logikal yang dapat dieksekusi ketika view sedang loading.
- IoC Container, memungkinkan obyek baru dihasilkan dengan pembalikan controller.
- Migration, menyediakan sistem kontrol untuk skema database.
- Unit Testing, banyak tes untuk mendeteksi dan mencegah regresi.
- Automatic Pagination, menyederhanakan tugas dari penerapan halaman.

## 2.8 CSS

CSS adalah kependekan dari Cascading Style Sheet. CSS merupakan salah satu kode pemrograman yang bertujuan untuk menghias dan mengatur gaya tampilan/layout halaman web supaya lebih elegan dan menarik. CSS adalah sebuah teknologi internet yang direkomendasikan oleh World Wide Web Consortium atau W3C pada tahun 1996. Awalnya, CSS dikembangkan di SGML pada tahun 1970, dan terus dikembangkan hingga saat ini. CSS telah mendukung banyak bahasa markup seperti HTML, XHTML, XML,

SVG (Scalable Vector Graphics) dan Mozilla XUL (XML User Interface Language).

Pada desember 1996, W3C memperkenalkan Level 1 spesifikasi CSS atau juga dikenal CSS1 yang mendukung format, warna font teks, dan lain-lain. Kemudian, Mei 1998, W3C menerbitkan CSS2 yang di dalamnya diatur fungsi peletakan elemen. Dan sekarang, W3C telah memperbaiki dan meningkatkan Kemampuan CSS2 ke CSS3.

CSS digunakan oleh web programmer dan juga blogger untuk menentukan warna, tata letak font, dan semua aspek lain dari presentasi dokumen di situs mereka. Saat ini, hampir tidak ada situs web yang dibangun tanpa kode css.

### **2.8.1 Bootstrap**

Bootstrap adalah aplikasi sumber terbuka yang berupa framework atau kerangka kerja untuk membangun website dinamis dengan menggunakan bahasa script CSS. Bootstrap memudahkan developer dan designer untuk membuat sebuah aplikasi web menjadi cepat dan mudah dibandingkan dengan membuatnya dari awal.

#### **Kerangka Kerja Aplikasi**

Kerangka Kerja (Framework) secara sederhana diartikan sebagai kumpulan dari fungsi-fungsi/prosedur-prosedur dan class-class untuk tujuan tertentu yang sudah siap digunakan sehingga bisa lebih mempermudah dan mempercepat pekerjaan seorang programmer, tanpa harus membuat fungsi atau class dari awal.

#### **Fitur Umum Framework:**

- Mempercepat dan mempermudah pembangunan sebuah aplikasi web.

- Relatif memudahkan dalam proses maintenance karena sudah ada pola tertentu dalam sebuah framework (dengan syarat programmer mengikuti pola standar yang ada)
- Umumnya framework menyediakan fasilitas-fasilitas yang umum dipakai sehingga kita tidak perlu membangun dari awal.
- Lebih bebas dalam pengembangan jika dibandingkan CMS

## 2.9 Javascript

JavaScript adalah bahasa pemrograman tingkat tinggi dan dinamis. JavaScript populer di internet dan dapat bekerja di sebagian besar penjelajah web populer seperti Internet Explorer (IE), Mozilla Firefox, Netscape dan Opera. Kode JavaScript dapat disisipkan dalam halaman web menggunakan tag SCRIPT

JavaScript pertama kali dikembangkan oleh Brendan Eich dari Netscape di bawah nama Mocha, yang nantinya namanya diganti menjadi LiveScript, dan akhirnya menjadi JavaScript. Navigator sebelumnya telah mendukung Java untuk lebih bisa dimanfaatkan para programmer yang non-Java. Maka dikembangkanlah bahasa pemrograman bernama LiveScript untuk mengakomodasi hal tersebut. Bahasa pemrograman inilah yang akhirnya berkembang dan diberi nama JavaScript, walaupun tidak ada hubungan bahasa antara Java dengan JavaScript. JavaScript bisa digunakan untuk banyak tujuan, misalnya untuk membuat efek rollover baik di gambar maupun teks, dan yang penting juga adalah untuk membuat AJAX.[8] JavaScript adalah bahasa yang digunakan untuk AJAX.

### 2.9.1 JQuery

JQuery adalah pustaka JavaScript lintas-platform yang didesain untuk menyederhanakan client-side scripting pada HTML. Dewasa ini, JQuery merupakan pustaka JavaScript yang paling populer, dengan 65% pemasangan dari 10 juta situs Web dengan pengunjung tertinggi. JQuery adalah gratis, dan bersifat kode sumber terbuka dibawah lisensi MIT.

Sintaks pada JQuery didesain untuk memudahkan dalam navigasi sebuah dokumen, pemilihan elemen DOM, pembuatan animasi, penanganan event, dan pengembangan aplikasi berbasis Ajax. JQuery juga menyediakan kemampuan bagi para pengembang untuk dapat membuat plug-in pada pustaka JavaScript ini. Ini memungkinkan mereka untuk membuat abstraksi pada interaksi dan animasi tingkat-rendah, efek lanjutan, serta tampilan widget yang dapat dimodifikasi. Pendekatan modular pada JQuery memungkinkan kita dalam pembuatan halaman Web yang dinamis dan aplikasi berbasis Web yang ajib.

Sekumpulan fitur inti JQuery—yakni pemilihan elemen DOM, transferal dan manipulasi—dimungkinkan berkat adanya selector engine yang bernama Sizzle (sejak versi 1.3), yang membuat sebuah "gaya pemrograman baru", memadukan antara algoritma dan data struktur DOM. Gaya ini dipengaruhi oleh arsitektur JavaScript lainnya seperti YUI v3 dan Dojo, yang nantinya menstimulasi pembuatan standar Selector API.

### **2.9.2 AngularJS**

AngularJS adalah sebuah framework Javascript yang di kembangkan atau di bangun oleh tim Google. angularjs menggunakan metode MVC yang membuat source code aplikasi kita menjadi bersih dan mudah di kembangkan. sampai saat ini angularjs sudah menjadi salah satu framework javascript yang paling populer dan sangat banyak di gunakan oleh para developer di seluruh dunia untuk membangun aplikasi. kehandalan dari angularjs sendiri tidak bisa di ragukan lagi, salah satu faktor nya karena di kembangkan oleh para tim ahli dari google. angular JS juga menyediakan panduan cara penggunaannya pada situs resmi angularJS. untuk panduan penggunaan angularJS yang di sediakan oleh angularJS dapat di lihat di sini. dan angularJS juga di gunakan untuk membangun aplikasi



yang bersifat single page application. Tutorial AngularJS Pengertian Dan Cara Menggunakan AngularJS

Bagian – bagian utama AngularJS :

- **ng-app** yang berfungsi sebagai pen-definisi arahan dan link dari aplikasi Angular JS ke html.
- **ng-model** berfungsi untuk arahan data dari aplikasi Angular JS ke input kontrol html.
- **ng-bind** berfungsi sebagai arahan data dari aplikasi Angular JS ke tag html.

Beberapa fitur utama yang diunggulkan dalam framework AngularJS sebagai berikut.

**Fitur Utama AngularJs :**

1. JavaScript Framework Angular JS merupakan sebuah framework yang sangat efisien dalam dan mengembangkan Rich Internet Applications (RIA).
2. JavaScript Framework Angular JS merupakan aplikasi MVC (Model, View, Controller).
3. Dengan menggunakan Aplikasi yang berbasis Angular JS bisa diakses dari semua web browser.
4. Framework Angular JS merupakan framework JavaScript Open Source alias gratis, dan memiliki komunitas pengembang aplikasi yang banyak.

**Fitur Core Angular JS :**

1. Data binding yang berfungsi untuk sinkronisasi data antara Model dan View yang dimiliki Angular JS.
2. Scope adalah suatu objek yang mengacu pada sebuah model, sebagai perekat antara Controller dan View.
3. Controller Merupakan kumpulan dari fungsi-fungsi javascript yang dibuat dalam Angular JS.

4. Model View Whatever yang artinya Angular JS tidak mengimplementasikan MVC secara umum, melainkan sesuatu yang lebih dekat sebagai MVVM (Model View View Model).

#### **Keuntungan menggunakan Angular JS :**

Dengan menggunakan Angular JS dalam pembuatan aplikasi akan memudahkan developer, karena AngularJS sendiri memiliki kemampuan dalam membuat Single Page Application dengan cepat dan mudah.

Framework Angular JS mampu berinteraksi dengan HTML dan memiliki kemampuan data binding pada html dengan efesiensi yang tinggi.

Angular JS Menyediakan Dependency Injection (DI) dimana akan memungkinkan sang developer menulis beberapa komponen dan code yang terpisah satu sama lain. Kita hanya perlu memanggil komponen tersebut jika diperlukan dan bisa digunakan secara berulang.

Dengan menggunakan Angular JS sebagai pengembang aplikasi sang developer dapat membuat fungsi dan code yang lebih baik dan lebih ringkas.

## **2.10 Web Service**

Web Service merupakan suatu sistem yang menyediakan pelayanan yang dibutuhkan oleh klien. Klien dari web service tidak hanya berupa aplikasi web, tapi juga bisa sebuah aplikasi enterprise. Jadi web service tidak sama dengan web server, bahkan sebuah aplikasi web pada web server dapat menjadi klien dari web service. Pelayanan yang diberikan web service bisa berupa XML yang berisi data yang dibutuhkan klien. Web service tidak menampilkan sebuah halaman web, tapi hanya memberikan pelayanan permintaan klien yang memiliki izin akses terhadap pelayanan yang diminta.

## 2.11 Client Server

Client-server adalah suatu bentuk arsitektur, dimana client adalah perangkat yang menerima yang akan menampilkan dan menjalankan aplikasi (software komputer) dan server adalah perangkat yang menyediakan dan bertindak sebagai pengelola aplikasi, data dan keamanannya. Server biasanya terhubung dengan client melalui kabel UTP dan sebuah kartu jaringan (network card).

Kartu jaringan ini biasanya berupa kartu PCI atau ISA. Dalam teknologi informasi, client-server. Dalam model client-server, sebuah aplikasi dibagi menjadi dua bagian yang terpisah (tetapi masih dalam sebuah kesatuan) yakni komponen client dan komponen server. Komponen client dijalankan pada sebuah workstation memasukkan data dengan menggunakan teknologi pemrosesan tertentu, kemudian mengirimkan ke komponen server, umumnya berupa permintaan layanan tertentu yang dimiliki oleh server. Komponen Server akan menerima permintaan layanan tersebut dan langsung memprosesnya serta mengambilkan hasil pemrosesan kepada client. Client pun menerima informasi hasil pemrosesan data tadi dan menampilkannya kepada pemakai dengan menggunakan aplikasi yang digunakan oleh pemakai dengan menggunakan aplikasi yang digunakan oleh pemakai. Sebuah contoh dari aplikasi client-server sederhana adalah aplikasi web yang di desain dengan menggunakan Active Server Pages (ASP). Skrip ASP akan dijalankan di dalam web server (Apache atau internet Information Services), sementara skrip yang berjalan dipihak client akan dijalankan oleh web browser pada komputer client (workstation). Client-server merupakan penyelesaian masalah pada software yang menggunakan database sehingga setiap komputer tidak perlu diinstal database. Dengan metode client-server database dapat diinstal pada client. Komponen client juga sering disebut sebagai front-end, sementara komponen server disebut sebagai back-end. Client server diartikan sebagai hubungan sebuah computer dalam sebuah jaringan, dimana server sebagai

penyedia aplikasi, data dan informasi yang bisa diakses client. (Arlinda, 2005)

## 2.12 JSON

JSON (*JavaScript Object Notation*) adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (*generate*) oleh komputer. Format ini dibuat berdasarkan bagian dari [Bahasa Pemrograman JavaScript, Standar ECMA-262 Edisi ke-3 - Desember 1999](#). JSON merupakan format teks yang tidak bergantung pada bahasa pemrograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh programmer keluarga C termasuk C, C++, C#, Java, JavaScript, Perl, Python dll. Oleh karena sifat-sifat tersebut, menjadikan JSON ideal sebagai bahasa pertukaran-data.

JSON terbuat dari dua struktur:

- Kumpulan pasangan nama/nilai. Pada beberapa bahasa, hal ini dinyatakan sebagai objek (*object*), rekaman (*record*), struktur (*struct*), kamus (*dictionary*), tabel hash (*hash table*), daftar berkunci (*keyed list*), atau *associative array*.
- Daftar nilai terurutkan (*an ordered list of values*). Pada kebanyakan bahasa, hal ini dinyatakan sebagai larik (*array*), vektor (*vector*), daftar (*list*), atau urutan (*sequence*).

## BAB III

### ANALISA DAN PERANCANGAN SISTEM

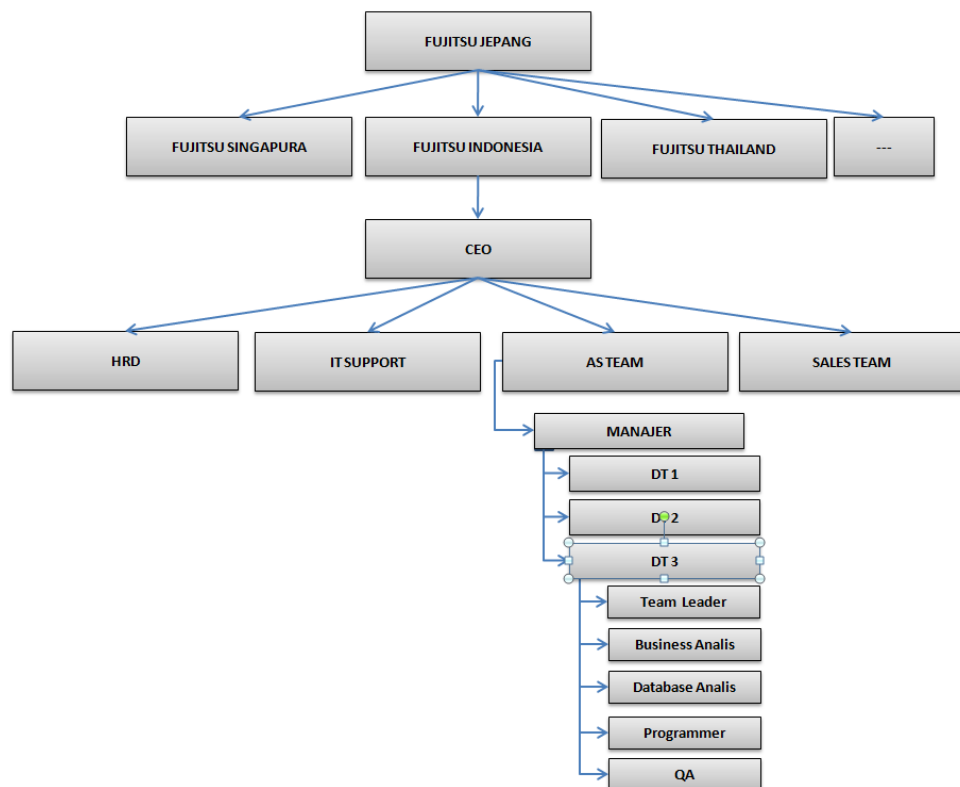
#### 3.1 Gambaran Umum Perusahaan

##### 3.1.1 Sejarah Singkat Perusahaan

PT. Fujitsu Indonesia adalah perusahaan yang bergerak dibidang informasi dan teknologi. Produk yang ditawarkan adalah berbagai produk teknologi, solusi dan layanan service. Berdiri sejak tahun 1995 dengan kantor pusat berada di Jakarta. Salah satu layanan yang diberikan PT. Fujitsu Indonesia adalah mengembangkan software untuk kebutuhan perusahaan tertentu.

##### 3.1.2 Struktur Organisasi

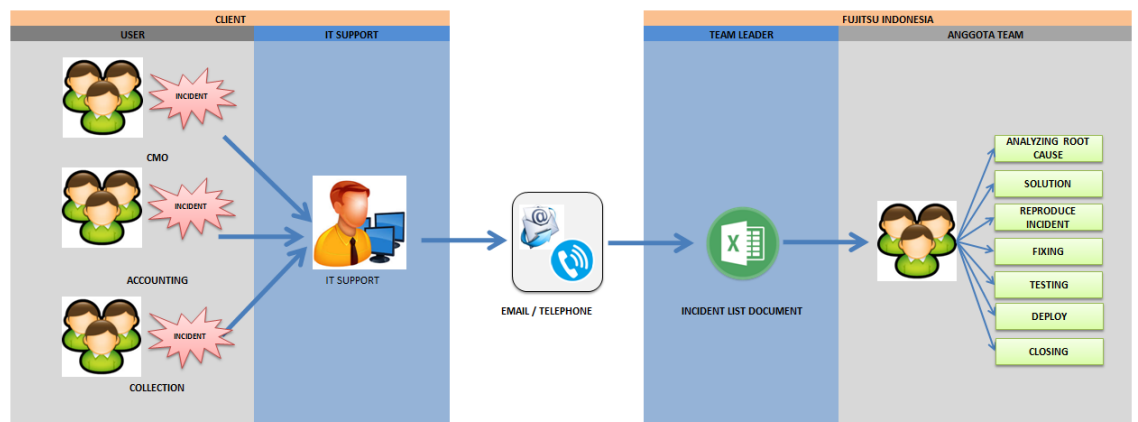
Struktur organisasi Fujitsu Indonesia adalah sebagai berikut



## 3.2 Analisa Sistem Berjalan

### 3.2.2 Flow Map Penanganan Insiden

Ketika end user menemukan insiden, maka user akan melaporkan insiden tersebut kepada teknisi IT di perusahaan mereka, kemudian meneruskan informasi mengenai insiden tersebut melalui email/telepon kepada tim Fujitsu. Tim fujitsu melalui Team Leader, menerima informasi tersebut dan memasukkannya kedalam dokumen insiden list. Dokumen tersebut di bagikan kepada anggota tim untuk di analisa root causenya, di cari solusi nya, di reproduce insidennya, di perbaiki, di testing dan kemudian di close.



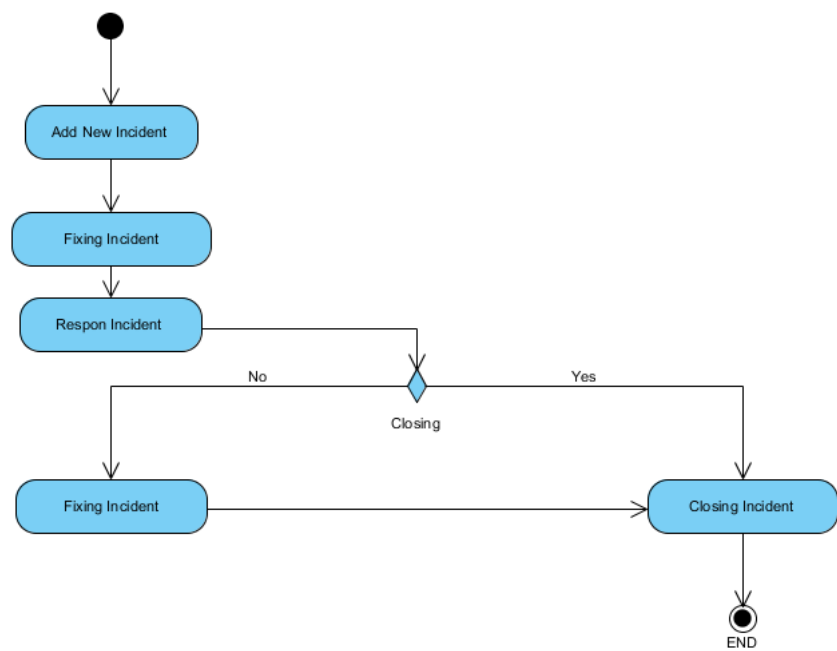
### 3.3 Analisa Aplikasi Manajemen Insiden

#### 3.3.1 Rancangan System Flow

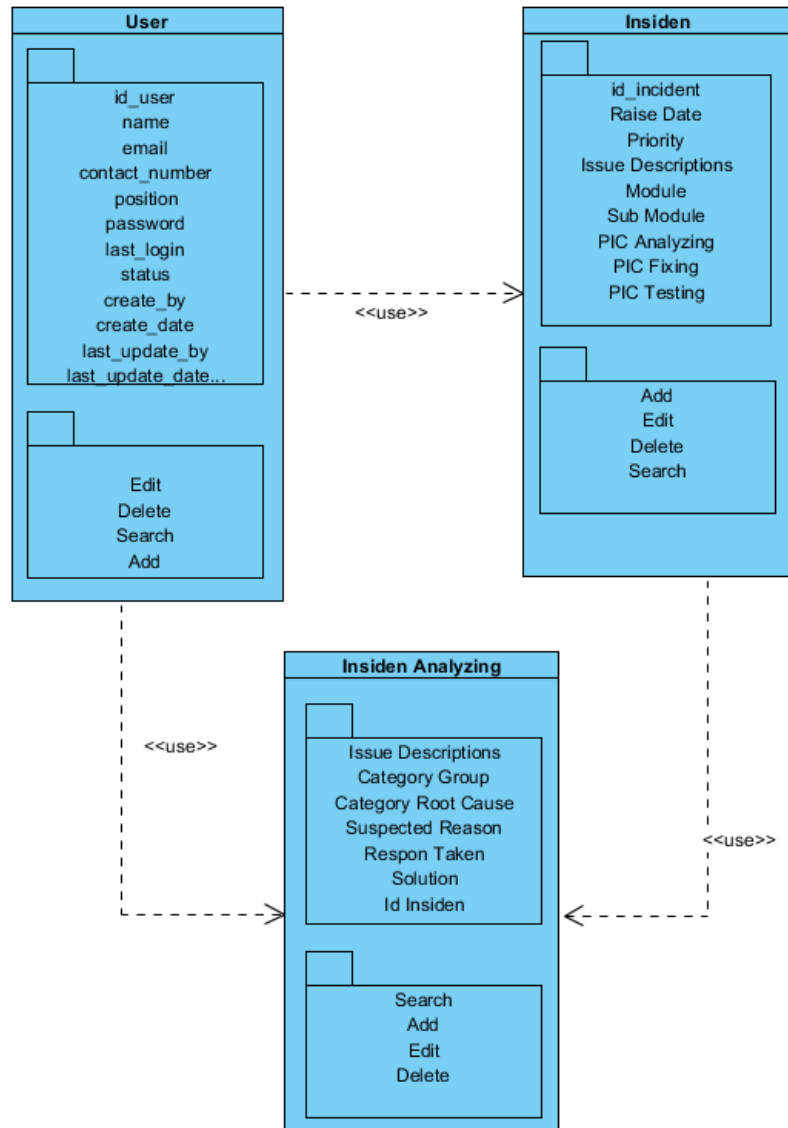
##### 3.3.3.1 Use Case Diagram



##### 3.3.3.2 Activity Diagram

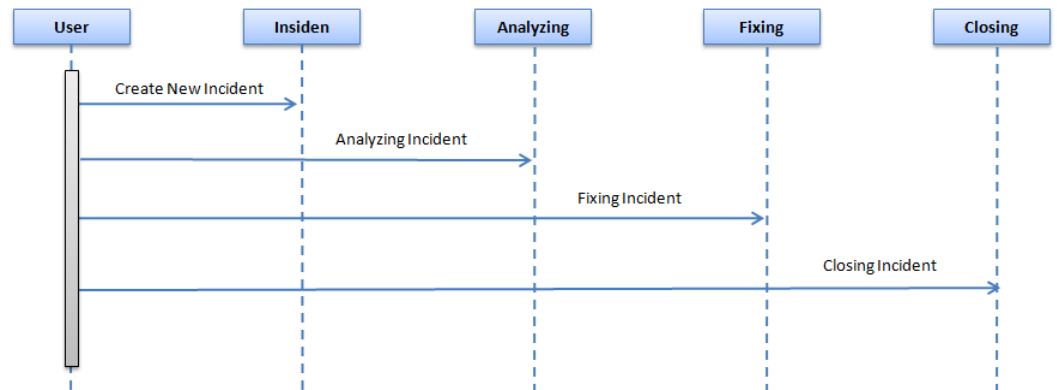


### 3.3.3.3 Class Diagram





### 3.3.3.4 Sequence Diagram



## 3.4 Rancangan Database

### 3.4.1 Struktur Tabel

#### 3.4.1.1 Tabel User

**Nama Tabel :** Tbl\_User  
**Isi :** Data User  
**Media :** Hard disk  
**Primary key:** Id\_User

Field	Type	Description
id_user (PK)	Varchar (50)	Id User
name	Varchar (250)	Nama User
email	Varchar (50)	Email
contact_number	Varchar (50)	Contact No
position	Varchar (50)	Jabatan, tugas dari user
password	Varchar (50)	password
last_login	Datetime	tanggal terakhir login
status	Varchar (50)	status aktif user
create_by	Varchar (50)	create user by
create_date	Datetime	tanggal create user
last_update_by	Varchar (50)	update user by
last_update_date	Datetime	tanggal update user

#### 3.4.1.2 Tabel Incident

**Nama Tabel :** Tbl\_Incident  
**Isi :** Data User

**Media :** Hard disk  
**Primary key:** Id\_Incident

Field	Type	Description
id_incident (PK)	Varchar (50)	Id insiden
raise_date	Datetime	tanggal terjadinya insiden
priority	Varchar (50)	level priority dari insiden
status	Varchar (50)	status closing insiden
module	Varchar (50)	module aplikasi
sub_module	Varchar (50)	sub modul aplikasi
pic	Varchar (50)	user yang menemukan insiden
category_group	Varchar (50)	category insiden
category_root_cause	Varchar (50)	category root cause
issue_description	Text	detail lengkap dari insiden
suspected_reason	Text	bagaimana program seharusnya berjalan
respon_taken	Text	respon yang diambil
decided_solution	Text	solusi dari insiden
target_fixed_date	Datetime	target fixing
finish_fixed_date	Datetime	target finish
closed_date	Datetime	closing insiden
closed_by	Varchar (50)	closed by
pic_analyzing	Varchar (50)	yang bertugas menganalisa insiden
finish_analyzing	Datetime	tanggal selesai analisis insiden
pic_fixing	Varchar (50)	yang bertugas memperbaiki insiden
finish_fixing	Datetime	tanggal selesai memperbaiki insiden
pic_testing	Varchar (50)	yang bertugas testing insiden
finish_testing	Datetime	tanggal selesai

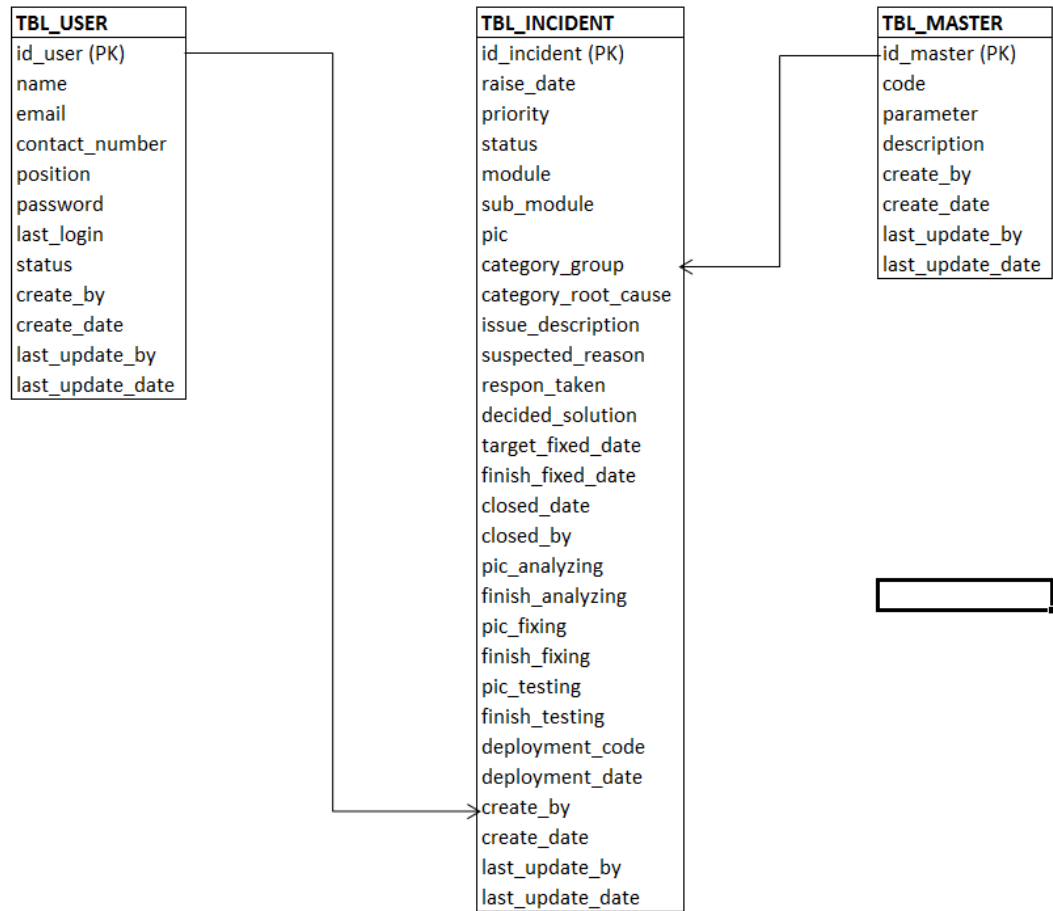
		testing insiden
deployment_code	Varchar (50)	code deploy
deployment_date	Datetime	tanggal deploy
create_by	Varchar (50)	create user by
create_date	Datetime	tanggal create user
last_update_by	Varchar (50)	update user by
last_update_date	Datetime	tanggal update user

#### 3.4.1.3 Tabel Master

**Nama Tabel :** Tbl\_Master  
**Isi :** Data Master  
**Media :** Hard disk  
**Primary key:** id\_master

Field	Type	Description
id_master (PK)	Integer	Id master data
code	Varchar (50)	code master data
parameter	Varchar (50)	parameter master data
description	Varchar (50)	description master data
create_by	Varchar (50)	create user by
create_date	Datetime	tanggal create user
last_update_by	Varchar (50)	update user by
last_update_date	Datetime	tanggal update user

#### 3.4.2 Relasi Database



### 3.5 Rancangan Tampilan

#### 3.5.1 Screen User

##### 3.5.1.1 Screen User List

User List					Add New User	
ID	Name	Email	Contact No	Position	Edit	Delete
1	Suwandi	<a href="mailto:suwandi@fujitsu.com">suwandi@fujitsu.com</a>	9701230	Manager	Edit	Delete
2	Faisal Amir	<a href="mailto:faisal@fujitsu.com">faisal@fujitsu.com</a>	8230982	Team Leader	Edit	Delete
	-----	-----	-----	-----	Edit	Delete
	-----	-----	-----	-----	Edit	Delete
	-----	-----	-----	-----	Edit	Delete
	-----	-----	-----	-----	Edit	Delete

##### 3.5.1.2 Screen Add User

### Add New User

X

Name

Email

Contact No

Position

Save

### 3.5.1.2 Screen Edit User

### Edit User

X

Name

Email

Contact No

Position

Save

## 3.5.2 Screen Pencatatan Insiden

### 3.5.2.1 Screen Incident List

Incident List					Add New Incident	
ID	Raise Date	Priority & Module	Issue Description	PIC	Edit	Delete
1	04/07/2017	Priority : High Module : Report Sub Module : Finance Report	1). Pretermination overdue days in printform is not correct. 2). Customer already paid for his 90 days overdue and remaining 25 days overdue (until 21 August 2015). 3). Pretermination printform shown that this customer has 115 days overdue (90+25) but the penalty amount is correct.	Analyzing : Nurochim Fixing : Nurochim Testing : Aulia		
2	11/07/2017	Priority : Medium Module : CDB Sub Module : Other	1). This customer have advance that fit for 1 installment payment. 2). Due date of this contract is 01 August 2015. 3). Fintegrity do the automated allocation for August Installment on 2 August 2015 ~ that's why this contract have 1 days overdue. 4). MF record that this customer have paid the same time with due date 01 August 2015. 5). Automatic allocation should be on 1st day of each month	Analyzing : Aulia Fixing : Nurochim Testing : Aulia	Edit	Delete
	-----	-----	-----	-----		
	-----	-----	-----	-----		
	-----	-----	-----	-----		

### 3.5.2.2 Screen Add New Incident

Add New Incident

X

Raise Date

Priority

Issue Descriptions

Module

Sub Module

PIC Analyzing

PIC Fixing

PIC Testing

Save

3.5.2.3 Screen Edit Incident

Edit Incident

x

Raise Date

Priority

Issue Descriptions

Module

Sub Module

PIC Analyzing

PIC Fixing

PIC Testing

Save

### 3.5.3 Screen Analisa Insiden

#### 3.5.3.1 Screen Incident List Analyzing

##### Incident List Analyzing

ID	Raise Date	Priority & Module	Issue Description	
1	04/07/2017	Priority : Medium Module : CDB Sub Module : Other Category Group : System Issue Root Cause : Business Case	<b>Issue Descriptions :</b> 1). This customer have advance that fit for 1 installment payment. 2). Due date of this contract is 01 August 2015. 3). Fintegrity do the automated allocation for August Installment on 2 August 2015 ~ that's why this contract have 1 days overdue. 4). MF record that this customer have paid the same time with due date 01 August 2015. 5). Automatic allocation should be on 1st day of each month <b>Suspected Reason :</b> User Input not match wit SOP <b>Respon Taken :</b> Prevent Code with More Validation <b>Decided solution :</b> Prevent Code with More Validation	<a href="#">Edit</a>
2	11/07/2017	Priority : Critical Module : General Ledger Sub Module : Payment Category Group : Root Cause :	<b>Issue Descriptions :</b> Sorting on Cash Payment Task has changed. Usually system shows sorting on Cash Transaction ID. <b>Suspected Reason :</b> <b>Respon Taken :</b> <b>Decided solution :</b>	<a href="#">Edit</a>
	*****	*****	*****	
	*****	*****	*****	
	*****	*****	*****	



3.5.3.2 Screen Edit Incident Analyzing

Incident Analyzing x

Issue Descriptions

Category Group

Category Root Cause

Suspected Reason

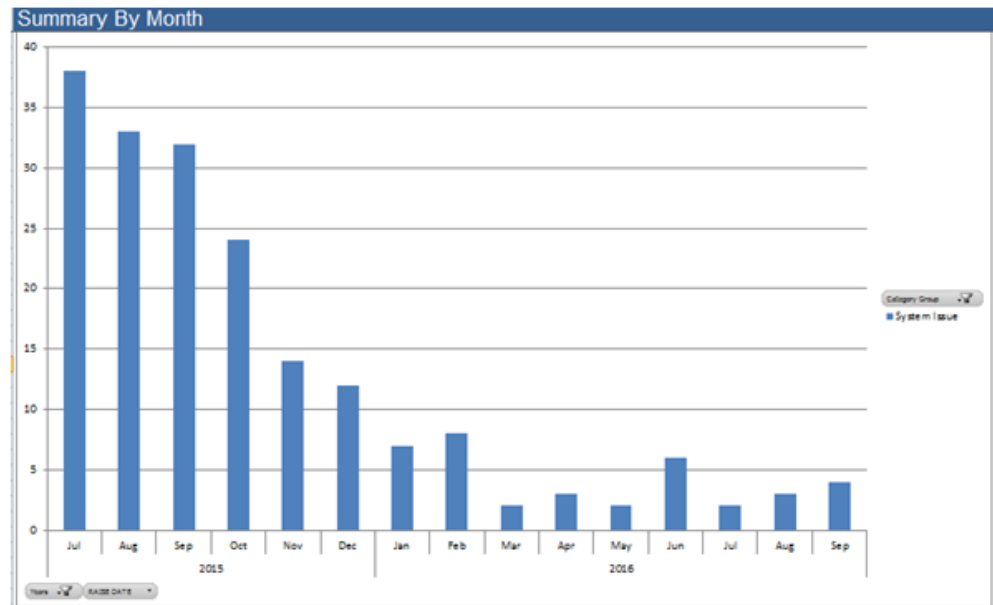
Respon Taken

Solution

Save

### 3.5.4 Screen Monitoring Insiden

## Incident Monitoring



## **BAB IV**

### **IMPLEMENTASI DAN PENGUJIAN**

#### **4.1 Implementasi**

Implementasi merupakan tahap pengembangan rancangan menjadi kode program. Pada awal bagian ini dijabarkan spesifikasi perangkat keras dan lunak pada mana program diimplementasikan. Bagian utama implementasi adalah penjabaran rancangan kelas menjadi kelas yang ditulis dalam sintaks HTML, PHP, AngularJS dan CSS.

#### **4.2 Batasan Implementasi**

Dalam implementasi perangkat lunak tentunya dibutuhkan dukungan perangkat lunak dan perangkat keras yang baik agar mendukung saat penelitian program atau coding. Perangkat yang penyusun gunakan pada saat melakukan implementasi adalah sebagai berikut

##### **4.2.1 Implementasi Perangkat Lunak**

Dalam menerapkan rancangan yang telah di buat, di butuhkan beberapa software untuk membuat program aplikasi manajemen insiden yaitu :

1. Sistem operasi yang digunakan yaitu Windows 7 (64 bit)
2. IDE yang digunakan untuk memudahkan dalam pengembangan aplikasi yaitu Netbean 7.1 yang mana IDE ini memiliki beberapa fasilitas yang diperlukan dalam pembangunan perangkat lunak.
3. Browser yang digunakan yaitu Mozilla Firefox, dengan fitur inspect elementnya yang sangat membantu peneliti dalam melakukan debugging saat terjadi error.
4. Laravel 5.4 sebagai framework untuk merancang web service pada aplikasi manajemen insiden.
5. HTML 5, AngularJS 2, Bootstrap digunakan peneliti untuk merancang front end dari aplikasi manajemen insiden.
6. MySQL digunakan peneliti sebagai databasenya.

## 4.2.2 Implementasi Perangkat Keras

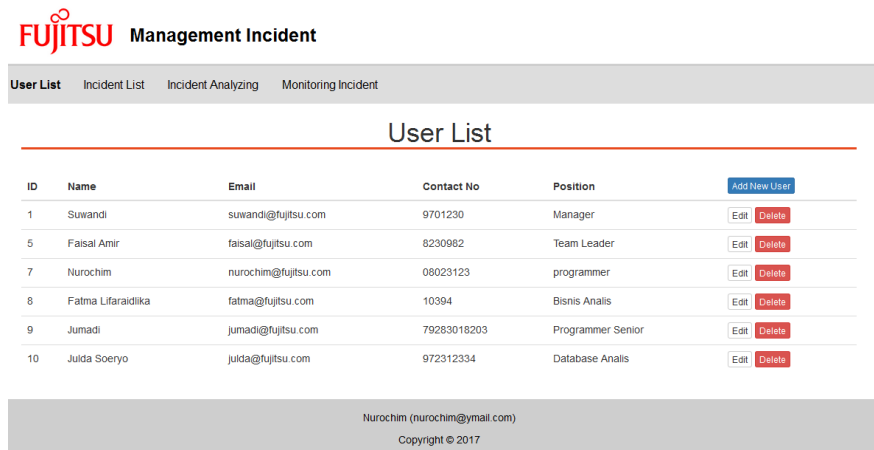
Dalam menerapkan dari rancangan yang telah di jelaskan sebelumnya di butuhkan beberapa perangkat keras untuk menyajikan aplikasi ini. Adapun alat-alat yang di butuhkan adalah :

1. Kebutuhan pengembang  
Satu unit PC dengan spesifikasi antara lain :
  - a. Processor : Intel Core i3
  - b. RAM : 8 GB
  - c. Hardisk : 500 GB
  - d. Browser : Mozilla Firefox, Google Chrome yang mendukung HTML 5
2. Kebutuhan minimal pemakai  
Komputer dengan spesifikasi apapun yang telah mendukung HTML 5

## 4.2.3 Implementasi Antar Muka

Implementasi merupakan tahapan yang bertujuan mengubah hasil dari rancangan sistem menjadi bentuk nyata, dalam hal ini berupa aplikasi manajemen insiden.

1. Tampilan User List



ID	Name	Email	Contact No	Position	
1	Suwandi	suwandi@fujitsu.com	9701230	Manager	<a href="#">Add New User</a>
5	Faisal Amir	faisal@fujitsu.com	8230982	Team Leader	<a href="#">Edit</a> <a href="#">Delete</a>
7	Nurochim	nurochim@fujitsu.com	08023123	programmer	<a href="#">Edit</a> <a href="#">Delete</a>
8	Fatma Lifaraidika	fatma@fujitsu.com	10394	Bisnis Analis	<a href="#">Edit</a> <a href="#">Delete</a>
9	Jumadi	jumadi@fujitsu.com	79283018203	Programmer Senior	<a href="#">Edit</a> <a href="#">Delete</a>
10	Julda Soeryo	julda@fujitsu.com	972312334	Database Analis	<a href="#">Edit</a> <a href="#">Delete</a>

Nurochim (nurochim@gmail.com)  
Copyright © 2017

2. Tampilan Add New User

### Add New User



Name

Fullname

Email

Email Address

Contact Number

Contact Number

Position

Position

Save changes

### 3. Tampilan Edit User

#### Employee Detail



Name

Suwandi

Email

suwandi@fujitsu.com

Contact Number

9701230

Position

Manager

Save changes

### 4. Tampilan Incident List

#### FUJITSU Management Incident

User List Incident List Incident Analyzing Monitoring Incident

#### Incident List

ID	Raise Date	Priority & Module	Issue Description	PIC	<a href="#">Add New Incident</a>
9	2017-07-04	Priority: High Module: Report Sub Module: Finance Report	1). Pretermination overdue days in printform is not correct. 2). Customer already paid for his 90 days overdue and remaining 25 days overdue (until 21 August 2015). 3). Pretermination printform shown that this customer has 115 days overdue (90+25) but the penalty amount is correct.	Analyzing: Nurochim Fixing: Nurochim Testing: Aulia	<a href="#">Edit</a> <a href="#">Delete</a>
11	2017-07-11	Priority: Medium Module: CDB Sub Module: Other	1). This customer have advance that fit for 1 installment payment. 2). Due date of this contract is 01 August 2015. 3). Fintegrity do the automated allocation for August Installment on 2 August 2015 - that's why this contract have 1 days overdue. 4). MF record that this customer have paid the same time with due date 01 August 2015. 5). Automatic allocation should be on 1st day of each month	Analyzing: Aulia Fixing: Nurochim Testing: Aulia	<a href="#">Edit</a> <a href="#">Delete</a>
12	2017-07-11	Priority: Critical Module: General Ledger Sub Module: Payment	Sorting on Cash Payment Task has changed. Usually system shows sorting on Cash Transaction ID.	Analyzing: Aulia Fixing: Faisai Testing: Aulia	<a href="#">Edit</a> <a href="#">Delete</a>

Nurochim (nurochim@gmail.com)

Copyright © 2017

## 5. Tampilan Add New Incident

Add New Incident

Raise Date

Priority

Issue Descriptions

Issue descriptions

Module

Sub Module

PIC Analyzing

PIC Fixing

PIC Testing

Save changes

## 6. Tampilan Edit Incident

Add New Incident

Raise Date

2017-07-04

Priority

High

Issue Descriptions

1). Pretermination overdue days in printform is not correct.  
2). Customer already paid for his 90 days overdue and remaining 25 days overdue (until 21 August 2015).  
3). Pretermination printform shown that this

Module

Report

Sub Module

Finance Report

PIC Analyzing

Nurochim

PIC Fixing

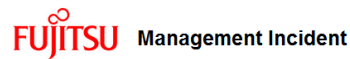
Nurochim

PIC Testing

Aulia Siahaan

Save changes

## 7. Tampilan Incident Analyzing



User List	Incident List	Incident Analyzing	Monitoring Incident
-----------	---------------	--------------------	---------------------

### Incident List Analyzing

ID	Raise Date	Priority & Modul	Issue Analyzing	Action
11	2017-07-11	Priority : Medium Module : CDB Sub Module : Other Category Group : System Issue Root Cause : Business Case	<b>Issue Descriptions :</b> 1). This customer have advance that fit for 1 installment payment. 2). Due date of this contract is 01 August 2015. 3). Fintegrity do the automated allocation for August installment on 2 August 2015 ~ that's why this contract have 1 days overdue. 4). MF record that this customer have paid the same time with due date 01 August 2015. 5). Automatic allocation should be on 1st day of each month <b>Suspected Reason :</b> User Input not match wit SOP <b>Respon Taken :</b> Prevent Code with More Validation <b>Decided solution :</b> Prevent Code with More Validation	<a href="#">Edit</a>
12	2017-07-11	Priority : Critical Module : General Ledger Sub Module : Payment Category Group : Root Cause :	<b>Issue Descriptions :</b> Sorting on Cash Payment Task has changed. Usually system shows sorting on Cash Transaction ID. <b>Suspected Reason :</b>  <b>Respon Taken :</b>  <b>Decided solution :</b>	<a href="#">Edit</a>

Nurochim (nurochim@gmail.com)  
Copyright © 2017

## 8. Tampilan Edit Incident Analyzing

### Incident Detail Analyzing

<b>Issue Descriptions</b>	<div>1). This customer have advance that fit for 1 installment payment.</div> <div>2). Due date of this contract is 01 August 2015.</div>
<b>Category Group</b>	<div>System Issue</div>
<b>Category Root Cause</b>	<div>Business Case</div>
<b>Suspected Reason</b>	<div>User Input not match wit SOP</div>
<b>Respon Taken</b>	<div>Prevent Code with More Validation</div>
<b>Solution</b>	<div>Prevent Code with More Validation</div>

Save changes

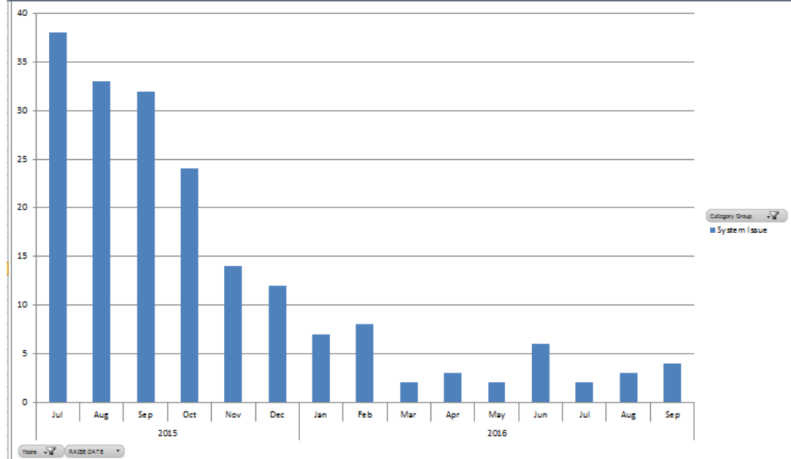
## 9. Tampilan Incident Monitoring

### FUJITSU Management Incident

User List Incident List Incident Analyzing Incident Monitoring

#### Incident Monitoring

Summary By Month



Nurochim (nurochim@gmail.com)

Copyright © 2017

## 4.2.4 Penggunaan Aplikasi

Berikut ini adalah cara penggunaan aplikasi manajemen insiden agar berjalan dan dapat di mainkan sebagai mana mestinya dengan penjelasan fungsi. Adapun cara penggunaanya sebagai berikut :

### 4.2.4.1 Masuk ke Aplikasi Manajemen Insiden

1. Buka browser kemudian masukkan URL dari aplikasi manajemen insiden
2. Tampilan utama saat masuk ke aplikasi adalah tampilan User List

### FUJITSU Management Incident

User List Incident List Incident Analyzing Monitoring Incident

#### User List

ID	Name	Email	Contact No	Position	<a href="#">Add New User</a>
1	Suwandi	suwandi@fujitsu.com	9701230	Manager	<a href="#">Edit</a> <a href="#">Delete</a>
5	Faisal Amir	faisal@fujitsu.com	8230982	Team Leader	<a href="#">Edit</a> <a href="#">Delete</a>
7	Nurochim	nurochim@fujitsu.com	08023123	programmer	<a href="#">Edit</a> <a href="#">Delete</a>
8	Fatma Lilaraidika	fatma@fujitsu.com	10394	Bisnis Analis	<a href="#">Edit</a> <a href="#">Delete</a>
9	Jumadi	jumadi@fujitsu.com	79283018203	Programmer Senior	<a href="#">Edit</a> <a href="#">Delete</a>
10	Julda Soeryo	julda@fujitsu.com	972312334	Database Analis	<a href="#">Edit</a> <a href="#">Delete</a>

Nurochim (nurochim@gmail.com)

Copyright © 2017



- Menu yang tersedia adalah Menu User List, Menu Incident List, Menu Incident Analyzing, Menu Incident Monitoring

#### 4.2.4.2 Input dan Edit Insiden

- Untuk dapat mencatat insiden baru, masuk ke menu “Incident List”

**FUJITSU** Management Incident

User List **Incident List** Incident Analyzing Monitoring Incident

### Incident List

ID	Raise Date	Priority & Module	Issue Description	PIC	
9	2017-07-04	Priority: High Module: Report Sub Module: Finance Report	1). Pretermination overdue days in printform is not correct. 2). Customer already paid for his 90 days overdue and remaining 25 days overdue (until 21 August 2015). 3). Pretermination printform shown that this customer has 115 days overdue (90+25) but the penalty amount is correct.	Analyzing: Nurochim Fixing: Nurochim Testing: Aulia	<a href="#">Add New Incident</a> <a href="#">Edit</a> <a href="#">Delete</a>
11	2017-07-11	Priority: Medium Module: CDB Sub Module: Other	1). This customer have advance that fit for 1 installment payment. 2). Due date of this contract is 01 August 2015. 3). Fintegrity do the automated allocation for August installment on 2 August 2015 – that's why this contract have 1 days overdue. 4). MP record that this customer have paid the same time with due date 01 August 2015. 5). Automatic allocation should be on 1st day of each month	Analyzing: Aulia Fixing: Nurochim Testing: Aulia	<a href="#">Edit</a> <a href="#">Delete</a>
12	2017-07-11	Priority: Critical Module: General Ledger Sub Module: Payment	Sorting on Cash Payment Task has changed. Usually system shows sorting on Cash Transaction ID.	Analyzing: Aulia Fixing: Faisal Testing: Aulia	<a href="#">Edit</a> <a href="#">Delete</a>

Nurochim (nurochim@gmail.com)  
Copyright © 2017

- Klik tombol “Add New Incident List” maka akan tampil form input insiden
- Isi Form Input Insiden, dan klik tombol “Save” untuk menyimpan data

**Add New Incident** ✕

**Raise Date**

**Priority**

**Issue Descriptions**

**Module**

**Sub Module**

**PIC Analyzing**

**PIC Fixing**

**PIC Testing**

[Save changes](#)

4. Untuk melakukan edit, klik tombol “Edit” maka akan tampil “Form Edit Incident”, kemudian edit sesuai kebutuhan dan klik tombol “Save” untuk menyimpan

Add New Incident

Raise Date

2017-07-04

Priority

High

Issue Descriptions

1). Pretermination overdue days in printform is not correct.  
2). Customer already paid for his 90 days overdue and remaining 25 days overdue (until 21 August 2015).  
3). Pretermination printform shown that this

Module

Report

Sub Module

Finance Report

PIC Analyzing

Nurochim

PIC Fixing

Nurochim

PIC Testing

Aulia Siahaan

Save changes

#### 4.2.4.3 Menganalisa Insiden

1. User yang menganalisa insiden bisa melakukan input data hasil analisisnya kedalam aplikasi melalui menu “Incident Analyzing”

#### FUJITSU Management Incident

User List Incident List **Incident Analyzing** Monitoring Incident

#### Incident List Analyzing

ID	Raise Date	Priority & Modul	Issue Analyzing	Action
11	2017-07-11	Priority : Medium Module : CDB Sub Module : Other Category Group : System Issue Root Cause : Business Case	<u>Issue Descriptions :</u> 1). This customer have advance that fit for 1 installment payment. 2). Due date of this contract is 01 August 2015. 3). Fintegrity do the automated allocation for August installment on 2 August 2015 – that's why this contract have 1 days overdue. 4). MF record that this customer have paid the same time with due date 01 August 2015. 5). Automatic allocation should be on 1st day of each month <u>Suspected Reason :</u> User Input not match wit SOP <u>Respon Taken :</u> Prevent Code with More Validation <u>Decided solution :</u> Prevent Code with More Validation	<a href="#">Edit</a>
12	2017-07-11	Priority : Critical Module : General Ledger Sub Module : Payment Category Group : Root Cause :	<u>Issue Descriptions :</u> Sorting on Cash Payment Task has changed. Usually system shows sorting on Cash Transaction ID. <u>Suspected Reason :</u> <u>Respon Taken :</u> <u>Decided solution :</u>	<a href="#">Edit</a>

Nurochim (nurochim@gmail.com)

Copyright © 2017

2. Klik tombol “Edit” maka akan muncul “Form Edit Incident Analyzing”
3. Isi form edit dan klik tombol “Save” untuk menyimpan

#### Incident Detail Analyzing

##### Issue Descriptions

1). This customer have advance that fit for 1 installment payment.  
2). Due date of this contract is 01 August 2015.

##### Category Group

System Issue

##### Category Root Cause

Business Case

##### Suspected Reason

User Input not match wit SOP

##### Respon Taken

Prevent Code with More Validation

##### Solution

Prevent Code with More Validation

Save changes

### 4.3 Pengujian

Pengujian merupakan bagian penting dalam siklus pengembangan perangkat lunak. Pengujian dilakukan untuk menjamin kualitas dan juga mengetahui kelemahan dari perangkat lunak. Tujuan dari pengujian perangkat lunak ini adalah untuk menjamin bahwa perangkat lunak yang dibangun memiliki kualitas dan dapat diandalkan. Pengujian perangkat lunak ini menggunakan metode pengujian Black Box. Pengujian Black Box Digunakan untuk menguji fungsi-fungsi khusus dari aplikasi perangkat lunak yang dirancang.

#### 4.3.1 Rencana Pengujian

Adapun rencana pengujian sistem yang akan di uji dengan teknik pengujian BlackBox akan penyusun kelompokkan dalam tabel di bawah ini. Berikut ini adalah hasil dari pengujian dari aplikasi manajemen insiden.

No	Komponen yang di uji	Skenario butir uji	Hasil pengujian
1	Screen User List	Memilih menu "User List"	Black Box
2	Input user baru	Klik tombol "Add New User" di Screen "User List"	Black Box
3	Edit User yang sudah ada	Klik tombol "Edit" di Screen "User List"	Black Box
4	Delete User	Klik tombol "Delete" di Screen "User List"	Black Box
5	Validasi Input User	Input field mandatory	Black Box
6	Screen Incident List	Memilih menu "Incident List"	Black Box
7	Input insiden baru	Klik tombol "Add New Incident" di Screen "Incident List"	Black Box
8	Edit Insiden yang sudah ada	Klik tombol "Edit" di Screen "Incident List"	Black Box
9	Delete insiden	Klik tombol "Delete" di Screen "Incident List"	Black Box
10	Validasi Input insiden	Input field mandatory, field "Issue Description"	Black Box
11	Screen Incident Analyzing	Memilih menu "Incident Analyzing"	Black Box
12	Edit hasil analisa insiden	Klik tombol "Edit" di Screen "Incident Analyzing"	Black Box

**Tabel 5.1** Rencana Pengujian

#### 4.3.2 Kasus dan Hasil Pengujian

Berikut ini adalah kasus untuk menguji perangkat lunak yang sudah dibangun menggunakan metode BlackBox berdasarkan Tabel 5.1. Untuk hasil pengujiannya tersedia pada lampiran.

No	Komponen yang di uji	Skenario butir uji	Hasil yang di harapkan	Hasil pengujian
1	Screen User List	Memilih menu "User List"	Muncul screen baru "User List"	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Tidak berhasil
2	Input user baru	Klik tombol "Add New User" di Screen "User List"	Muncul popup form input "Add New User"	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Tidak berhasil
3	Edit User yang sudah ada	Klik tombol "Edit" di Screen "User List"	Muncul popup form input "Edit User"	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Tidak berhasil
4	Delete User	Klik tombol "Delete" di Screen "User List"	Muncul popup konfirmasi untuk melakukan "Delete" data user	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Tidak berhasil
5	Validasi Input User	Input field mandatory	Tidak bisa save form input user, jika field mandatory tidak di isi	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Tidak berhasil
6	Screen Incident List	Memilih menu "Incident List"	Muncul screen baru "Incident List"	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Tidak berhasil
7	Input insiden baru	Klik tombol "Add New Incident" di Screen "Incident List"	Muncul popup form input "Add New Incident"	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Tidak berhasil
8	Edit Insiden yang sudah ada	Klik tombol "Edit" di Screen "Incident List"	Muncul popup form input "Edit Incident"	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Tidak berhasil
9	Delete insiden	Klik tombol "Delete" di Screen "Incident List"	Muncul popup konfirmasi untuk melakukan "Delete" data Insiden	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Tidak berhasil

10	Validasi Input insiden	Input field mandatory, field "Issue Description"	Tidak bisa save form input insiden, jika field mandatory tidak di isi	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Tidak berhasil
11	Screen Incident Analyzing	Memilih menu "Incident Analyzing"	Muncul screen baru "Incident Analyzing List"	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Tidak berhasil
12	Edit hasil analisa insiden	Klik tombol "Edit" di Screen "Incident Analyzing"	Muncul popup form input "Edit Incident" dari screen "Incident Analyzing List"	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Tidak berhasil

**Tabel 5.2** Kasus dan Hasil Uji

#### 4.3.3 Kesimpulan Hasil Pengujian

Berdasarkan hasil pengujian aplikasi manajemen insiden diatas maka dapat disimpulkan bahwa pembangunan aplikasi manajemen insiden pada pengembangan perangkat lunak dengan studi kasus pada PT. Fujitsu Indonesia ini sudah sesuai dengan fungsional dan memberikan hasil yang di harapkan.

## **BAB V**

### **KESIMPULAN DAN SARAN**

#### **5.1 Kesimpulan**

Dengan menganalisa dan merancang aplikasi sesuai dengan langkah-langkah dan prosedur yang sudah peneliti uraikan sebelumnya maka diperoleh hasil jadi berupa “Aplikasi Manajemen Insiden” yang bisa digunakan untuk mencatat insiden, membagi tugas kepada tim, mencatat akar masalah (*root cause*) dan solusi (*solution*) dari insiden.

#### **5.2 Saran**

Aplikasi ini belum lengkap dan sempurna, banyak hal yang perlu diperbaiki dan ditambahkan untuk pengembangan lebih lanjut dari aplikasi ini, antara lain:

1. Aplikasi ini sudah menerapkan Web Service sebagai back-end nya, sehingga kedepan bisa dikembangkan kearah aplikasi mobile berbasis android yang lebih memanjakan pengguna dalam melakukan manajemen insiden
2. Aplikasi manajemen insiden ini masih terbatas pada satu project kecil yang studi kasusnya di PT. Fujitsu Indonesia, perlu kiranya di uji di tempat lain dengan proyek pengembangan perangkat lunak yang beragam sehingga aplikasi ini menjadi lebih lengkap, fleksibel, dan mudah diterapkan di berbagai proyek.
3. Belum ada nya screen yang memantau alokasi penggunaan web service, menjadi kelemahan terbesar aplikasi ini. Screen ini berfungsi untuk memantau resource web service sehingga ketika terjadi ketika web service mengalami beban kerja yang berat bisa dilihat bagian mana yang menyebabkan aplikasi berjalan lama.

## DAFTAR PUSTAKA

Darmayuda, Ketut, 2009, *Pemrograman Aplikasi Database dengan Microsoft Visual Basic.Net 2008*, Informatika, Bandung.

Hartono, Jogiyanto, 1990, *Analisis & Desain Sistem Informasi: Pendekatan Terstruktur*, Andi Offset, Yogyakarta.

Herlambang, Soendoro, dan Haryanto Tanuwijaya, 2005, *Sistem Informasi: konsep, teknologi, dan manajemen*, Graha Ilmu, Yogyakarta.

Kendall, dan Kendall, 2003, *Analisis dan Perancangan Sistem Jilid 1*, Prenhallindo, Jakarta.

Kendall, dan Kendall, 2003, *Analisis dan Perancangan Sistem Jilid 2*, Prenhallindo, Jakarta.

Marlinda, Linda, S.Kom, 2004, *Sistem Basis Data, ANDI OFFSET*, Yogyakarta.

NN, 2009, *Shortcourse SQL Server 2008 Express*, Andi Offset, Yogyakarta.

Safaat H, Nazaruddin (2011). *ANDROID Pemograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android* : Informatika Bandung.

Safaat H, Nazaruddin (2013). *Aplikasi Berbasis Android : Berbagai Implementasi dan Pengembangan Aplikasi Mobile berbasis Android* : Informatika Bandung.

Arlinda. 2005. *Kamus Komputer*, Jakarta: RESTU AGUNG.

Rumbaugh, J., Jacobson, I., & Booch, G. (1999) *The Unified Modelling Language Reference Manual*, Addison-Wesley

Booch, G., Rumbaugh, J. & Jacobson, I. (1998) *The Unified Modelling Language User Guide*. Addison-Wesley.



<https://idcloudhost.com/pengertian-dan-keunggulan-framework-laravel/>

<http://www.devaradise.com/id/2013/08/mengenal-apa-itu-css-dan-bagaimana-menggunakannya.html>

<http://mengenalangularjs.blogspot.co.id/>

<https://id.scribd.com/doc/76449900/Pengertian-Insiden>