

PEMBANGUNAN SISTEM MANAJEMEN PROYEK BERBASIS WEB (STUDI KASUS :PT. MITRA INTI CEMERLANG)

by Muhammad Faris Auzan

Submission date: 17-Dec-2019 04:24PM (UTC+0700)

Submission ID: 1235934706

File name: turnitin.docx (3.33M)

Word count: 18742

Character count: 117949

BAB 1 PENDAHULUAN

1.1 Latar Belakang

9

Proyek adalah sebuah kegiatan yang bersifat sementara yang telah ditetapkan awal pekerjaannya dan waktu selesainya, untuk mencapai tujuan dan hasil yang spesifik dan unik, dan pada umumnya untuk menghasilkan sebuah perubahan yang bermanfaat atau yang mempunyai nilai tambah. Manajemen Proyek adalah penerapan pengetahuan, keterampilan, sarana dan teknik pada kegiatan proyek, suatu proses kegiatan untuk melakukan perencanaan, pengorganisasian, pengarahan dan pengendalian atas sumber daya organisasi yang dimiliki perusahaan untuk mencapai tujuan tertentu dalam waktu dan sumber daya tertentu pula. Manajemen proyek adalah usaha pada suatu kegiatan agar tujuan adanya kegiatan tersebut dapat tercapai secara efisien dan efektif (Azahra, 2015). Dalam melakukan manajemen proyek bertujuan untuk mencapai hasil yang efektif dan efisien. Efektif dalam hal ini adalah dimana hasil penggunaan sumber daya dan kegiatan sesuai dengan sasarnya yang meliputi kualitas, biaya, waktu dan lain-lainnya. Sedangkan efisien diartikan penggunaan sumber daya dan pemilihan bagian kegiatan secara tepat yang meliputi jumlah, jenis, saat penggunaan sumber lain dan lain-lain. Oleh sebab itu manajemen proyek pada suatu proyek konstruksi merupakan suatu hal yang tidak dapat diabaikan begitu saja. Karena tanpa manajemen, suatu proyek konstruksi akan sulit berjalan sesuai dengan harapan, baik berupa biaya, waktu maupun kualitas. Manajemen yang buruk akan berimbas pada terjadinya keterlambatan dalam menyelesaikan suatu proyek, adanya biaya tambahan yang dikeluarkan diluar dari anggaran yang telah disusun, kualitas konstruksi yang tidak sesuai dengan persyaratan yang ada, dan terhambatnya proses pembangunan.

50

PT. Mitra Inti Cemerlang (PT. MIC) adalah sebuah perusahaan yang bergerak dibidang konstruksi gedung tempat tinggal. Yang berlokasi di Taman Palem Lestari Blok L No. 39 RT 013 RW 008, Cengkareng Barat, Cengkareng, Jakarta Barat, DKI Jakarta. Perusahaan ini didirikan oleh Liem Erik WIjaya dan saat ini dipimpin oleh Hendra Wirijadinata. Perusahaan ini pernah mendapatkan penghargaan pada kategori "Health, Safety & Environment Awards" pada tahun 2016 yang diberikan oleh Ciputra Residences. Perusahaan ini sudah mengerjakan banyak proyek, dimana nilai kontrak yang dikerjakan berkisar di antara tiga puluh juta rupiah hingga tiga puluh dua miliar rupiah sejak tahun 2011 hingga tahun 2018. Dimana sebagian besar proyek yang sudah dikerjakan berada di daerah jabodetabek (PT. MIC, 2018).

PT. MIC biasa mengerjakan 5-7 proyek dalam waktu yang sama, membuat direktur perusahaan dan *project manager* sulit dalam memantau proses yang sedang berjalan terhadap suatu proyek. Saat ini untuk melaporkan hasil kerja/proses pekerjaan yang sedang berjalan dilakukan dengan cara menyerahkan foto proses kerja dan laporan pekerjaan diserahkan ke kantor ataupun dengan cara mengirimkan lewat *email*. Dari hasil laporan ini, akan diproses untuk membuat

progress kerja mingguan. Dimana *progress* kerja mingguan ini digunakan sebagai acuan dalam melakukan evaluasi kerja setiap minggunya.

Sebelum suatu perusahaan mendapatkan suatu proyek pembangunan, akan terjadi proses pelelangan tender. Dimana pihak pemilik proyek akan mengirimkan undangan tender ke beberapa perusahaan, yang nantinya setiap perusahaan akan mengirimkan penawaran harga yang berupa Rencana anggaran biaya (RAB) terhadap tender tersebut. Penyusunan Rencana anggaran biaya (RAB) yang buruk bisa berimbas pada penggunaan dana yang tidak tepat dan kacaunya proses pembangunan. Penyusunan RAB terdiri dari material yang digunakan, upah tenaga, volume dan harga satuan. Dalam menentukan harga material dan upah tenaga, seringkali terdapat perbedaan nilai harga dari beberapa orang yang mengerjakan penyusunan RAB. Sehingga akan menyebabkan perhitungan menjadi salah dan dapat merugikan pihak kontraktor. Karena dalam proses penyusunan RAB di PT. MIC dikerjakan oleh beberapa orang, sehingga dapat mempercepat pekerjaan dan selesai sebelum deadline penawaran tender. Karena pada proses ini, kontraktor hanya diberikan waktu yang sedikit untuk membuat suatu penawaran tender. Selain itu dalam menjalankan penggerjaan proyek, suatu perusahaan memiliki jangka waktu penggerjaan proyek yang harus dicapai. Jika perusahaan kontraktor tidak bisa mencapai target waktu penggerjaan, perusahaan akan mendapatkan sanksi. Dalam hal ini Project Manager menentukan target penggerjaan yang harus dicapai setiap minggunya yang mana disebut *time scheduling*, dan ini dijadikan sebagai acuan jika bobot penggerjaan tidak sesuai target yang diharapkan. Saat ini PT. MIC masih menggunakan cara print manual *time scheduling* lalu menempelkannya di kantor, sehingga untuk melihat target pekerjaan minggu depan karyawan harus melihat dulu ke kantor.

Berdasarkan pada uraian diatas, maka sistem manajemen proyek ini dibangun untuk mengatasi permasalahan tersebut. Dimana nantinya sistem ini digunakan sebagai alat bantu untuk mengelola manajemen terhadap suatu proyek. Sehingga dapat membantu dalam melakukan penyusunan RAB hingga dapat memantau proses penggerjaan proyek.

16

1.2 Rumusan masalah

Berikut rumusan masalah pada penelitian ini:

1. Bagaimana hasil analisis dan perancangan "Sistem Manajemen Proyek Berbasis Web (Studi Kasus : PT. MITRA INTI CEMERLANG)" ?
2. Bagaimana hasil implementasi "Sistem Manajemen Proyek Berbasis Web (Studi Kasus : PT. MITRA INTI CEMERLANG)" ?
3. Bagaimana hasil pengujian *white-box testing* dan *black-box testing* "Sistem Manajemen Proyek Berbasis Web (Studi Kasus : PT. MITRA INTI CEMERLANG)" ?

1

1.3 Tujuan

Tujuan dari penelitian ini yaitu untuk mengembangkan sistem manajemen proyek pada PT. Mitra Inti Cemerlang yang diharapkan dapat membantu karyawan yang terlibat di dalam proses pengerjaan proyek untuk melakukan pengerjaan proyek mulai dari pengerjaan tender hingga pembangunan suatu proyek selesai.

2

1.4 Manfaat

Manfaat dari penelitian ini adalah memberikan alat bantu untuk melakukan pengelolaan manajemen terhadap suatu proyek. Sehingga kedepannya permasalahan yang ada sebelumnya pada PT. Mitra Inti Cemerlang dapat teratasi.

7

1.5 Batasan masalah

Batasan masalah pada penelitian ini adalah:

1. Fokus pada pembangunan sistem manajemen proyek pada PT. Mitra Inti Cemerlang. Meliputi daftar harga material, daftar harga upah, rencana anggaran biaya, laporan lapangan, dan *time scheduling* proyek.
2. Implementasi menggunakan bahasa pemrograman PHP dan menggunakan kerangka kerja *Codeigniter*.

25

1.6 Sistematika pembahasan

8

Sistematika pembahasan penelitian ini disusun menjadi beberapa bab sebagai berikut:

BAB 1 PENDAHULUAN

Pada bab ini mendeskripsikan tentang latar belakang dari permasalahan yang diangkat, rumusan masalah, tujuan penelitian, batasan masalah, manfaat penelitian dan sistematika pembahasan.

BAB 2 LANDASAN KEPUSTAKAAN

2

Pada bab ini berisikan tentang kajian terhadap tentang teori, konsep, model, metode, atau sistem dari pustaka ilmiah, yang berkaitan dengan tema, masalah, atau pertanyaan penelitian.

14

BAB 3 METODOLOGI

Pada bab ini menjelaskan tentang tipe penelitian, strategi dan rancangan penelitian seperti metode metode yang digunakan selama penelitian, langkah langkah kerja yang dilakukan selama penelitian

BAB 4 ANALISIS KEBUTUHAN

Pada bab ini membahas tentang analisa kebutuhan dan perancangan sistem. Di dalam analisis kebutuhan meliputi kebutuhan fungsional dan

kebutuhan non fungsional. Terdapat pemodelan sistem, perancangan sistem dan perancangan antarmuka pengguna.

BAB 5 PERANCANGAN DAN IMPLEMENTASI

Pada bab ini membahas perancangan sistem dan diagram perancangan perangkat lunak. Perancangan akan menjelaskan mengenai proses perancangan yang dilakukan untuk memenuhi kebutuhan sistem yang telah didapat dari proses rekayasa kebutuhan. Dan pada implementasi melakukan implementasi dari proses perancangan yang dibuat.

BAB 6 PENGUJIAN

10

Bagian ini akan menguraikan hasil pengujian dan analisis dari sistem yang dibangun. Mulai dari pengujian unit, integrasi, validasi dan compatibility.

BAB 7 PENUTUP

Pada bab ini menjelaskan tentang kesimpulan dan saran. Kesimpulan ini diperoleh dari hasil dan pembahasan dalam pembangunan "Sistem manajemen proyek (studi kasus : PT. MITRA INTI CEMERLANG)". Serta saran ini nantinya bisa digunakan dalam pengembangan selanjutnya.

BAB 2 LANDASAN KEPUSTAKAAN

Pada bagian ini akan menjelaskan tentang penelitian sebelumnya yang telah dilakukan untuk memecahkan masalah. Bagian ini juga akan menjelaskan dasar teori yang digunakan pada penelitian ini.

2.1 Tinjauan Pustaka

Penelitian sebelumnya yang telah dilakukan oleh Alkautsar (2014) dengan menggunakan metode *Rational Unified Process* (RUP). Pada penelitian ini terdapat masalah dalam hal penghitungan biaya upah untuk para pekerja, biaya bahan baku, dan anggaran dalam melakukan suatu proyek pembangunan. Untuk mendapatkan kebutuhan sistem, peneliti tersebut menggunakan kerangka PIECES (*Performance, Information, Economics, Control, Efficiency, and Service*) untuk melakukan identifikasi masalah, menganalisa dan memecahkan permasalahan yang ada. Dan hasil dari penelitian ini memudahkan perusahaan dalam melakukan pengawasan terhadap bahan baku dan melakukan pengawasan perkembangan yang terjadi di lapangan.

Pada penelitian kedua yang dilakukan Setiawan (2017), peneliti tersebut menggunakan metode pengembangan *waterfall* dan menggunakan metode *conceptual framework*. Hasil dari penelitian ini adalah pemanfaatan sistem informasi pada manajemen proyek perlu dilakukan untuk memberikan berbagai kemudahan pada pelaksanaan proyek yang dijalankan. Persamaan dan perbedaan dari ketiga penelitian diatas dapat dilihat pada Tabel 2.1.

Tabel 2.1 Kajian Pustaka

No.	Judul Penelitian <small>(13)</small>	Persamaan	Perbedaan	
			Penelitian Terdahulu	Rencana Penelitian
1	Sistem Informasi Manajemen Proyek Pada PT. Anugrah Pertiwi Kontrindo Palembang (Alkautsar, Raudah dan Welda, 2014)	- Sistem memberikan informasi mengenai laporan lapangan - Sistem memberikan informasi data tender proyek	- Sistem tidak menyediakan fitur pengelolaan jadwal progres proyek - Sistem tidak menyediakan fitur analisa pekerjaan	- Sistem menyediakan fitur pengelolaan jadwal progres proyek - Sistem menyediakan fitur analisa pekerjaan
2	Perancangan Sistem Informasi Manajemen	- Sistem memberikan informasi	- Sistem tidak menyediakan fitur	- Sistem menyediakan fitur

	Proyek : Sistem Informasi Kontraktor (Setiawan, Qadafi, 2017)	mengenai laporan lapangan - Sistem memberikan informasi mengenai jadwal pekerjaan	pengelolaan jadwal progres proyek - Sistem tidak menyediakan fitur analisa pekerjaan	pengelolaan jadwal progres proyek - Sistem menyediakan fitur analisa pekerjaan
--	---	--	---	---

5

2.2 Manajemen Proyek

Tujuan utama dari manajemen proyek adalah agar proyek dapat dilaksanakan dengan efisien, tepat waktu, dan mencapai hasil yang diinginkan. Sering terjadi pada sebuah proyek yang berlarut pada pekerjaannya sehingga pada akhirnya harus mengalami penjadwalan ulang. Oleh karena itu, peran perencanaan dalam suatu proyek sangat penting, segala sesuatu harus dimulai dari rencana dan harus disepakati bersama antara para stakeholder yang terlibat pada proyek. Stakeholder yang dimaksud di dalam proyek adalah pemilik proyek (*project owner*), komite pengarah (*steering committee*), pengguna hasil proyek dan pelaksana proyek (Heryanto, 2015). Manajemen proyek meliputi beberapa proses seperti perencanaan kegiatan (*planning*), pengorganisasian, pelaksanaan dan pengawasan. Tujuan dari manajemen proyek adalah :

1. Proyek yang dikerjakan dapat terselesaikan tepat waktu
2. Tidak terdapat biaya tambahan diluar dari biaya yang sudah dianggarkan
3. Memenuhi persyaratan kualitas yang sesuai dengan prosedur
4. Proses kegiatan yang berjalan lancar sesuai dengan perencanaan

Proses perencanaan proyek dapat dibagi menjadi dua tahap, yaitu manajemen konsultan dan manajemen kontraktor. Perencanaan yang dilakukan oleh konsultan meliputi perencanaan fisik struktur secara terperinci hingga sampai perencanaan anggaran biaya dan durasi pekerjaan. Sedangkan perencanaan yang dilakukan oleh kontraktor meliputi perencanaan metode kontraktor, rencana anggaran dalam pelaksanaan, dan perencanaan administrasi lapangan maupun perusahaan. Metode manajemen proyek yang digunakan oleh pelaksana dari pihak kontraktor dan konsultan pada umumnya berpatokan pada laporan-laporan tertulis mengenai keadaan nyata di lapangan. Laporan-laporan tersebut dapat berupa laporan harian, mingguan, dll.

7

2.3 Rencana Anggaran Biaya (RAB)

Rencana anggaran biaya (RAB) adalah perhitungan biaya yang diperlukan untuk bahan, upah dan alat, ataupun biaya-biaya lain yang berhubungan dalam pelaksanaan pekerjaan tersebut (Joko, 2018). Cara penggerjaan RAB yaitu dengan

2
cara mempelajari terlebih dahulu gambar rencana detail, dokumen rencana kerja, dan syarat-syarat yang ada. Lalu buat uraian pekerjaan dan atau barang yang diperlukan untuk menyelesaikan pekerjaan. Hitung *volume* pekerjaan atau barang sesuai dengan uraian yang telah dibuat sebelumnya. Buat analisis harga satuan pekerjaan untuk setiap jenis pekerjaan yang telah diuraikan. Selanjutnya jumlahkan harga dari seluruh uraian pekerjaan dan tambahkan PPN pada jumlah tersebut sehingga didapatkan nilai akhir RAB. Adapun alur dalam penyusunan RAB yaitu :

1. Persiapan
Pada tahap ini menyiapkan peralatan kerja yang mendukung dalam melakukan survey ke lokasi lapangan.
2. Survey Lapangan
Pada tahap ini melakukan topografi, *quarry* (sumber material), dan memperkirakan mengenai tenaga kerja, material, alat dan upah yang dibutuhkan dalam penggerjaan.
3. Analisis Perancangan
Melakukan perhitungan *volume* dari dokumen tender yang telah didapatkan.
4. Penyusunan Rancangan Teknis
Melakukan spesifikasi teknis, mendata daftar harga satuan, perhitungan RAB, membuat jadwal pelaksanaan, legalitas, dan lain-lain.

2.4 Pengawasan Lapangan

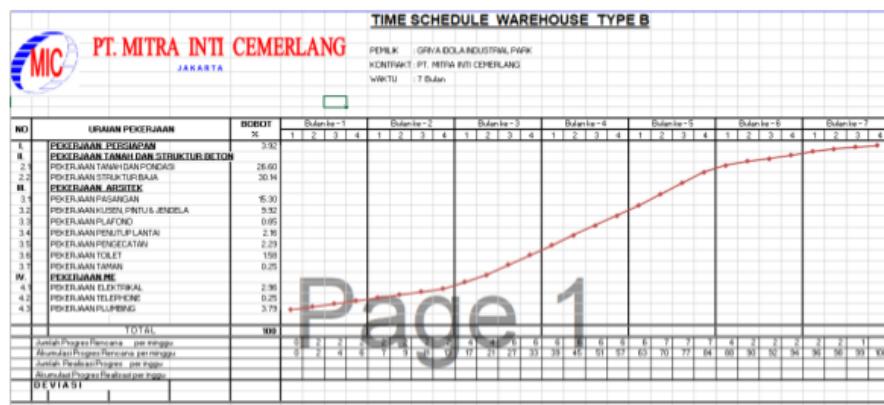
Pengawasan lapangan adalah suatu tindakan mendeteksi, mengawasi, membimbing dan mengarahkan kepada orang lain, kelompok lain maupun diri sendiri dengan tujuan agar rencana pekerjaan dapat dikerjakan dengan efisien dan memenuhi kualitas, kuantitas serta ketetapan waktu (PUPR, 2017). Pengawasan lapangan merupakan bagian dari kegiatan pengendalian kualitas pekerjaan, baik administrasi maupun teknis lapangan. Terdapat beberapa unsur untuk membantu dalam mencapai tujuan yaitu :

1. Pemeriksaan
Adalah suatu pengamatan yang pada umumnya dilakukan dari jarak dekat dengan jalan mengadakan perbandingan sesuatu yang telah atau akan dilakukan dengan yang seharusnya dilaksanakan menurut ukuran dan norma tertentu.
2. Pengawasan
Adalah suatu bentuk pengamatan yang pada umumnya dilakukan secara menyeluruh dengan jalan mengadakan perbandingan antara kenyataan yang dilakukan dengan yang seharusnya dilakukan atau yang seharusnya terjadi.
3. Pengendalian
Adalah tindakan pengaturan atau pengarahan pelaksanaan dengan maksud agar suatu tujuan tertentu dapat dicapai secara efisien dan efektif.

2.5 Time Schedule proyek

6
Secara umum dapat diartikan bahwa penjadwalan proyek merupakan sebuah jadwal proyeksi dari suatu proyek yang akan berfungsi sebagai pedoman utama

dalam pelaksanaan proyek. Penjadwalan proyek merupakan salah satu elemen hasil perencanaan, yang dapat memberikan informasi tentang jadwal rencana dan kemajuan proyek dalam hal kinerja sumber daya berupa biaya, tenaga kerja, peralatan, dan material serta rencana durasi proyek dan progres waktu untuk penyelesaian proyek. Dalam proses penjadwalan, penyusunan kegiatan dan hubungan antar kegiatan dibuat lebih terperinci dan sangat detail. Hal ini dimaksudkan untuk membantu pelaksanaan evaluasi proyek. Selama proses pengendalian proyek, penjadwalan mengikuti perkembangan proyek dengan berbagai permasalahannya. Proses monitoring yang berkala selalu dilakukan untuk mendapatkan penjadwalan yang paling realistik agar alokasi sumber daya dan penetapan durasinya sesuai dengan sasaran dan tujuan proyek (Situmorang, 2017). Gambar 2.1 adalah contoh dari *time schedule* proyek.



Gambar 2.1 Contoh Time Scheduling

19

2.6 Sistem Informasi

Sistem adalah elemen-elemen yang berinteraksi dalam mencapai tujuannya (Setiawan, 2006). Data yang diproses dan atau telah memiliki arti adalah informasi (Fatta, 2007). Sistem informasi adalah operasi organisasi manajerial dengan kegiatan strategi yang dipertemukan dengan pengolahan transaksi harian dengan tujuan menyediakan informasi yang diperlukan kepada pihak luar (Sutabri, 2012). Adapun konsep sistem informasi dapat dilihat pada gambar 2.2.



Gambar 2.2 Konsep Sistem Informasi

Sumber : Fatta (2007)

43
Penjelasan dari gambar 2.1 adalah sebagai berikut:

23

1. *Input data* adalah data yang masuk ke dalam sistem. Dimana di dalamnya termasuk metode-metode dan juga media yang digunakan untuk menangkap data yang akan dimasukkan. ¹¹
2. Pemrosesan adalah kegiatan untuk menerima *input data*, menyimpan dan mengakses data, menghasilkan sekaligus mengirimkan *output data* dan membantu pengendalian dari sistem secara keseluruhan.
3. *Output data* adalah sebuah produk dari hasil pemrosesan suatu data yang telah dimasukkan ke dalam sistem (Fatta, 2007).

1

2.7 Rekayasa Perangkat Lunak

Rekayasa perangkat lunak merupakan ilmu yang membahas aspek-aspek yang ada pada produksi perangkat lunak, mulai dari spesifikasi sistem hingga ke pemeliharaan sistem (Somerville, 2010). Sedangkan menurut Pressman (2010) di dalam rekayasa perangkat lunak terdapat suatu proses, suatu kumpulan metode-metode dan kumpulan perkakas yang memungkinkan para profesional melakukan pengembangan perangkat lunak komputer yang berkualitas tinggi.

2.7.1 Metode Pengembangan Perangkat Lunak

Perangkat lunak dikembangkan dengan sebuah metode pengembangan atau biasa disebut SDLC (*Software Development Life Cycle*). *Software development life cycle* merupakan suatu metode kegiatan untuk membangun sebuah perangkat lunak yang sistematis. Pengembangan dapat dilakukan mulai dari awal atau dapat melakukan penambahan dari sistem perangkat lunak yang telah ada sebelumnya (Somerville, 2010).

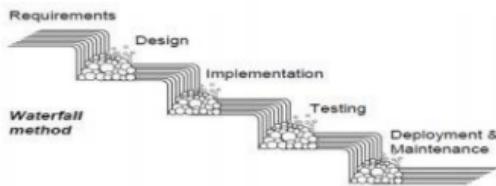
2

Ada beberapa jenis *software development life cycle* antara lain *waterfall model*, *prototyping model*, *rapid application diagram (RAD)*, *incremental model*, *spiral model*, dan lain – lain. Seiring berjalananya waktu semakin banyak *software development life cycle* lain hasil dari pengembangan yang sudah ada.

14

2.7.2 Waterfall Model

Waterfall model atau *Classic Life Cycle* adalah model klasik dari pengembangan perangkat lunak yang bersifat sistematis dan berurutan dalam membangun sebuah perangkat lunak (Pressman, 2010). Pengembangan perangkat lunak dilakukan tahap demi tahap, dimana setiap tahap pengembangan berlangsung berurutan tanpa ada tumpang tindih. Model pengembangan *waterfall model* dapat dilihat pada gambar 2.1 berikut.



Gambar 2.3 Waterfall Model

14

Penjelasan dari gambar 2.2 adalah sebagai berikut:

1. *Requirements*

Pada tahap ini diperlukan adanya komunikasi antara pengembang dan klien untuk memahami dan mendapatkan tujuan yang akan dicapai. Tahap ini merupakan inisiasi untuk mendapatkan kebutuhan-kebutuhan sistem yang diperlukan untuk membangun atau mengembangkan suatu sistem perangkat lunak.

2. *Design*

Setelah mendapatkan data atau informasi dari klien dari tahap sebelumnya. Pada tahap ini kita melakukan proses perencanaan dan pemecahan masalah untuk solusi perangkat lunak. Hal ini berkaitan dengan memilih desain algoritme yang sesuai, desain arsitektur perangkat lunak, database.

3. *Implementation*

Pada fase ini dilakukan pengkodean hasil rancangan untuk menjadi sistem perangkat lunak menggunakan bahasa pemrograman.

4. *Testing*

Pada fase ini dilakukan untuk menemukan kesalahan yang ada pada sistem. Selain itu pada tahap ini digunakan untuk melakukan pengecekan apakah sistem dapat berjalan sesuai rencana atau tidak. Dan juga dilakukan pengecekan apakah kebutuhan sistem sudah terpenuhi semua atau belum.

5. *Deployment and Maintenance*

Setelah perangkat lunak sudah dirilis, mungkin perlu beberapa modifikasi, perbaikan, koreksi kesalahan, dan penyempurnaan yang sesuai. Dimana pada tahap ini melakukan perbaikan error yang tidak ditemukan pada pengujian sebelumnya. Penelitian ini tidak melakukan tahapan deployment and maintenance karena pada penelitian ini hanya sampai pada tahap testing.

8

2.7.3 Object Oriented Analysis (OOA)

Object oriented analysis (OOA) adalah sebuah metode untuk memformulasikan model dari sistem perangkat lunak yang diinginkan dalam hal objek dan interaksinya (Stoecklin, 1988). OOA biasanya menghasilkan produk berupa UML diagram, yang dimana sekarang menjadi standar untuk pemodelan objek dan bahkan telah di mekanisasi dalam berbagai alat *computer-aided software engineering* (CASE). Keuntungan dari menggunakan UML diagram selama melakukan analisis sistem untuk membuat transisi yang mudah dan lancar untuk desain dan pengkodean (Sommerville, 2007).

2.7.4 Unified Modeling Language (UML)

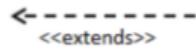
Unified modeling language (UML) adalah pemodelan perangkat lunak yang digunakan untuk pendekatan berorientasi objek. UML digunakan untuk memodelkan sebuah sistem perangkat lunak. Pemodelan digunakan untuk menyederhanakan permasalahan yang kompleks sehingga lebih mudah dipahami (Nugroho, 2009).

2.7.4.1 Use Case Diagram

Use case diagram adalah salah satu bagian dari UML yang merupakan bentuk pemodelan dari perilaku (*behavior*) dari sistem atau perangkat lunak yang akan dibuat. *Use case* menjelaskan interaksi antara satu atau lebih aktor dengan sistem atau perangkat lunak yang akan dibuat. *Use case* dipakai guna mengetahui fungsi-fungsi apa saja yang ada di dalam sebuah sistem atau perangkat lunak yang akan dibuat dan siapa saja yang mempunyai hak untuk menggunakan fungsi-fungsi tersebut (Sommerville, 2007). Di dalam *use case* ada dua hal utama yaitu apa yang disebut dengan aktor dan *use case*.

Aktor adalah orang, proses, atau sistem lain yang berinteraksi dengan sistem atau perangkat lunak yang akan dibuat di luar sistem atau perangkat lunak itu sendiri. Aktor digambarkan dengan simbol orang tetapi aktor belum tentu merupakan orang. *Use case* adalah bentuk fungsionalitas yang disediakan oleh sistem atau perangkat lunak. Pada Tabel 2.2 akan dijelaskan beberapa notasi yang ada pada *Use Case Diagram*.

Tabel 2.2 Notasi pada *Use Case Diagram*

Nama	Simbol	Deskripsi
<i>Use Case</i>		<i>Use Case</i> menggambarkan fungsionalitas yang terdapat di dalam sistem, yang dinamakan dengan kata kerja. Setiap <i>use case</i> harus terhubung dengan aktor, tetapi setiap <i>use case</i> tidak harus terhubung dengan aktor.
Asosiasi		Aktor di dalam sebuah <i>use case</i> ditunjukkan dengan aktor yang berhubungan dengan <i>use case</i> digambarkan dengan garis yang solid.
<i>Extend</i>		<i>Extend</i> merupakan sebuah relasi di dalam <i>use case</i> yang menggambarkan hubungan antara suatu <i>use case</i> dengan <i>use case</i> lain dimana <i>use case</i> tersebut merupakan fungsionalitas tambahan.

<i>Include</i>		<i>Include</i> merupakan sebuah relasi di dalam <i>use case</i> yang menggambarkan hubungan antara <i>use case</i> induk (<i>parent</i>) dengan <i>use case</i> lain (<i>child</i>) dimana jika terdapat pemanggilan fungsi pada <i>use case parent</i> , fungsi pada <i>use case child</i> harus dilaksanakan terlebih dahulu (<i>required</i>).
<i>Aktor</i>		<i>Aktor</i> merupakan representasi orang, proses, atau sistem lain yang berinteraksi dengan sistem.

1 2.7.4.2 Sequence Diagram

Sequence diagram merupakan diagram yang menyajikan interaksi antar objek yang digambarkan dalam grafik dua dimensi. Sumbu horizontal merepresentasikan objek/ individu yang berperan dalam proses interaksi. Sumbu vertikal merepresentasikan waktu, dimana dimensi waktu bergerak dari atas ke bawah. Garis waktu setiap objek digambarkan sebagai garis terputus – putus di bawah masing – masing objek. Proses interaksi/komunikasi antar objek dilakukan melalui pengiriman pesan (*message*) yang digambarkan sebagai sebuah garis panah lurus dari satu objek ke objek lainnya. Pada Tabel 2.3 akan dijelaskan beberapa notasi yang ada pada *Sequence Diagram*.

Tabel 2.3 Notasi pada Sequence Diagram

Nama	Simbol	Deskripsi
<i>Aktor</i>		<i>Aktor</i> berperan sebagai objek dan yang berinteraksi dengan sistem. Seorang aktor selalu berada di luar ruang lingkup sistem.
<i>Controller</i>		<i>Controller</i> adalah objek yang mengatur aliran interaksi antara <i>Boundary</i> dan <i>Entity</i> .
<i>Entity</i>		<i>Entity</i> merupakan objek yang merepresentasikan sistem data.
<i>Boundary</i>		<i>Boundary</i> merupakan objek yang berinteraksi dengan aktor.

<i>Message</i>	→	Komunikasi antar objek digambarkan menggunakan <i>message</i> .
<i>Return</i>	←	<i>Return</i> digunakan untuk menggambarkan pesan yang dikirim dari penerima ke pengirim.

12

2.7.4.3 Class Diagram

Class diagram adalah diagram yang digunakan untuk menampilkan beberapa kelas serta paket – paket yang ada dalam sistem/perangkat lunak yang sedang kita kembangkan. Diagram ini memberi gambaran atau *diagram statis* tentang sistem/perangkat lunak serta relasi – relasi yang ada di dalamnya. *Class diagram* merupakan himpunan dari objek-objek yang sejenis. Sebuah objek memiliki keadaan sesaat (*state*) dan perilaku (*behavior*). *State* sebuah objek adalah kondisi objek tersebut yang dinyatakan dalam *attribute/properties*. Sedangkan perilaku suatu objek mendefinisikan bagaimana sebuah objek bertindak/beraksi dan memberikan reaksi. Pada Tabel 2.4 akan dijelaskan beberapa notasi yang ada pada *Class Diagram*.

Tabel 2.4 Notasi pada *Class Diagram*

Nama	Simbol	Deskripsi
Asosiasi	_____	Asosiasi merupakan jenis hubungan yang mewakili hubungan statis antara kelas A dan kelas B
Generalisasi	→ ▷	Generalisasi membantu menghubungkan antara <i>subclass</i> dengan <i>superclass</i> nya. <i>Class Diagram</i> memungkinkan pewarisan dari beberapa <i>superclass</i> nya.
Agregasi	→ ◇	Agregasi adalah jenis asosiasi khusus yang memodelkan hubungan seluruh bagian antara agregat dan bagian-bagiannya.
Kelas	<div style="background-color: #00AEEF; color: white; padding: 5px; text-align: center;"> ClassName attributes operations </div>	Kelas menggambarkan model data untuk sistem informasi yang sangat kompleks. Ini memberikan gambaran tentang bagaimana sistem disusun sebelum mempelajari kode yang sebenarnya

11

2.7.5 Hypertext Preprocessor (PHP)

PHP (*Hypertext Preprocessor*) adalah sebuah bahasa pemrograman tingkat tinggi yang digunakan pada dokumen HTML (Lena, 2008). PHP digunakan secara luas untuk mengembangkan web. PHP difokuskan sebagai penulisan dari sisi *server side*.

39

2.7.6 HyperText Markup Language (HTML)

HTML merupakan sebuah bahasa pemrograman yang digunakan untuk menampilkan konten di web (Winarno, 2014). HTML digunakan untuk menampilkan informasi atau konten di dalam sebuah *web browser*. World Wide Web Consortium (W3C) menetapkan HTML sebagai standar internet.

14

2.7.7 Cascading Style Sheets (CSS)

CSS (*Cascading Style Sheets*) merupakan suatu standar pada dokumen struktur yang digunakan untuk pembuatan dan pemakaian style (Krause, 2016). Untuk mempersingkat penulisan *tag* yang terdapat di dalam HTML, maka digunakanlah CSS yang berfungsi untuk meringkas penulisan *tag* pada HTML sehingga tidak terjadi pengulangan penulisan.

1

2.7.8 JavaScript

Javascript merupakan bahasa pemrograman yang dipakai untuk membuat halaman web menjadi lebih interaktif dan dinamis (Prasetio, 2014). Javascript dapat langsung terintegrasi dengan HTML. Javascript pemrosesannya dilakukan di sisi klien, karena Javascript merupakan bahasa pemrograman yang bersifat *Client Side Programming*.

2.7.9 CodeIgniter

22

CodeIgniter merupakan kerangka kerja PHP yang kuat dengan *footprint* yang sangat kecil, dibangun untuk pengembang yang membutuhkan *toolkit* sederhana dan elegan untuk membuat aplikasi web berfitur lengkap (CodeIgniter, 2019). CodeIgniter memungkinkan pengembang untuk mengembangkan proyek lebih cepat dengan menyediakan serangkaian *library* yang biasanya dibutuhkan. CodeIgniter memungkinkan pengerjaan proyek dengan meminimalkan jumlah kode yang diperlukan.

2.7.10 Bootstrap

Bootstrap adalah *front-end component library* yang sangat populer. Karena dengan menggunakan bootstrap kita dapat membangun tampilan web menjadi responsif. Bootstrap merupakan sebuah alat *open source* untuk mengembangkan HTML, CSS, dan Javascript (Bootstrap, 2019).

2.7.11 DataTables

DataTables merupakan *plugin* untuk jQuery Javascript *library*. DataTables adalah alat yang sangat fleksibel, dibangun diatas dasar peningkatan progresif,

yang menambahkan semua fitur canggih ini ke tabel HTML apa pun. (DataTables Developer, 2019).

2.7.12 MySQL

11

SQL merupakan kepanjangan dari *structured query language*. MySQL adalah sebuah aplikasi sistem manajemen *database* SQL (DBMS) yang multi-user dan *multithread*. MySQL mengimplementasi sistem manajemen basis data relasional (RDBMS) (Lena, 2008). Mysql merupakan sebuah perangkat lunak *database* yang *open source* dan memiliki dua bentuk lisensi, yaitu *free software* dan *shareware* namun biasa kita menggunakan yang *free software*. Kita dapat menggunakan mysql dengan bebas baik itu untuk keperluan pribadi atau usaha tanpa harus membayar ataupun membeli lisensinya (Nugroho, 2005).

MySQL digunakan untuk menyimpan data yang diperlukan oleh aplikasi (Raharjo, 2016). Mysql adalah salah satu *database* yang paling populer di antara database yang lain. MySQL mampu menangani data dalam jumlah besar dan mampu mengirim dan menerima data dengan cepat (Komputer, 2010).

2.7.13 Pengujian Perangkat Lunak

Perangkat lunak membutuhkan perancangan kasus uji dengan tujuan untuk mendapatkan kesalahan dengan waktu yang relatif singkat dan usaha yang minimum. Ada banyak metode metode perancangan kasus uji yang telah berkembang guna menyediakan pendekatan pengujian secara sistematis bagi para pengembang. Bahkan metode metode perancangan kasus uji ini juga menyediakan mekanisme yang dapat membantu kelengkapan dari pengujian dan memberikan kemungkinan tertinggi untuk dapat menemukan berbagai kesalahan dalam sebuah sistem atau perangkat. Ada beberapa teknik pengujian yang biasa dipakai yaitu *white-box* dan *black-box*. *White-box* adalah teknik pengujian yang menganalisa logika dari kode program, sedangkan *black-box* mengarah pada segi fungsional dan sistem.

1. Pengujian Unit

Pengujian unit adalah pengujian pada perangkat lunak yang dilakukan pada masing-masing *unit* atau komponen perangkat lunak (SoftwareTestingFundamentals, 2019). Tujuannya adalah untuk memvalidasi apakah setiap unit perangkat lunak dapat berjalan sebagaimana dirancang. Unit adalah bagian terkecil yang dapat diuji dari semua perangkat lunak. Pada pengujian unit , pengujian perangkat lunak dilakukan di setiap unit yang ada pada sistem untuk dilakukan pengujian. Tujuan dari melakukan pengujian unit untuk memeriksa bagian kode di dalam sistem untuk memastikan tidak ada nya *bug* atau *error*.

2. Pengujian Integrasi

Pengujian integrasi adalah kelanjutan dari pengujian unit. Dalam bentuk yang sederhana, dua unit yang telah dilakukan pengujian digabungkan menjadi sebuah komponen dan *interface* dan hubungan keduanya

dilakukan pengujian (Hoobs, 2015). Pengujian integrasi dilakukan untuk menguji hubungan setiap kelas yang ada di dalam sistem.

1
3. Pengujian Validasi

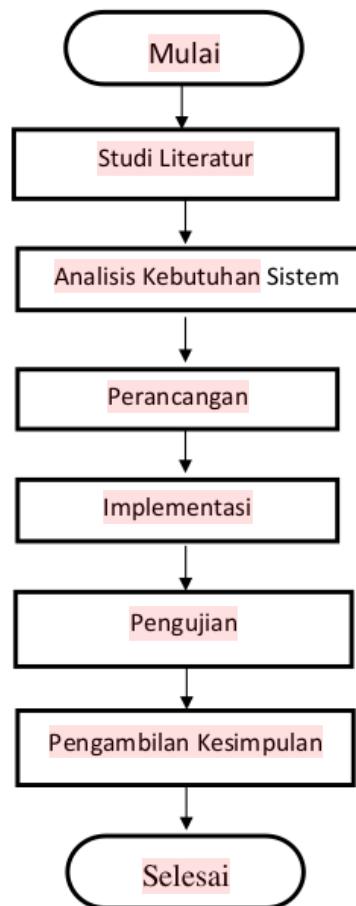
Pengujian validasi merupakan tahap dimana sistem diuji apakah semua fungsionalitas yang ada di dalam sistem sudah sesuai dengan *use case scenario*. Pengujian validasi dilakukan dengan menggunakan metode *black-box* dimana pengujian dilakukan hanya dengan melihat hasil keluaran tanpa menelusuri kode program. Validasi berarti model dapat diterima karena memenuhi persyaratan kinerja yang ditentukan sebelumnya (Rykiel, 1996).

4. Pengujian *Compatibility*

5
Untuk mengetahui apakah sistem yang telah kita buat dapat berjalan dengan baik di berbagai *browser* dan device maka perlu untuk melakukan pengujian *compatibility*.

BAB 3 METODOLOGI PENELITIAN

Ada beberapa tahap metodologi yang akan dilakukan di dalam penelitian ini diantaranya yaitu studi literatur, analisis kebutuhan sistem, perancangan dan implementasi sistem, pengujian, kesimpulan dan saran. Berikut ini merupakan diagram alir tahapan-tahapan metodologi penelitian yang dijelaskan pada gambar 3.1.



Gambar 3.1 Diagram alir metode penelitian

3.1 Studi Literatur

Studi literatur merupakan tahapan-tahapan yang dilakukan untuk mengumpulkan data atau referensi yang akan dijadikan sebagai acuan dalam menjelaskan dasar teori yang menunjang penelitian. Referensi yang digunakan memiliki bahasan yang berhubungan dengan pengembangan penelitian ini. Dasar teori pendukung tersebut meliputi:

1. Rekayasa Perangkat Lunak
2. Manajemen Proyek
3. Rencana Anggaran Biaya (RAB)
4. *Time Schedule* Proyek
5. *Pengawasan Lapangan*
6. *Waterfall Model*
7. *Unified Modeling Language*
8. Pengujian Perangkat Lunak

1

3.2 Analisis Kebutuhan Sistem

Analisis kebutuhan sistem digunakan untuk mendapatkan kebutuhan-kebutuhan apa saja yang diperlukan dalam penelitian ini. Analisis kebutuhan Sistem yang akan dibahas berupa kebutuhan fungsional dan kebutuhan non-fungsional. Dimana kebutuhan fungsional untuk menjelaskan proses – proses apa saja yang akan dilakukan oleh sistem. Sedangkan kebutuhan non-fungsional menjelaskan perilaku yang dimiliki oleh sistem. Hasil dari analisis kebutuhan sistem berupa elisitasi kebutuhan, gambaran umum sistem, identifikasi aktor, identifikasi kebutuhan dan pemodelan sistem.

3.2.1 Elitisasi Kebutuhan

Tahapan elitisasi kebutuhan adalah tahapan yang yang dilakukan untuk mengetahui informasi tentang kebutuhan-kebutuhan dari sistem perangkat lunak yang akan dibangun. Dalam tahapan elitisasi kebutuhan, penulis melakukan wawancara dengan karyawan PT. Mitra Inti Cemerlang. Dalam melakukan wawancara, penulis melakukan penggalian informasi lebih dalam terkait dengan kebutuhan-kebutuhan yang diperlukan dan diterapkan di sistem. Hasil yang nantinya didapatkan dalam tahapan analisis kebutuhan adalah penulis dapat mengetahui tentang gambaran umum sistem yang akan dibangun, dapat mengetahui siapa saja aktor yang terlibat dan dapat mengetahui apa saja kebutuhan dari sistem yang akan dibangun.

7

3.2.2 Gambaran Umum Sistem

8

Gambaran umum sistem adalah tahapan yang dilakukan untuk dapat mengetahui gambaran umum tentang sistem yang akan dibangun. Dalam melakukan tahapan ini, penulis melakukan analisis terkait dengan hasil wawancara yang dilakukan selama proses tahapan elitisasi kebutuhan. Hasil yang didapatkan dalam tahapan ini adalah penulis dapat memberikan penjelasan tentang gambaran umum sistem dan penjelasan fitur-fitur utama yang dapat dilakukan oleh sistem.

3.2.3 Identifikasi Aktor

44

Identifikasi aktor adalah tahapan yang dilakukan untuk mengetahui siapa saja aktor yang terlibat dalam penggunaan sistem beserta tugas masing-masing. Dalam melakukan identifikasi aktor, penulis melakukan analisis berdasarkan hasil gambaran umum sistem yang telah didapatkan sebelumnya. Penulis melihat siapa siapa saja aktor yang terlibat. Penulis memperhatikan apa-apa saja tugas dari masing masing aktor yang terlibat. Setelah melakukan hal tersebut, hasil yang didapatkan adalah penulis dapat merumuskan nama-nama aktor yang terlibat di sistem beserta pembagian tugas masing-masing.

3.2.4 Identifikasi Kebutuhan

1

Identifikasi kebutuhan adalah tahapan yang dilakukan untuk mengetahui apa saja kebutuhan yang harus disediakan oleh sistem. Dalam melakukan tahapan ini, penulis melakukan analisis setiap informasi yang didapatkan dalam proses elisitasi kebutuhan. Penulis dapat mendapatkan kebutuhan dengan memperhatikan setiap detail informasi yang didapatkan dalam proses elisitasi kebutuhan. Kebutuhan sistem dibagi menjadi 2 jenis, kebutuhan fungsional dan kebutuhan non fungsional. Peneliti harus memperhatikan daftar kebutuhan dari hasil analisis ini harus dapat dilakukan, diukur, diuji, terkait dengan proses bisnis yang teridentifikasi, serta didefinisikan sampai tingkat detail yang memadai untuk desain sistem. Fitur-fitur yang telah diidentifikasi kemudian dispesifikasikan kedalam kalimat kebutuhan. Hasil yang didapatkan dalam proses ini adalah penulis mendapatkan kalimat kebutuhan dan spesifikasi kebutuhan beserta aktor.

3.2.5 Pemodelan Kebutuhan

Pemodelan kebutuhan adalah tahapan yang dilakukan untuk memberikan penjelasan tentang bagaimana hubungan interaksi antara aktor dengan masing-masing kebutuhan yang ada di sistem. Dalam melakukan pemodelan kebutuhan, penulis dapat memodelkan kebutuhan dengan cara memperhatikan kembali hasil yang didapatkan dalam kalimat atau spesifikasi kebutuhan sistem. Penulis kemudian dapat memodelkan setiap kebutuhan yang telah didapatkan dan siapa aktor yang menjalankannya. Pemodelan setiap kebutuhan dengan aktor yang terlibat kemudian dapat digambarkan ke dalam sebuah *use case diagram*. Setelah mendapatkan *model use case diagram*, penulis kemudian dapat menjelaskan detail urutan langkah-langkah dalam mencapai tujuan dari sebuah *use case*. Langkah-langkah tersebut kemudian digambarkan ke dalam sebuah *use case scenario*. Hasil yang didapatkan dalam pemodelan kebutuhan nantinya adalah berupa *model* kebutuhan yang dituangkan ke dalam *use case diagram* dan *use case scenario*.

3.3 Perancangan

Perancangan sistem dilakukan berguna untuk mempermudah dalam proses melakukan penelitian. Perancangan sistem dimulai dari perancangan *use case diagram* dan *use case scenario*, perancangan *sequence diagram*, dan perancangan

1
class diagram. Perancangan sistem dibutuhkan dengan tujuan untuk menjelaskan bagaimana langkah-langkah kerja dari sistem secara keseluruhan dan untuk mempermudah dalam implementasi maupun pengujian. Hasil dari perancangan adalah perancangan arsitektur, komponen dan antarmuka.

1 3.3.1 Perancangan Arsitektur

Pada perancangan arsitektur melakukan pemodelan *sequence diagram* dan *class diagram*. Dalam melakukan pembuatan *sequence diagram*, diperlukan memperhatikan hasil dari *use case diagram* dan *use case scenario* yang telah dibuat sebelumnya. Pada *sequence diagram* akan menggambarkan interaksi antara satu objek dengan objek yang lainnya. Dan *class diagram* menggambarkan kelas-kelas yang digunakan di dalam sistem.

1 3.3.2 Perancangan Komponen

Pada perancangan komponen adalah tahapan yang **1** dilakukan untuk membuat rancangan komponen dari sistem yang akan dibuat. Pada perancangan komponen akan memuat tiga sampel algoritme yang ada pada sistem yang dibangun. Algoritme tersebut akan ditulis dalam bentuk *pseudocode*.

1 3.3.3 Perancangan Antarmuka

Perancangan antarmuka adalah tahapan yang dilakukan untuk membuat rancangan tampilan antarmuka sistem yang akan dibangun. Pada perancangan antarmuka akan menggambarkan halaman-halaman pada sistem yang akan dibangun mengenai tata letak komponen yang ada di dalam sistem. Perancangan antarmuka ini akan digunakan sebagai acuan dalam melakukan implementasi antarmuka sistem.

3.4 Implementasi

Dalam tahap implementasi merupakan mengubah bentuk perancangan kedalam bentuk kode atau ²⁴ bahasa pemrograman yang digunakan di dalam penelitian. Implementasi pada sistem ini menggunakan bahasa pemrograman **PHP** dan menggunakan *framework* Codeigniter. Dan untuk implementasi pada antarmuka sistem menggunakan **HTML, CSS, Javascript** dan **Bootstrap**. Sedangkan pada basis data menggunakan MySQL.

1 3.5 Pengujian

Pengujian digunakan untuk memastikan bahwa penelitian yang dilakukan berjalan sesuai perancangan sistem dan memenuhi kebutuhan. Pengujian yang digunakan pada sistem ini adalah pengujian *black box*. Pada pengujian *black box* digunakan metode pengujian validasi bertujuan untuk mengetahui apakah sistem yang dibangun sesuai dengan kebutuhan perangkat lunak yang sudah ditentukan di awal. Dalam penelitian ini menggunakan pengujian unit, pengujian integrasi dan pengujian validasi.

3.5.1 Pengujian Unit

Pengujian unit dilakukan untuk menguji kelas-kelas yang telah didefinisikan sebelumnya. Pengujian ini dilakukan menggunakan teknik pengujian *white box* dengan menggunakan metode *basis path testing*. Kode program yang ada di dalam sistem akan diuji berdasarkan algoritme yang ada pada setiap kelas yang ada.

1 3.5.2 Pengujian Integrasi

Pengujian integrasi dilakukan untuk menguji hubungan setiap kelas yang ada di dalam sistem. Pengujian ini dilakukan dengan teknik pengujian *white box* menggunakan metode *basis path testing*.

3.5.3 Pengujian Validasi

Pengujian validasi dilakukan untuk menguji apakah semua kebutuhan fungsional sistem dapat berjalan dengan baik. Pengujian ini bertujuan untuk mengetahui apakah semua kebutuhan fungsional yang ada telah berjalan sesuai dengan tujuannya. Hasil akhir dari pengujian ini adalah untuk mengetahui apakah setiap kebutuhan fungsional sistem yang ada sudah valid.

3.5.4 Pengujian Compability

5
Untuk mengetahui apakah sistem yang telah kita buat dapat berjalan dengan baik di berbagai *browser* dan device maka perlu untuk melakukan pengujian *compatibility*. Pengujian ini dilakukan dengan bantuan aplikasi sortsite, dimana kita dapat menjalankan sistem kita pada aplikasi tersebut

1 3.6 Pengambilan Kesimpulan

Kesimpulan dan saran dilakukan setelah semua tahap selesai dilakukan dan menganalisa secara keseluruhan. Tahapan yang dilakukan yaitu meliputi perancangan, implementasi, dan pengujian sistem. Tahap kesimpulan dilakukan untuk menjawab rumusan masalah yang dituliskan sebelumnya. Sehingga diperoleh saran dari kesimpulan tersebut untuk menyempurnakan penelitian yang sedang dilakukan serta menjelaskan kekurangan yang ada pada penelitian ini agar dapat dikembangkan lagi di penelitian selanjutnya.

BAB 4 REKAYASA KEBUTUHAN

Rekayasa kebutuhan merupakan tahapan pertama yang dilakukan pada pengembangan sebuah sistem. Tahap ini dilakukan untuk mencari tahu dan menentukan kebutuhan yang akan dibuat di dalam sistem. Dalam menggali kebutuhan dilakukan wawancara dengan karyawan pada PT. Matahari Indah Cemerlang (MIC) yang nantinya akan menggunakan sistem. Pada tahap ini dibuat diagram *Use Case* yang bertujuan untuk menjelaskan interaksi yang dapat dilakukan oleh aktor yang ada di dalam sistem, diagram *Use Case* berguna untuk memberi penjelasan kepada aktor mengenai sistem yang akan dibuat. Selain itu juga terdapat skenario *Use Case* yang berguna untuk menjelaskan secara lebih rinci suatu *Use Case* yaitu tahapan apa saja yang harus dilakukan seorang aktor dalam melakukan sebuah aksi.

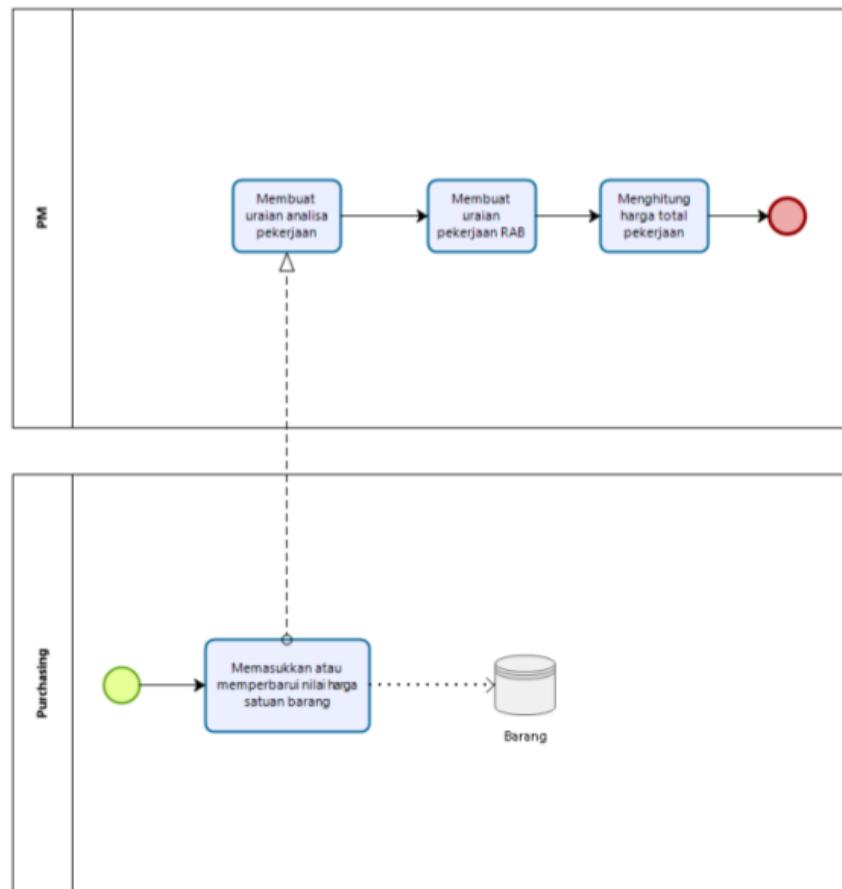
4.1 Elisitasi Kebutuhan

Hasil dari proses penggalian kebutuhan akan menjadi dasar pada identifikasi masalah. Pada tahap ini dijelaskan letak permasalahan yang dihadapi saat melakukan manajemen proyek secara konvensional. Hasil dari elisitasi kebutuhan meliputi 1) Pembuatan rencana anggaran biaya (RAB), 2) Pembuatan *time scheduling*, dan 3) *Monitoring* lapangan.

4.1.1 Pembuatan Rencana Anggaran Biaya (RAB)

Pembuatan rencana anggaran biaya (RAB) dilakukan pada saat proses pelelangan tender. Pada tahap RAB perusahaan-perusahaan yang mengikuti proses lelang akan membuat penawaran harga kepada pemilik proyek. Dimana penawaran harga dibuat berdasarkan penghitungan pekerjaan yang telah ditentukan pada dokumen tender. RAB dikerjakan oleh *Project Manager* (PM), dimana PM melakukan perancangan BoQ (*Bill of Quantities*) untuk menghitung *volume*, analisa harga satuan (terdiri dari analisa bahan dan upah kerja), jasa kontraktor, & estimasi *overhead cost*. Pada proses pembuatan RAB, *Purchasing* akan melakukan pendataan nilai harga satuan barang dan alat yang akan digunakan pada pekerjaan proyek. Setelah data harga sudah didapatkan maka PM akan membuat uraian analisa pekerjaan yang didapatkan dari harga barang, harga alat, harga upah pekerja dan menambahkan nilai jasa sebesar 10% dari harga total uraian analisa pekerjaan. Setelah analisa telah selesai dibuat maka proses selanjutnya membuat uraian pekerjaan pada RAB, dimana uraian pekerjaan pada RAB diambil dari analisa pekerjaan yang telah dibuat sebelumnya. Setelah RAB telah dibuat maka akan dilakukan penjumlahan harga total pekerjaan ditambah dengan PPN 10%. Proses pembuatan RAB akan dijelaskan pada gambar 4.1. Pada proses diatas terdapat beberapa kendala, yaitu saat mulai melakukan pengeringan RAB PM harus meminta daftar harga barang dari *purchasing* karena PM tidak memiliki daftar harga material. Dan *project manager* (PM) sulit untuk mengetahui sudah sejauh mana pekerjaan RAB yang telah dilakukan, karena biasanya rentang waktu pengeringan RAB pada PT. Mitra Inti Cemerlang (PT. MIC)

sangat terbatas, dan sebelum RAB dikumpulkan perlu dilakukan pengecekan akhir oleh PM untuk memastikan tidak ada kesalahan atau kekurangan dalam memasukkan harga di RAB.



Gambar 4.1 Proses Pembuatan Rencana Anggaran Biaya

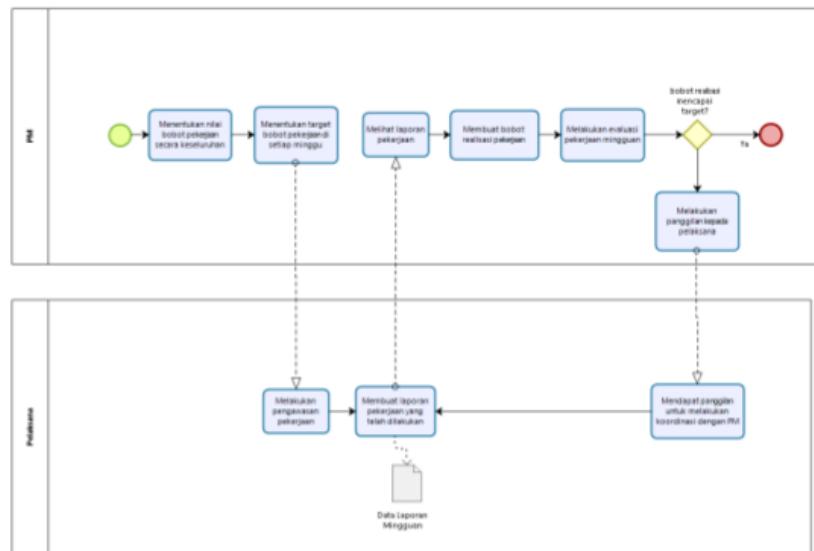
4.1.2 Pengawasan Lapangan

Pengawasan lapangan dibutuhkan untuk melakukan pengawasan dalam proses penggerjaan proyek. Orang yang bertanggung jawab dalam proses pengawasan proyek adalah pelaksana. Pelaksana akan memberitahukan kepada PM mengenai apa saja yang sudah dilakukan selama proses penggerjaan proyek. Selain itu pelaksana akan melampirkan foto untuk dikirimkan kepada PM sebagai bukti pekerjaan apa saja yang telah dilakukan di lapangan.

4.1.3 Pembuatan *Time Scheduling*

Time Scheduling merupakan suatu titik acuan pada saat proses penggerjaan proyek. *Time Scheduling* dibuat untuk memonitor sudah sejauh mana bobot

pekerjaan yang telah dilakukan dalam suatu project. *Time Scheduling* dibuat oleh *Project Manager* (PM). Pembuatan *Time Scheduling* sesuai dengan durasi pengerjaan yang telah ditentukan di dalam surat perjanjian kerja. *Time Scheduling* dibagi menjadi proses pekerjaan per minggu dengan target bobot yang ditentukan. Terdapat *S-Curve* sebagai alat kontrol kemajuan pekerjaan. Pada proses pembuatan *Time Scheduling* PM akan membuat bobot pekerjaan dari setiap uraian pekerjaan yang ada pada RAB. Setelah itu PM akan menentukan berapa lama waktu pengerjaan yang diperlukan oleh setiap uraian pekerjaan dan membuat target bobot pekerjaan di setiap minggunya. Setelah target bobot pekerjaan telah ditentukan, PM akan membuat bobot realisasi pekerjaan yang telah dilakukan setiap minggunya. Dimana bobot realisasi ini didapatkan dari hasil laporan pekerjaan mingguan yang telah dibuat oleh pelaksana. Setelah PM menentukan bobot realisasi pekerjaan, maka PM akan membandingkannya dengan target bobot pekerjaan yang telah dibuat pada saat pekerjaan proyek dilakukan. Dari hasil evaluasi ini jika target bobot pekerjaan tidak tercapai, maka PM akan memanggil seluruh pelaksana pada proyek tersebut untuk melakukan koordinasi mengenai target yang harus dicapai pada minggu berikutnya dan jika target bobot pekerjaan telah tercapai maka pekerjaan tetapi dilakukan sesuai target pekerjaan yang telah ditentukan. Proses pembuatan *Time Scheduling* dapat dilihat pada gambar 4.2. Saat ini dalam menentukan bobot realisasi pekerjaan diperlukan tatap muka antara pelaksana dan PM untuk melihat hasil laporan mingguan.



Gambar 4.2 Proses Pembuatan *Time Scheduling*

4.2 Deskripsi Umum Sistem

Sistem manajemen proyek pada PT. MIC merupakan suatu sistem yang digunakan untuk melakukan pembuatan rencana anggaran biaya, daftar harga satuan material, menampilkan daftar BoQ, menampilkan daftar proyek, dan penjadwalan proyek. Sistem ini berbasis website dan dikembangkan dengan kerangka ke Codeigniter. Dalam sistem ini memiliki enam aktor yaitu *Purchasing*, Pelaksana, Direktur, *Project Manager* (PM), Pengguna dan Admin. PM dapat melakukan pembuatan rencana anggaran biaya. Bagian *Purchasing* yang melakukan update harga terbaru dari bahan bahan material yang dibutuhkan di dalam suatu BoQ. Pelaksana melakukan monitoring di lapangan lalu membuat suatu lampiran terhadap proses pekerjaan yang telah dilakukan. Yaitu dengan melampirkan foto sebagai bukti bahwa kita telah mengerjakan bagian yang telah kita laporkan. PM dapat melakukan analisa harga akhir dari BoQ dan membuat *Time Scheduling*. Direktur dapat melihat Daftar RAB, Daftar Proyek, dan *Time Scheduling*.

1

4.3 Identifikasi Aktor

Tahap ini menggambarkan aktor-aktor yang terlibat atau berinteraksi dengan sistem. Identifikasi aktor menjelaskan apa yang bisa dilakukan oleh aktor terhadap sistem. Terdapat beberapa aktor yang berperan dalam menggunakan sistem, yaitu *Purchasing*, Pelaksana, PM (*Project Manager*), Direktur, Pengguna dan Admin. Secara umum, aktor-aktor tersebut dikatakan sebagai pengguna. Penjelasan mengenai aktor-aktor pada sistem ini dapat dilihat pada Tabel 4.1.

Tabel 4.1 Identifikasi Aktor

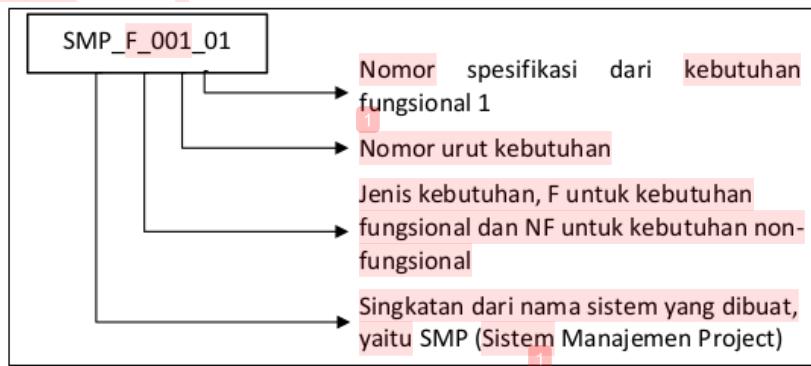
No	Nama Aktor	Deskripsi
1	<i>Purchasing</i>	<i>Purchasing</i> memiliki tugas untuk melakukan update harga nilai satuan material dan memasukkan harga material yang dibutuhkan pada perancangan BoQ
2	Pelaksana	Melakukan pengawasan yang terjadi di lapangan lalu membuat laporan mengenai apa saja yang sudah dikerjakan dan melampirkan foto hasil pekerjaan
3	PM (<i>Project Manager</i>)	Melakukan melakukn perancangan Rencana Anggaran Biaya (RAB), membuat <i>Time Scheduling</i> & dapat melihat proses pekerjaan yang telah dikerjakan pada setiap proyek
4	Direktur	Dapat melihat proses pekerjaan yang telah dikerjakan pada setiap proyek
5	Pengguna	Pengguna merupakan aktor yang dapat masuk ke dalam sistem dan keluar dari sistem.

6	Admin	Admin bertugas untuk mengolah akun yang ada pada sistem. Hal yang dapat dilakukan oleh admin adalah melakukan tambah pengguna, hapus pengguna dan lihat pengguna.
---	-------	---

1

4.4 Identifikasi Kebutuhan

Tahap ini menggambarkan daftar kebutuhan yang bisa dilakukan oleh sistem. Identifikasi kebutuhan menjelaskan tentang kebutuhan fungsional dan non-fungsional sistem. Pada setiap kebutuhan terdapat kode fungsi untuk mengidentifikasi kebutuhan tersebut. Struktur penamaan kode fungsi dapat dilihat pada Gambar 4.3.



Gambar 4.3 Struktur penamaan kode fungsi

4.4.1 Kebutuhan Fungsional

1 Kebutuhan fungsional menggambarkan apa yang bisa dilakukan oleh sistem. Kebutuhan ini menunjukkan fitur-fitur utama yang bisa dilakukan oleh aktor melalui sistem. Kebutuhan ini disimbolkan dengan huruf F. Contoh kebutuhan fungsional dituliskan sebagai berikut:

- a. SMP_F_001 : kebutuhan fungsional 1
- b. SMP_F_001_01 : spesifikasi 1 dari kebutuhan fungsional 1

Kebutuhan fungsional pada sistem manajemen proyek dapat dilihat pada Tabel 4.2.

Tabel 4.2 Kebutuhan Fungsional

No	Kode	Nama Fungsi	Deskripsi dan Spesifikasi Kebutuhan	Aktor
1	SMP_F_001	Masuk	Sistem menyediakan fungsi memberikan akses kepada Pengguna	Pengguna

			<p>pengguna untuk dapat masuk ke dalam sistem.</p>	
2	SMP_F_001_01	Masuk	<p><i>Login menggunakan data username dan password.</i></p>	Pengguna
3	SMP_F_002	Menambah Pengguna	<p>Sistem menyediakan fungsi untuk menambahkan pengguna baru.</p>	Admin
4	SMP_F_002_01	Menambah Pengguna	<p>Sistem menyimpan data username dan password.</p>	Admin
5	SMP_F_003	Keluar	<p>Sistem menyediakan fungsi untuk dapat menghapus sesi pengguna dan mengeluarkan pengguna dari dalam sistem.</p>	Pengguna
6	SMP_F_004	Melihat daftar pengguna	<p>Sistem menyediakan fungsi untuk menampilkan daftar pengguna.</p>	Admin
7	SMP_F_004_01	Melihat daftar pengguna	<p>Sistem memberikan data pengguna yang terdaftar di dalam sistem.</p>	Admin
8	SMP_F_005	Mengubah pengguna	<p>Sistem menyediakan fungsi untuk mengubah data pengguna.</p>	Admin
9	SMP_F_005_01	Mengubah pengguna	<p>Sistem menyimpan perubahan data pengguna.</p>	Admin
10	SMP_F_006	Menghapus pengguna	<p>Sistem menyediakan fungsi untuk menghapus pengguna.</p>	Admin
11	SMP_F_07	Melihat daftar barang	<p>Sistem menyediakan fungsi untuk menampilkan daftar barang.</p>	Purchasing & PM
12	SMP_F_07_01	Melihat daftar barang	<p>Sistem memberikan data nama, satuan, dan harga barang dari database</p>	Purchasing & PM
13	SMP_F_08	Menambah barang	<p>Sistem menyediakan fungsi untuk memasukkan data material baru.</p>	Purchasing
14	SMP_F_08_01	Menambah barang	<p>Sistem menyimpan data nama, satuan dan harga barang.</p>	Purchasing

15	SMP_F_09	Mengubah barang	Sistem menyediakan fungsi untuk merubah data material terbaru.	Purchasing
16	SMP_F_09_01	Mengubah barang	Sistem menyimpan perubahan data nama, satuan dan harga barang.	Purchasing
17	SMP_F_010	Menghapus barang	Sistem menyediakan fungsi untuk menghapus barang.	Purchasing
18	SMP_F_011	Melihat daftar laporan	Sistem menyediakan fungsi untuk menampilkan laporan pekerjaan.	Pelaksana & PM
19	SMP_F_011_01	Melihat daftar laporan	Sistem memberikan data nama proyek, tanggal, no spk, uraian pekerjaan, tenaga kerja, peralatan dan catatan.	Pelaksana & PM
20	SMP_F_012	Menambah laporan	Sistem menyediakan fungsi untuk memasukkan data laporan proses pekerjaan proyek.	Pelaksana
21	SMP_F_012_01	Menambah laporan	Sistem menyimpan data nama proyek, tanggal, no spk, uraian pekerjaan, tenaga kerja, peralatan dan catatan.	Pelaksana
22	SMP_F_013	Mengubah laporan	Sistem menyediakan fungsi mengubah data laporan pekerjaan.	Pelaksana & PM
23	SMP_F_013_01	Mengubah laporan	Sistem menyimpan perubahan data nama proyek, tanggal, no spk, uraian pekerjaan, tenaga kerja, peralatan dan catatan.	Pelaksana & PM
24	SMP_F_014	Menghapus laporan	Sistem menyediakan fungsi untuk menghapus laporan pekerjaan.	Pelaksana & PM
25	SMP_F_015	Melihat upah	Sistem menyediakan fungsi untuk menampilkan upah.	PM
26	SMP_F_015_01	Melihat upah	Sistem memberikan data nama pekerjaan, harga dan lokasi.	PM
27	SMP_F_016	Menambah upah	Sistem menyediakan fungsi untuk memasukkan data upah.	PM

28	SMP_F_016_01	Menambah upah	Sistem menyimpan data nama pekerjaan, harga dan lokasi.	PM
29	SMP_F_017	Mengubah upah	Sistem menyediakan fungsi mengubah data upah.	PM
30	SMP_F_017_01	Mengubah upah	Sistem menyimpan perubahan data nama pekerjaan, harga dan lokasi.	PM
31	SMP_F_018	Menghapus upah	Sistem menyediakan fungsi untuk menghapus data upah.	PM
32	SMP_F_019	Melihat analisis	Sistem menyediakan fungsi untuk menampilkan analisis.	PM
33	SMP_F_019_01	Melihat analisis	Sistem memberikan data nama analisa, satuan, lokasi dan harga.	PM
34	SMP_F_020	Menambah analisis	Sistem menyediakan fungsi untuk memasukkan data analisis.	PM
35	SMP_F_020_01	Menambah analisis	Sistem memberikan data nama analisa, satuan, lokasi, uraian, koefisien, dan harga total.	PM
36	SMP_F_021	Mengubah analisis	Sistem menyediakan fungsi mengubah data analisis.	PM
37	SMP_F_021_01	Mengubah analisis	Sistem memberikan data nama analisa, satuan, lokasi, uraian, koefisien, dan harga total.	PM
38	SMP_F_022	Menghapus analisis	Sistem menyediakan fungsi untuk menghapus data analisis.	PM
39	SMP_F_023	Melihat RAB	Sistem menyediakan fungsi untuk menampilkan RAB.	Direktur & PM
40	SMP_F_023_01	Melihat RAB	Sistem memberikan data nama RAB, satuan, nama pemilik proyek, lokasi dan total.	Direktur & PM
41	SMP_F_024	Menambah RAB	Sistem menyediakan fungsi untuk memasukkan data RAB.	PM
42	SMP_F_024_01	Menambah RAB	Sistem memberikan data nama RAB, nama proyek, uraian pekerjaan, volume dan harga total.	PM

43	SMP_F_025	Mengubah RAB	Sistem menyediakan fungsi mengubah data RAB.	PM
44	SMP_F_025_01	Mengubah RAB	Sistem memberikan data nama RAB, nama proyek, uraian pekerjaan, volume dan harga total.	PM
45	SMP_F_026	Menghapus RAB	Sistem menyediakan fungsi untuk menghapus data RAB.	PM
46	SMP_F_027	Melihat Proyek	Sistem menyediakan fungsi untuk menampilkan Proyek.	Direktur & PM
47	SMP_F_027_01	Melihat Proyek	Sistem memberikan data kode SPK, nama pemilik proyek dan lokasi.	Direktur & PM
48	SMP_F_028	Menambah Proyek	Sistem menyediakan fungsi untuk memasukkan data Proyek.	PM
49	SMP_F_028_01	Menambah Proyek	Sistem memberikan data kode SPK, nama pemilik proyek, deskripsi dan lokasi.	PM
50	SMP_F_029	Mengubah Proyek	Sistem menyediakan fungsi mengubah data Proyek.	PM
51	SMP_F_029_01	Mengubah Proyek	Sistem memberikan data kode SPK, nama pemilik proyek, deskripsi dan lokasi.	PM
52	SMP_F_030	Menghapus Proyek	Sistem menyediakan fungsi untuk menghapus data Proyek.	PM
53	SMP_F_031	Melihat <i>Time Scheduling</i>	Sistem menyediakan fungsi melihat <i>Time Scheduling</i> .	Direktur & PM
54	SMP_F_031_01	Melihat <i>Time Scheduling</i>	Sistem memberikan data bobot pekerjaan, bobot rencana, dan bobot realisasi.	Direktur & PM
55	SMP_F_032	Menambah <i>Time Scheduling</i>	Sistem menyediakan fungsi untuk memasukkan data <i>Time Scheduling</i> .	PM
56	SMP_F_032_01	Menambah <i>Time Scheduling</i>	Sistem menyimpan data bobot pekerjaan, bobot rencana, dan bobot realisasi.	PM
57	SMP_F_033	Mengubah <i>Time Scheduling</i>	Sistem menyediakan fungsi untuk mengubah data <i>Time Scheduling</i> .	PM

58	SMP_F_033_01	Mengubah <i>Time Scheduling</i>	Sistem menyimpan perubahan data bobot pekerjaan, bobot rencana, dan bobot realisasi.	PM
59	SMP_F_034	Menghapus <i>Time Scheduling</i>	Sistem menyediakan fungsi untuk menghapus data <i>Time Scheduling</i> .	PM

2 4.4.2 Kebutuhan Non Fungsional

Kebutuhan non-fungsional merupakan kebutuhan perilaku yang dimiliki sistem. Kebutuhan non-fungsional akan dijelaskan di Tabel 4.3. Kemampuan yang harus dimiliki sistem adalah dapat diakses dari berbagai *browser* yang digunakan di pengguna yaitu Google Chrome dan Mozilla Firefox.

Tabel 4.3 Kebutuhan non-Fungsional

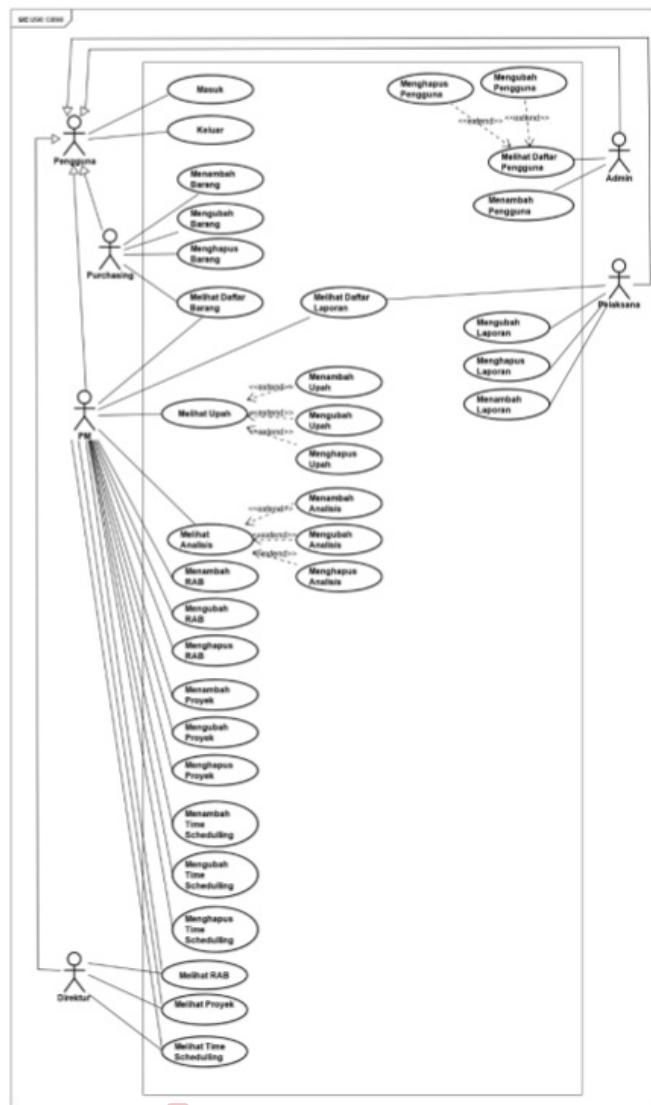
No	Kode	Nama	Keterangan
1.	SPM_NF_001	<i>Compatibility</i>	Sistem dapat berjalan dengan baik pada <i>browser</i> Google Chrome Desktop versi 77.0.3865.120 dan Mozilla Firefox Desktop versi 68.2.0

1 4.5 Pemodelan Kebutuhan

Pemodelan kebutuhan berfungsi untuk menggambarkan fungsi yang bisa dilakukan oleh sistem. Pada tahap ini dibuat *use case diagram* dan *use case scenario*. *Use case diagram* menjelaskan apa yang bisa dilakukan aktor pada sistem manajemen proyek. *Use case scenario* menjelaskan secara rinci *use case* yang dilakukan oleh sistem.

4.5.1 Use Case Diagram

Berdasarkan kebutuhan fungsional pada Tabel 4.2, maka dibuat sebuah *use case diagram* berdasarkan peran aktor yang ada pada Tabel 4.1. Gambar 4.4 mewakili setiap kebutuhan fungsional yang ada pada sistem manajemen proyek.



Gambar 4.4 Use Case Diagram

4.5.2 Use Case Scenario

Use case scenario merupakan urutan aksi yang terjadi saat fungsi pada sistem dijalankan. *Use case scenario* dapat berisi tujuan dari *use case (objective)*, aktor yang menjalankan (*actor*), kondisi awal (*pre-condition*), kondisi akhir

(postcondition), alur utama (main flow) saat aktor melakukan *use case* tersebut, serta alur alternatif (alternative flow) jika dibutuhkan. Dibawah ini adalah *use case* skenario yang digunakan untuk menggambarkan urutan kerja sistem berdasarkan *use case* yang telah didefinisikan sebelumnya.

4.5.2.1 Use Case Scenario Masuk

Use case scenario masuk menggambarkan alur kegiatan yang dilakukan pengguna untuk masuk ke dalam sistem. Alur kerja tersebut ditunjukkan pada Tabel 4.5.

Tabel 4.5 *Use Case Scenario* masuk

Masuk (SMP_F_001)	
Objective	Pengguna dapat masuk ke dalam sistem dan mendapatkan hak akses sesuai dengan peran dari masing-masing aktor.
Actor	Pengguna
Pre-condition	Pengguna berada di halaman awal.
Main Flow	<ol style="list-style-type: none"> 1. Pengguna memasukkan data <i>username</i> dan <i>password</i>. 2. Pengguna menekan tombol "login". 3. Sistem melakukan validasi data. 4. Pengguna masuk ke dalam sistem sesuai dengan hak aksesnya.
Alternative Flow	<ol style="list-style-type: none"> a Jika pengguna tidak mengisi <i>username</i> dan <i>password</i>, maka sistem akan menampilkan "Data tidak boleh kosong". b Jika pengguna memasukkan data yang tidak valid, maka sistem akan menampilkan "Data tidak ditemukan".
Post Condition	Pengguna masuk ke dalam sistem sesuai dengan hak aksesnya.

4.5.2.2 Use Case Scenario Menambah Pengguna

Use case scenario menambah pengguna menggambarkan alur kegiatan yang dilakukan admin untuk mendaftarkan anggota baru. Alur kerja tersebut ditunjukkan pada Tabel 4.6.

Tabel 4.6 *Use case scenario* menambah pengguna

Menambah Pengguna (SMP_F_002)	
Objective	Admin berhasil menambah anggota baru.
Actor	Admin

Pre-condition	Admin berada di halaman daftar pengguna.
Main Flow	<ol style="list-style-type: none"> 1. Admin menekan tombol "Tambah Pengguna". 2. Sistem menampilkan <i>form</i> pada halaman tambah pengguna. 3. Admin memasukkan data dan menekan tombol daftar. 4. Sistem melakukan validasi data. 5. Pengguna baru berhasil ditambahkan
Alternative Flow	<ol style="list-style-type: none"> 4.a Jika admin tidak mengisi kolom data yang ada, maka sistem akan menampilkan "Data tidak boleh kosong". 4.b Jika pengguna mengisi <i>username</i> yang telah terdaftar di dalam sistem, maka sistem akan menampilkan "Username telah terdaftar".
Post Condition	Kembali ke halaman awal .

4.5.2.3 Use Case Scenario Keluar

Use case scenario keluar menggambarkan alur kegiatan yang dilakukan pengguna untuk keluar dari sistem. Alur kerja tersebut ditunjukkan pada Tabel 4.7.

Tabel 4.7 Use case scenario keluar

Keluar (SMP_F_003)	
Objective	Pengguna keluar dari sistem.
Actor	Pengguna
Pre-condition	Pengguna telah masuk ke dalam sistem.
Main Flow	<ol style="list-style-type: none"> 1. Pengguna menekan tombol "keluar". 2. Sistem menghapus sesi pengguna.
Alternative Flow	-
Post Condition	Pengguna keluar dari sistem.

4.5.2.4 Use Case Scenario Melihat Daftar Pengguna

Use case scenario melihat daftar pengguna menggambarkan alur kegiatan yang dilakukan admin untuk melihat daftar pengguna. Alur kerja tersebut ditunjukkan pada Tabel 4.8.

Tabel 4.8 Use case scenario melihat daftar pengguna

Melihat Daftar Pengguna (SMP_F_004)	
Objective	Admin dapat melihat data pengguna.
Actor	Admin
Pre-condition	Admin telah masuk ke dalam sistem.
Main Flow	<ol style="list-style-type: none">1. Admin memilih menu daftar data pengguna.2. Sistem menampilkan daftar data pengguna yang terdapat di dalam database.
Alternative Flow	-
Post Condition	Halaman daftar pengguna berhasil ditampilkan.

4.5.2.5 Use Case Scenario Mengubah Pengguna

Use case scenario mengubah pengguna menggambarkan alur kegiatan yang dilakukan admin untuk mengubah data pengguna dari sistem. Alur kerja tersebut ditunjukkan pada Tabel 4.9.

Tabel 4.9 Use case scenario mengubah pengguna

Mengubah Pengguna (SMP_F_005)	
Objective	Admin dapat mengubah data pengguna dari sistem.
Actor	Admin
Pre-condition	Admin berada pada halaman daftar pengguna.
Main Flow	<ol style="list-style-type: none">1. Admin menekan tombol "edit" pada data pengguna yang dipilih.2. Sistem menampilkan data pengguna yang dipilih.3. Admin memasukkan data yang akan dirubah pada data pengguna yang dipilih.4. Admin menekan tombol "update".5. Sistem menyimpan perubahan data ke dalam sistem.
Alternative Flow	-
Post Condition	Kembali ke halaman daftar pengguna.

4.5.2.6 Use Case Scenario Menghapus Pengguna

Use case scenario menghapus pengguna menggambarkan alur kegiatan yang dilakukan admin untuk menghapus pengguna dari sistem. Alur kerja tersebut ditunjukkan pada Tabel 4.10.

Tabel 4.10 Use case scenario menghapus pengguna

Menghapus Pengguna (SMP_F_006)	
Objective	Admin dapat menghapus data pengguna dari sistem.
Actor	Admin
Pre-condition	Admin berada pada halaman daftar pengguna.
Main Flow	<ol style="list-style-type: none">1. Admin menekan tombol "delete" pada data pengguna yang akan dihapus.2. Sistem menghapus data pengguna yang dipilih dari sistem.
Alternative Flow	-
Post Condition	Data pengguna yang dipilih berhasil terhapus dari sistem.

1
4.5.2.7 Use Case Scenario Melihat Daftar Barang

Use case scenario melihat daftar barang menggambarkan alur kegiatan yang dilakukan Purchasing dan PM untuk melihat data barang yang ada di dalam sistem. Alur kerja tersebut ditunjukkan pada Tabel 4.11.

Tabel 4.11 Use case scenario melihat daftar barang

Melihat Daftar Barang (SMP_F_007)	
Objective	Purchasing dan PM dapat melihat data barang dari dalam sistem.
Actor	Purchasing dan PM
Pre-condition	Purchasing dan PM telah masuk ke dalam sistem.
Main Flow	<ol style="list-style-type: none">1. Purchasing dan PM menekan menu Master Barang.2. Sistem menampilkan data daftar barang yang terdiri dari aksi (edit/delete), nama barang, satuan barang dan harga barang
Alternative Flow	-
Post Condition	Sistem berhasil menampilkan daftar barang dari dalam sistem.

1
4.5.2.8 Use Case Scenario Menambah Barang

Use case scenario menambah barang menggambarkan alur kegiatan yang dilakukan purchasing untuk menambahkan data barang ke dalam sistem. Alur kerja tersebut ditunjukkan pada Tabel 4.12.

Tabel 4.12 Use case scenario menambah barang

Menambah Barang (SMP_F_008)	
Objective	<i>Purchasing</i> dapat menambahkan data barang baru ke dalam sistem.
Actor	<i>Purchasing</i>
Pre-condition	<i>Purchasing</i> berada pada halaman master barang.
Main Flow	<ol style="list-style-type: none"> 1. <i>Purchasing</i> menekan tombol “Tambahkan Barang”. 2. Sistem menampilkan form tambahkan barang yang terdiri dari nama barang, satuan barang, dan harga barang. 3. <i>Purchasing</i> memasukkan data barang baru yang dibutuhkan pada form, kemudian menekan tombol “simpan”. 4. Sistem menyimpan data barang baru ke dalam sistem. 5. Sistem menampilkan halaman master barang.
Alternative Flow	-
Post Condition	Kembali ke halaman master barang.

1 4.5.2.9 Use Case Scenario Mengubah Barang

Use case scenario mengubah barang menggambarkan alur kegiatan yang dilakukan *purchasing* untuk merubah data barang dari dalam sistem. Alur kerja tersebut ditunjukkan pada Tabel 4.13.

Tabel 4.13 Use case scenario mengubah barang

Mengubah Barang (SMP_F_009)	
Objective	<i>Purchasing</i> merubah data barang dari dalam sistem.
Actor	<i>Purchasing</i>
Pre-condition	<i>Purchasing</i> berada pada halaman master barang.
Main Flow	<ol style="list-style-type: none"> 1. <i>Purchasing</i> menekan tombol “edit” pada data yang akan dirubah. 2. Sistem menampilkan halaman ubah data barang yang telah dipilih dan menampilkan form dengan data nama barang, satuan barang, dan harga barang. 3. <i>Purchasing</i> mengubah data harga barang dan memilih tombol “Simpan”.

	4. Sistem menyimpan perubahan data barang yang dipilih ke dalam sistem. 5. Sistem menampilkan halaman master barang.
Alternative Flow	-
Post Condition	Kembali ke halaman master barang.

1
4.5.2.10 Use Case Scenario Menghapus Barang

Use case scenario menghapus barang menggambarkan alur kegiatan yang dilakukan *purchasing* untuk menghapus data barang yang ada di dalam sistem. Alur kerja tersebut ditunjukkan pada Tabel 4.14.

Tabel 4.14 Use case scenario menghapus barang

Menghapus Barang (SMP_F_010)	
Objective	<i>Purchasing</i> menghapus data barang yang ada di dalam sistem.
Actor	<i>Purchasing</i>
Pre-condition	<i>Purchasing</i> berada pada halaman master barang.
Main Flow	1. <i>Purchasing</i> menekan tombol “delete” pada data yang dipilih. 2. Sistem menghapus data barang yang dipilih dari dalam sistem. 3. Sistem menampilkan halaman master barang.
Alternative Flow	-
Post Condition	Kembali ke halaman daftar barang.

1
4.5.2.11 Use Case Scenario Melihat Daftar Laporan

Use case scenario melihat daftar laporan menggambarkan alur kegiatan yang dilakukan Pelaksana & PM untuk melihat daftar laporan suatu proyek. Alur kerja tersebut ditunjukkan pada Tabel 4.15.

Tabel 4.15 Use case scenario melihat daftar laporan

Melihat Daftar Laporan (SMP_F_011)	
Objective	Pelaksana & PM dapat melihat daftar laporan suatu proyek
Actor	Pelaksana & PM
Pre-condition	Pelaksana & PM telah masuk ke dalam sistem.
Main Flow	1. <i>Purchasing</i> dan PM menekan menu Master Barang.

1	2. Sistem menampilkan data daftar barang yang terdiri dari nama laporan, nama proyek, nama pelaksana, tanggal dan detail(<i>edit/ delete</i>).
Alternative Flow	-
Post Condition	Sistem berhasil menampilkan daftar laporan dari dalam sistem.

4.5.2.12 Use Case Scenario Menambah Laporan

Use case scenario menambah laporan menggambarkan alur kegiatan yang dilakukan pelaksana untuk membuat laporan proses pekerjaan suatu proyek. Alur kerja tersebut ditunjukkan pada Tabel 4.16.

Tabel 4.16 Use case scenario menambah laporan

Menambah Laporan (SMP_F_012)	
Objective	Pelaksana dapat menambah laporan proses pekerjaan suatu proyek.
Actor	Pelaksana
Pre-condition	Pelaksana berada pada halaman daftar laporan.
Main Flow	<ol style="list-style-type: none"> 1. Pelaksana menekan tombol tambah laporan. 2. Sistem menampilkan halaman tambah laporan 3. Pelaksana menambahkan data laporan dan menekan tombol simpan. 4. Sistem menyimpan data laporan ke dalam sistem. 5. Sistem menampilkan halaman daftar laporan.
Alternative Flow	-
Post Condition	Kembali ke halaman daftar laporan.

4.5.2.13 Use Case Scenario Mengubah Laporan

Use case scenario mengubah laporan menggambarkan alur kegiatan yang dilakukan Pelaksana & PM untuk mengubah data laporan proses pekerjaan dari sistem. Alur kerja tersebut ditunjukkan pada Tabel 4.17.

Tabel 4.17 Use case scenario mengubah laporan

Mengubah Laporan (SMP_F_013)	
Objective	Pelaksana & PM dapat mengubah data laporan proses pekerjaan dari sistem.

Actor	Pelaksana & PM
Pre-condition	Pelaksana & PM berada pada halaman laporan.
Main Flow	<ol style="list-style-type: none"> 1. Pelaksana & PM menekan tombol "edit" pada data laporan yang dipilih. 2. Sistem menampilkan halaman ubah laporan pada data yang dipilih. 3. Pelaksana & PM merubah data laporan dan menekan tombol "update". 4. Sistem mengubah data laporan yang dipilih ke dalam sistem. 5. Sistem menampilkan halaman daftar laporan
Alternative Flow	-
Post Condition	Kembali ke halaman daftar laporan.

1 4.5.2.14 Use Case Scenario Menghapus Laporan

Use case scenario menghapus laporan menggambarkan alur kegiatan yang dilakukan Pelaksana & PM untuk menghapus data laporan proses pekerjaan dari sistem. Alur kerja tersebut ditunjukkan pada Tabel 4.18.

Tabel 4.18 Use case scenario menghapus laporan

Menghapus Laporan (SMP_F_014)	
Objective	Pelaksana & PM menghapus data laporan proses pekerjaan dari sistem.
Actor	Pelaksana & PM
Pre-condition	Pelaksana & PM berada pada halaman laporan.
Main Flow	<ol style="list-style-type: none"> 1. Pelaksana & PM memilih menekan tombol "delete" pada data laporan yang akan dihapus. 2. Sistem menghapus data laporan yang dipilih dari dalam sistem. 3. Sistem menampilkan halaman daftar laporan.
Alternative Flow	-
Post Condition	Kembali ke halaman daftar laporan.

4.5.2.15 Use Case Scenario Melihat Upah

Use case scenario melihat upah menggambarkan alur kegiatan yang dilakukan PM untuk menampilkan data upah dari sistem. Alur kerja tersebut ditunjukkan pada Tabel 4.19.

Tabel 4.19 Use case scenario melihat upah

Melihat Upah (SMP_F_015)	
Objective	PM dapat melihat data upah.
Actor	PM
Pre-condition	Aktor telah masuk ke dalam sistem
Main Flow	<ol style="list-style-type: none">1. PM menekan menu "Master Upah".2. Sistem menampilkan halaman master upah dan data upah yang terdiri dari aksi (edit/delete), nama upah, harga upah, dan lokasi upah.
Alternative Flow	-
Post Condition	Sistem berhasil menampilkan data upah dari dalam sistem.

4.5.2.16 Use Case Scenario Menambah Upah

Use case scenario menambah upah menggambarkan alur kegiatan yang dilakukan PM untuk menambah data upah ke sistem. Alur kerja tersebut ditunjukkan pada Tabel 4.20.

Tabel 4.20 Use case scenario menambah upah

Menambah Upah (SMP_F_016)	
Objective	PM dapat menambah data upah baru ke dalam sistem.
Actor	PM
Pre-condition	PM berada pada halaman master upah.
Main Flow	<ol style="list-style-type: none">1. PM menekan tombol "Tambahkan Upah".2. Sistem menampilkan form tambah upah yang terdiri dari nama upah, harga upah, dan lokasi upah.3. PM memasukkan data upah yang dibutuhkan form, dan menekan tombol "simpan".4. Sistem menyimpan data upah baru ke dalam sistem.5. Sistem menampilkan halaman master upah.
Alternative Flow	-
Post Condition	Kembali ke halaman melihat upah.

1

4.5.2.17 Use Case Scenario Mengubah Upah

Use case scenario mengubah upah menggambarkan alur kegiatan yang dilakukan PM untuk mengubah data upah dari sistem. Alur kerja tersebut ditunjukkan pada Tabel 4.21.

Tabel 4.21 Use case scenario mengubah upah

Mengubah Upah (SMP_F_017)	
Objective	PM dapat mengubah data upah.
Actor	PM
Pre-condition	PM berada pada halaman master upah.
Main Flow	<ol style="list-style-type: none"> 1. PM memilih tombol "edit" pada data upah yang dipilih. 2. Sistem mengambil data detail upah pada data yang dipilih dari sistem. 3. Sistem menampilkan halaman mengubah upah pada data yang dipilih beserta data yang terdiri dari nama upah, harga upah, dan lokasi upah. 4. PM mengubah data harga upah dan menekan tombol simpan. 5. Sistem menyimpan perubahan data upah ke sistem. 6. Sistem menampilkan halaman master upah.
Alternative Flow	-
Post Condition	Kembali ke halaman melihat upah.

1

4.5.2.18 Use Case Scenario Menghapus Upah

Use case scenario menghapus upah menggambarkan alur kegiatan yang dilakukan PM untuk menghapus data upah dari sistem. Alur kerja tersebut ditunjukkan pada Tabel 4.22.

Tabel 4.22 Use case scenario menghapus upah

Menghapus Upah (SMP_F_018)	
Objective	PM dapat menghapus data upah dari sistem.
Actor	PM
Pre-condition	PM berada pada halaman master upah.
Main Flow	<ol style="list-style-type: none"> 1. PM menekan tombol "delete" pada data upah yang dipilih.

	2. Sistem menghapus data upah yang dipilih dari dalam sistem. 3. Sistem menampilkan halaman master upah.
Alternative Flow	-
Post Condition	Kembali ke halaman melihat upah.

1
4.5.2.19 Use Case Scenario Melihat Analisis

Use case scenario melihat analisis menggambarkan alur kegiatan yang dilakukan PM untuk melihat analisis dari sistem. Alur kerja tersebut ditunjukkan pada Tabel 4.23.

Tabel 4.23 Use case scenario melihat analisis

Melihat Analisis (SMP_F_019)	
Objective	PM dapat melihat data analisis.
Actor	PM
Pre-condition	Aktor telah masuk ke dalam sistem
Main Flow	1. PM menekan menu analisa 2. Sistem menampilkan data analisis dari dalam database, data terdiri dari nama analisa, nama satuan, nama lokasi, harga, dan detail (lihat/ delete).
Alternative Flow	-
Post Condition	Sistem berhasil menampilkan data analisis dari sistem.

1
4.5.2.20 Use Case Scenario Menambah Analisis

Use case scenario menambah analisis menggambarkan alur kegiatan yang dilakukan PM untuk menambah analisis ke sistem. Alur kerja tersebut ditunjukkan pada Tabel 4.24.

Tabel 4.24 Use case scenario menambah analisis

Menambah Analisis (SMP_F_020)	
Objective	PM dapat menambah data analisis baru.
Actor	PM
Pre-condition	PM berada pada halaman masteranalisa.
Main Flow	1. PM menekan tombol tambah analisa. 2. Sistem mengambil data satuan barang dari sistem.

	<ol style="list-style-type: none"> 3. Sistem menampilkan halaman analisaAdd berupa <i>form</i> yang terdiri dari nama analisa, satuan analisa, lokasi analisa, uraian bahan, koefisien bahan, uraian upah dan koefisien upah. 4. PM memasukkan data analisis baru dan menekan tombol “simpan”. 5. Sistem menyimpan data analisis baru ke dalam sistem. 6. Sistem mengambil data <i>recent id</i>. 7. Sistem menyimpan data detail analisis barang ke dalam sistem. 8. Sistem menyimpan data detail analisis upah ke dalam sistem. 9. Sistem menampilkan halaman masteranalisa.
Alternative Flow	-
Post Condition	Kembali ke halaman masteranalisa.

4.5.2.21 Use Case Scenario Mengubah Analisis

Use case scenario mengubah analisis menggambarkan alur kegiatan yang dilakukan PM untuk mengubah data analisis dari sistem. Alur kerja tersebut ditunjukkan pada Tabel 4.25.

Tabel 4.25 Use case scenario mengubah analisis

Mengubah Analisis (SMP_F_021)	
Objective	PM dapat mengubah data analisis.
Actor	PM
Pre-condition	PM berada pada halaman masteranalisa.
Main Flow	<ol style="list-style-type: none"> 1. PM menekan tombol “lihat” pada data yang dipilih. 2. Sistem mengambil data <i>detail</i> barang dari dalam sistem. 3. Sistem mengambil data detail upah dari dalam sistem. 4. Sistem mengambil data detail analisa dari dalam sistem. 5. Sistem menampilkan halaman analisaEdit pada data yang dipilih dan menampilkan data nama analisa, satuan analisa, lokasi analisa, uraian bahan, koefisien bahan, uraian upah dan koefisien upah. 6. PM mengubah data analisis dan menekan tombol “simpan”.

	7. Sistem mengubah data analisis ke dalam sistem. 8. Sistem menghapus data detail analisis dari dalam sistem. 9. Sistem menyimpan data detail analisis barang ke dalam sistem. 10. Sistem menyimpan data detail analisis upah ke dalam sistem. 11. Sistem menampilkan halaman masteranalisa.
Alternative Flow	-
Post Condition	Kembali ke halaman masteranalisa.

1 4.5.2.22 Use Case Scenario Menghapus Analisis

Use case scenario menghapus analisis menggambarkan alur kegiatan yang dilakukan PM untuk menghapus data analisis dari sistem. Alur kerja tersebut ditunjukkan pada Tabel 4.26.

Tabel 4.26 Use case scenario menghapus analisis

Menghapus Analisis (SMP_F_022)	
Objective	PM dapat menghapus data analisis dari sistem.
Actor	PM
Pre-condition	PM berada pada halaman masteranalisa.
Main Flow	1. PM menekan tombol "delete" pada data yang dipilih. 2. Sistem menghapus data analisis yang dipilih dari dalam sistem. 3. Sistem menampilkan halaman masteranalisa.
Alternative Flow	-
Post Condition	Kembali ke halaman analisis.

1 4.5.2.23 Use Case Scenario Melihat RAB

Use case scenario melihat RAB menggambarkan alur kegiatan yang dilakukan PM & Direktur untuk melihat RAB dari dalam sistem. Alur kerja tersebut ditunjukkan pada Tabel 4.27.

Tabel 4.27 Use case scenario melihat RAB

Melihat RAB (SMP_F_023)	
Objective	PM & Direktur dapat melihat RAB dari dalam sistem.

Actor	PM & Direktur
Pre-condition	Aktor telah masuk ke dalam sistem.
Main Flow	<ol style="list-style-type: none"> 1. Aktor menekan menu RAB. 2. Sistem menampilkan data RAB dari dalam <i>database</i> yang berisi data nama RAB, nama pemilik <i>project</i>, lokasi <i>project</i>, total nilai, dan detail (lihat/ <i>delete</i>).
Alternative Flow	-
Post Condition	Sistem berhasil menampilkan RAB dari dalam sistem.

4.5.2.24 Use Case Scenario Menambah RAB

Use case scenario menambah RAB menggambarkan alur kegiatan yang dilakukan PM untuk menambah data RAB ke dalam sistem. Alur kerja tersebut ditunjukkan pada Tabel 4.28.

Tabel 4.28 Use case scenario menambah RAB

Menambah RAB (SMP_F_024)	
Objective	PM dapat menambah data baru RAB ke dalam sistem.
Actor	PM
Pre-condition	PM berada pada halaman RAB
Main Flow	<ol style="list-style-type: none"> 1. PM menekan menu “Tambah RAB”. 2. Sistem menampilkan halaman tambah RAB dan form yang berisi nama rab, nama <i>project</i>, uraian pekerjaan persiapan, <i>volume</i> pekerjaan persiapan, uraian pekerjaan tanah dan struktur, <i>volume</i> pekerjaan tanah dan struktur, uraian pekerjaan arsitek, <i>volume</i> pekerjaan arsitek, uraian pekerjaan MEP, <i>volume</i> pekerjaan MEP, uraian pekerjaan lain-lain, dan <i>volume</i> pekerjaan lain-lain. 3. PM mengisi data RAB baru dan menekan tombol “simpan”. 4. Sistem menyimpan data RAB ke dalam sistem. 5. Sistem menampilkan halaman RAB.
Alternative Flow	-
Post Condition	Kembali ke halaman RAB.

1

4.5.2.25 Use Case Scenario Mengubah RAB

Use case scenario mengubah RAB menggambarkan alur kegiatan yang dilakukan PM untuk mengubah data RAB dari sistem. Alur kerja tersebut ditunjukkan pada Tabel 4.29.

Tabel 4.29 Use case scenario mengubah RAB

Mengubah RAB (SMP_F_025)	
Objective	PM dapat mengubah data RAB dari sistem.
Actor	PM 1
Pre-condition	PM berada pada halaman RAB.
Main Flow	<ol style="list-style-type: none"> 1. PM menekan tombol "edit" pada data RAB yang dipilih. 2. Sistem menampilkan halaman mengubah RAB pada data yang dipilih dan form yang berisi nama rab, nama project, uraian pekerjaan persiapan, volume pekerjaan persiapan, uraian pekerjaan tanah dan struktur, volume pekerjaan tanah dan struktur, uraian pekerjaan arsitek, volume pekerjaan arsitek, uraian pekerjaan MEP, volume. 3. PM mengubah data RAB dan menekan tombol "update". 4. Sistem mengubah data RAB ke dalam sistem. 5. Sistem menampilkan halaman RAB.
Alternative Flow	-
Post Condition	Kembali ke halaman RAB.

1

4.5.2.26 Use Case Scenario Menghapus RAB

Use case scenario menghapus RAB menggambarkan alur kegiatan yang dilakukan PM untuk menghapus data RAB dari sistem. Alur kerja tersebut ditunjukkan pada Tabel 4.30.

Tabel 4.30 Use case scenario menghapus RAB

Menghapus RAB (SMP_F_026)	
Objective	PM dapat menghapus data RAB dari sistem.
Actor	PM 1
Pre-condition	PM berada pada halaman RAB.
Main Flow	<ol style="list-style-type: none"> 1. PM menekan tombol "delete" pada data RAB yang dipilih. 2. Sistem menghapus data RAB yang dipilih dari dalam sistem.

	3. Sistem menampilkan halaman RAB.
Alternative Flow	-
Post Condition	Kembali ke halaman RAB.

1 4.5.2.27 Use Case Scenario Melihat Proyek

Use case scenario melihat proyek menggambarkan alur kegiatan yang dilakukan PM & Direktur untuk melihat data proyek yang ada di dalam sistem. Alur kerja tersebut ditunjukkan pada Tabel 4.31.

Tabel 4.31 Use case scenario melihat proyek

Melihat Proyek (SMP_F_027)	
Objective	PM & Direktur dapat melihat data proyek.
Actor	PM & Direktur
Pre-condition	Aktor telah masuk ke dalam sistem.
Main Flow	<ol style="list-style-type: none"> 1. Aktor menekan menu proyek 2. Sistem menampilkan data proyek dari dalam sistem yang terdiri dari data no, kode SPK, nama proyek, nama pemilik proyek, lokasi proyek dan aksi (<i>detail, edit, delete</i>).
Alternative Flow	-
Post Condition	Sistem berhasil menampilkan data proyek dari dalam sistem.

1 4.5.2.28 Use Case Scenario Menambah Proyek

Use case scenario menambah proyek menggambarkan alur kegiatan yang dilakukan PM untuk menambah data proyek ke dalam sistem. Alur kerja tersebut ditunjukkan pada Tabel 4.32.

Tabel 4.32 Use case scenario menambah proyek

Menambah Proyek (SMP_F_028)	
Objective	PM dapat menambah data proyek ke dalam sistem.
Actor	PM
Pre-condition	PM berada pada halaman proyek.
Main Flow	<ol style="list-style-type: none"> 1. PM menekan menu “Tambah Proyek”. 2. Sistem menampilkan halaman tambah proyek berupa form yang terdiri dari nama proyek, no. SPK, deskripsi, pemilik proyek dan lokasi.

	3. PM memasukkan data proyek baru dan menekan tombol “simpan”. 4. Sistem menyimpan data proyek baru ke dalam sistem. 5. Sistem menampilkan halaman proyek.
Alternative Flow	-
Post Condition	Kembali ke halaman proyek.

1 4.5.2.29 Use Case Scenario Mengubah Proyek

Use case scenario mengubah proyek menggambarkan alur kegiatan yang dilakukan PM untuk mengubah data proyek dari dalam sistem. Alur kerja tersebut ditunjukkan pada Tabel 4.33.

Tabel 4.33 Use case scenario mengubah proyek

Mengubah Proyek (SMP_F_029)	
Objective	PM dapat mengubah data proyek dari dalam sistem.
Actor	PM 1
Pre-condition	PM berada pada halaman proyek.
Main Flow	1. PM menekan tombol “edit” pada data proyek yang dipilih. 2. Sistem menampilkan halaman mengubah proyek pada data yang dipilih dan menampilkan data nama proyek, kode proyek, deskripsi, pemilik proyek dan lokasi. 3. PM mengubah data proyek dan menekan tombol “simpan”. 4. Sistem mengubah data ke dalam sistem. 5. Sistem menampilkan halaman proyek.
Alternative Flow	-
Post Condition	Kembali ke halaman proyek.

1 4.5.2.30 Use Case Scenario Menghapus Proyek

Use case scenario menghapus proyek menggambarkan alur kegiatan yang dilakukan PM untuk menghapus data proyek dari dalam sistem. Alur kerja tersebut ditunjukkan pada Tabel 4.34.

Tabel 4.34 Use case scenario menghapus proyek

Menghapus Proyek (SMP_F_030)	
Objective	PM dapat menghapus data proyek dari dalam sistem.

Actor	PM
Pre-condition	PM berada pada halaman proyek.
Main Flow	<ol style="list-style-type: none"> 1. PM menekan tombol "delete" pada data proyek yang dipilih. 2. Sistem menghapus data proyek yang dipilih dari dalam sistem. 3. Sistem menampilkan halaman proyek.
Alternative Flow	-
Post Condition	Kembali ke halaman proyek.

4.5.2.31 Use Case Scenario Melihat Time Scheduling

Use case scenario melihat time scheduling menggambarkan alur kegiatan yang dilakukan PM & Direktur untuk melihat data Time Scheduling dari sistem. Alur kerja tersebut ditunjukkan pada Tabel 4.35.

Tabel 4.35 Use case scenario melihat time scheduling

Melihat Time Scheduling (SMP_F_031)	
Objective	PM & Direktur dapat melihat data Time Scheduling.
Actor	PM & Direktur
Pre-condition	PM & Direktur telah masuk ke dalam sistem.
Main Flow	<ol style="list-style-type: none"> 1. PM & Direktur menekan menu proyek 2. Sistem menampilkan data proyek dari dalam sistem yang terdiri dari data nama schedule, nama proyek, nama pelaksana, tanggal dan detail (edit, delete).
Alternative Flow	-
Post Condition	Sistem berhasil menampilkan time scheduling dari dalam sistem.

4.5.2.32 Use Case Scenario Menambah Time Scheduling

Use case scenario menambah time scheduling menggambarkan alur kegiatan yang dilakukan PM untuk memasukkan data Time Scheduling ke dalam sistem. Alur kerja tersebut ditunjukkan pada Tabel 4.36.

Tabel 4.36 Use case scenario menambah time scheduling

Menambah Time Scheduling (SMP_F_032)	
Objective	PM dapat Memasukan data Time Scheduling baru ke dalam sistem.
Actor	PM

Pre-condition	PM berada pada halaman <i>time scheduling</i> .
Main Flow	<ol style="list-style-type: none"> 1. PM menekan tombol tambah. 2. Sistem menampilkan halaman tambah <i>time scheduling</i>. 1 3. PM memasukkan data <i>time scheduling</i> dan menekan tombol “simpan”. 4. Sistem menyimpan data <i>time scheduling</i> ke dalam sistem. 5. Sistem menampilkan halaman <i>time scheduling</i>.
Alternative Flow	-
Post Condition	Kembali ke halaman <i>time scheduling</i> .

4.5.2.33 Use Case Scenario Mengubah Time Scheduling

Use case scenario mengubah *time scheduling* menggambarkan alur kegiatan yang dilakukan PM untuk mengubah data *Time Scheduling* dari dalam sistem. Alur kerja tersebut ditunjukkan pada Tabel 4.37.

Tabel 4.37 Use case scenario mengubah *time scheduling*

Mengubah Time Scheduling (SMP_F_033)	
Objective	PM dapat mengubah data <i>Time Scheduling</i> dari dalam sistem.
Actor	PM 1
Pre-condition	PM berada pada halaman <i>Time Scheduling</i> .
Main Flow	<ol style="list-style-type: none"> 1. PM menekan tombol “edit” pada data yang dipilih 2. Sistem menampilkan halaman ubah <i>time scheduling</i> dari data yang dipilih. 3. PM mengubah data <i>time scheduling</i> yang telah dipilih dan menekan tombol “simpan”. 4. Sistem mengubah data <i>time scheduling</i> yang dipilih dari dalam sistem. 1 5. Sistem menampilkan halaman <i>time scheduling</i>.
Alternative Flow	-
Post Condition	Kembali ke halaman <i>Time Scheduling</i> .

4.5.2.34 Use Case Scenario Menghapus Time Scheduling

Use case scenario menghapus *time scheduling* menggambarkan alur kegiatan yang dilakukan PM untuk menghapus data *Time Scheduling* dari dalam sistem. Alur kerja tersebut ditunjukkan pada Tabel 4.38.

Tabel 4.38 Use case scenario menghapus time scheduling

Menghapus Time Scheduling (SMP_F_034)	
Objective	PM dapat menghapus data <i>Time Scheduling</i> dari dalam sistem.
Actor	PM
Pre-condition	PM berada pada halaman <i>time scheduling</i> .
Main Flow	<ol style="list-style-type: none">1. PM menekan tombol “delete” pada data yang dipilih.2. Sistem menghapus data <i>time scheduling</i> dari dalam sistem.3. Sistem menampilkan halaman <i>time scheduling</i>.
Alternative Flow	-
Post Condition	Kembali ke halaman <i>time scheduling</i> .

BAB 5 PERANCANGAN DAN IMPLEMENTASI

5.1 Perancangan

Pada tahap perancangan dilakukan berdasarkan hasil dari rekayasa kebutuhan yang telah dilakukan sebelumnya. Perancangan terdiri dari perancangan alur proses program dengan menggunakan *sequence diagram*, pemodelan *class diagram*, perancangan komponen dan perancangan antarmuka.

5.1.1 Sequence Diagram

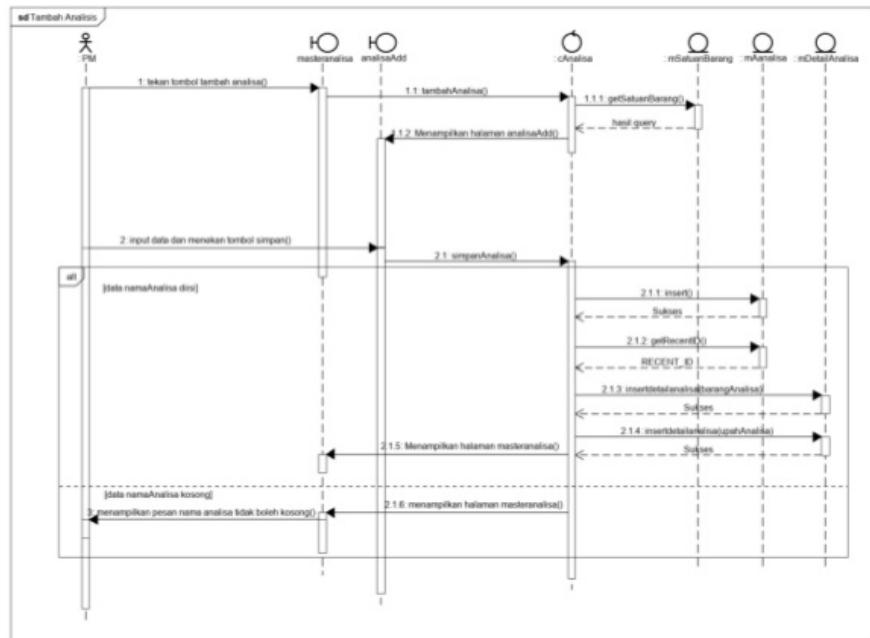
Sequence diagram adalah perancangan perangkat lunak yang digunakan untuk menggambarkan alur proses yang terjadi saat aktor berinteraksi dengan sistem. Yang akan dijelaskan akan meliputi *sequence diagram* tambah analisis, *sequence diagram* mengubah analisis dan *sequence diagram* menghapus analisis.

5.1.1.1 Sequence Diagram Tambah Analisis

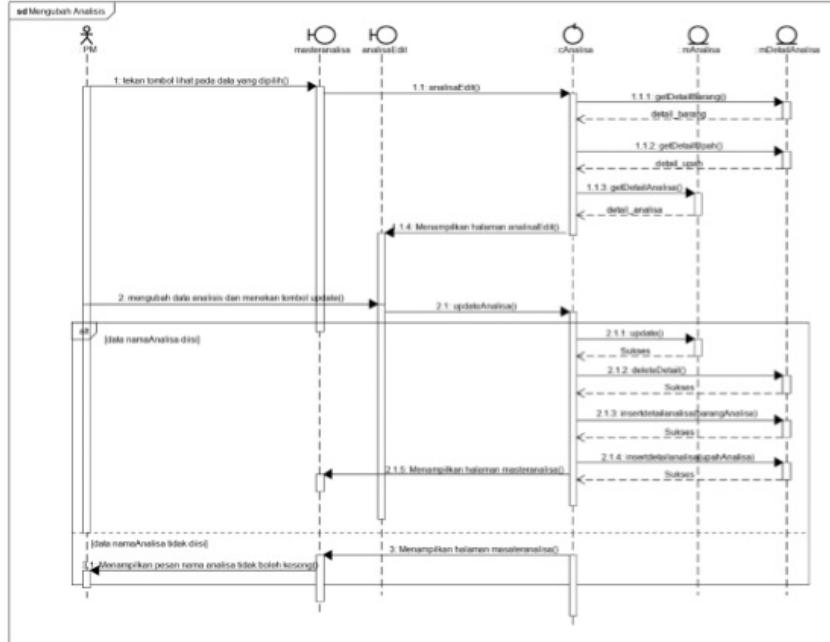
Pada gambar 5.1 yang merupakan *sequence diagram* tambah analisis terdapat aktor PM, *view* masteranalisa, *view* analisaAdd, *controller* analisa, *model* mAnalisa dan *model* mSatuanBarang. Proses yang terjadi pada *sequence diagram* tambah analisis adalah PM menekan tombol tambah analisa, selanjutnya *controller* akan menjalankan method tambahAnalisa(), lalu *controller* akan memanggil method getSatuanBarang() dari *model* mSatuanBarang, lalu mSatuanBarang akan memberikan hasil query ke *controller*, setelah itu *controller* akan menampilkan *view* analisaAdd kepada PM. Lalu PM akan memasukkan data dan menekan tombol simpan, lalu *controller* analisa akan menjalankan method simpanAnalisa(), lalu *model* akan menjalankan method insert(), getRecentID() dan insertAnalisa(), setelah itu data berhasil ditambahkan. Dan *controller* analisa akan menampilkan *view* masteranalisa.

5.1.1.2 Sequence Diagram Mengubah Analisis

Pada gambar 5.2 yang merupakan *sequence diagram* mengubah analisis terdapat aktor PM, *view* masteranalisa, *view* analisaEdit, *controller* analisa, dan *model* mAnalisa. PM menekan tombol lihat pada data yang akan diubah, lalu *controller* analisa akan menjalankan method analisaEdit(), lalu *model* mAnalisa akan menjalankan method getDataBarang(), getDataUpah() dan getDataAnalisa(). Lalu *model* mAnalisa akan memberikan hasil query kepada *controller* analisa, dan *controller* analisa akan menampilkan *view* analisaEdit kepada PM. PM akan mengubah data dan menekan tombol update pada *view* analisaEdit, lalu *controller* analisa akan menjalankan method updateAnalisa(). Lalu *model* mAnalisa akan menjalankan method update(), deleteDetail(), dan insertAnalisa(). Maka data yang telah dipilih telah berhasil diubah. Dan *controller* analisa akan menampilkan *view* masteranalisa.



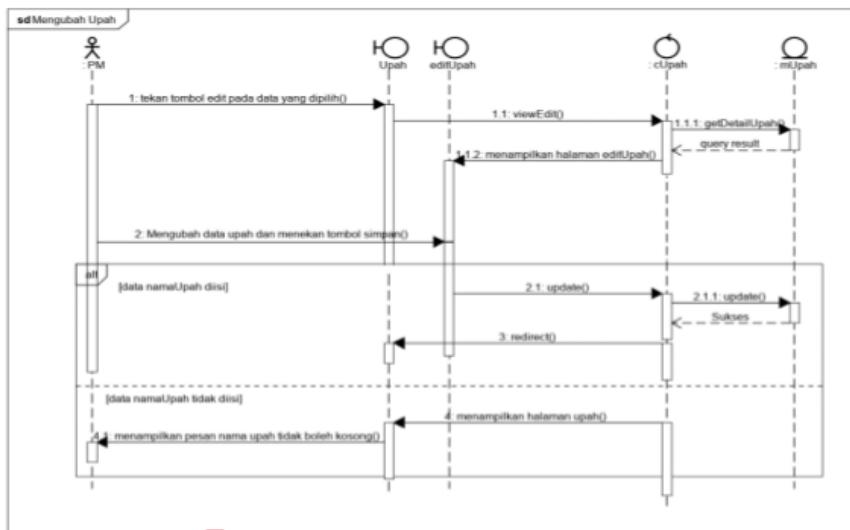
1
Gambar 5.1 Sequence Diagram Tambah Analisis



Gambar 5.2 Sequence Diagram Mengubah Analisis

5.1.1.3 Sequence Diagram Mengubah Upah

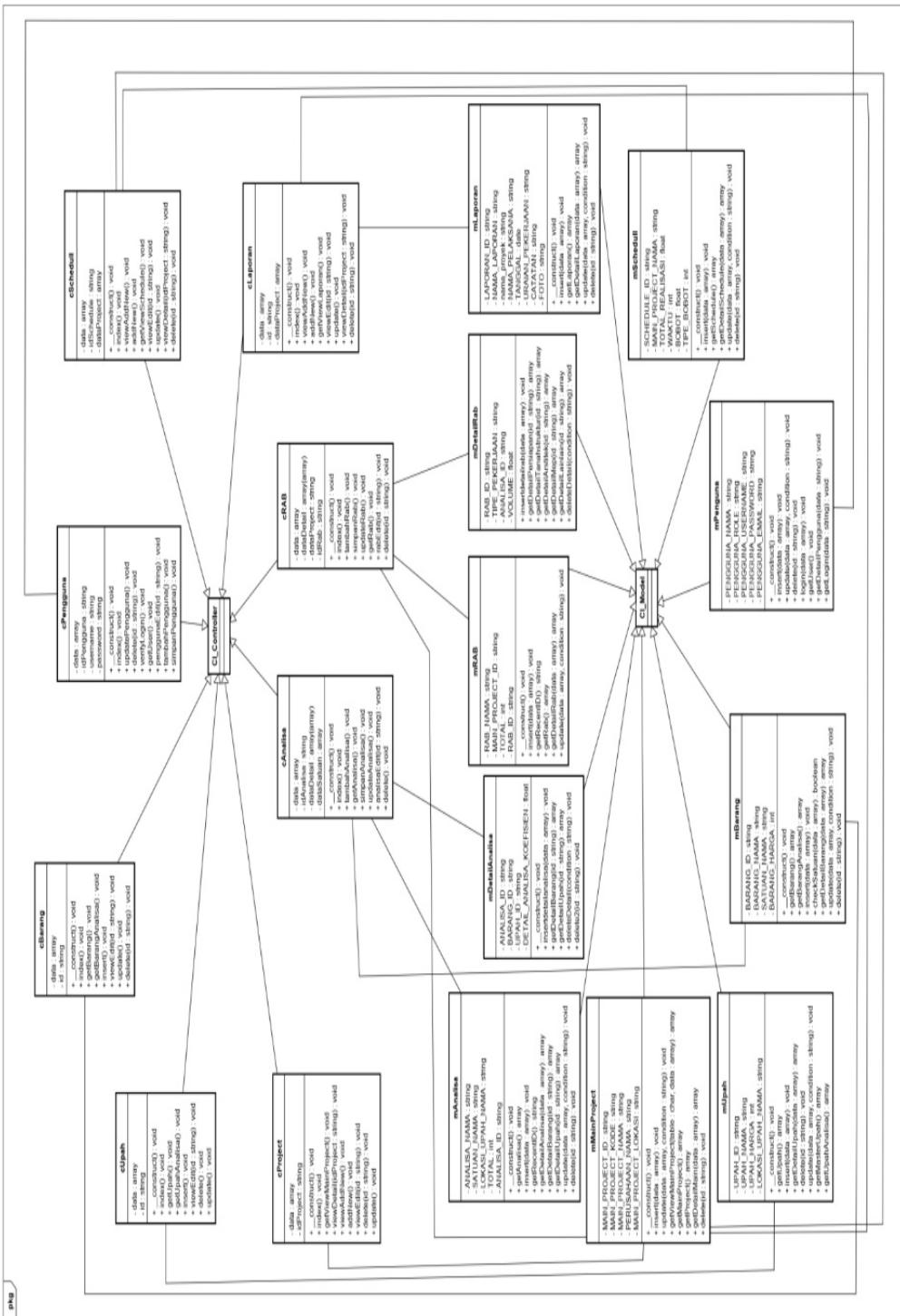
Pada gambar 5.3 yang merupakan *sequence diagram* menghapus analisis terdapat aktor PM, *view upah*, *view editUpah*, *controller upah*, dan *model mUpah*. PM menekan tombol *edit* pada data yang dipilih, lalu *controller upah* akan menjalankan method *viewEdit()*. Dan *model mUpah* akan menjalankan method *getDetailUpah()* dan memberikan hasil *query* ke *controller*. *Controller upah* akan menampilkan halaman *editUpah* kepada PM. PM mengubah data upah dan menekan tombol *simpan*. *Controller upah* akan menjalankan method *update()* lalu *model mUpah* akan menjalankan method *update()*. Data telah berhasil dilakukan perubahan. *Controller upah* akan me-*redirect* ke *view upah*.



1
Gambar 5.3 Sequence Diagram Menghapus Upah

5.1.2 Class Diagram

Class diagram digunakan untuk menggambarkan objek-objek yang terdapat di dalam sistem dan relasi antar objek-objek tersebut untuk membangun sebuah sistem. *Class diagram* sistem manajemen proyek dapat dilihat pada gambar 5.4.



Gambar 5.4 Class Diagram Sistem Manajemen Proyek

1

5.1.3 Perancangan Komponen

Perancangan komponen digunakan untuk menggambarkan komponen secara detail pada sistem. Perancangan komponen harus menjelaskan struktur komponen pada semua objek dan logika yang terdapat di dalam sistem. Perancangan komponen menjelaskan algoritma berdasarkan *sequence diagram* dan *class diagram* yang telah dibuat.

1

5.1.3.1 Perancangan Komponen Method simpanAnalisa() Klas cAnalisa

Method `simpanAnalisa()` digunakan untuk melakukan penambahan data analisis yang dilakukan oleh PM. `simpanAnalisa()` berisi data nama analisis, satuan, lokasi, uraian barang, uraian upah dan total. Perancangan komponen method `simpanAnalisa()` dijelaskan pada tabel 5.1.

Tabel 5.1 Pseudocode method simpanAnalisa() klas cAnalisa

Nama Method	simpanAnalisa(\$data, \$barang, \$upah)
No	Pseudocode
1	Start
2	Get data simpananalisa[]
3	Set array data
4	If namaAnalisa = empty then
5	Output = "gagal"
6	Else Set array data = barangAnalisa
7	Set array data = upahAnalisa
10	Output "success"
11	End if
12	End

1

5.1.3.2 Perancangan Komponen Method insert() Klas mAnalisa

Method `insert()` digunakan untuk melakukan penambahan data analisis ke dalam sistem yang dilakukan oleh PM. Perancangan komponen method `insert()` dijelaskan pada tabel 5.2.

Tabel 5.2 Pseudocode method insert() Klas mAnalisa

Nama Method	insert(\$data)
No	Pseudocode
1	Start
2	Get data insert[]
3	Set table = "master_analisa"
4	Set array data = data
5	Output "success"

6	End
---	-----

5.1.3.3 Perancangan Komponen Fungsi update()

Fungsi `update()` digunakan untuk melakukan pengubahan data upah yang dilakukan oleh PM. Mengubah data upah berisi data nama upah, harga dan lokasi. Perancangan komponen fungsi `update()` dijelaskan pada tabel 5.3.

Tabel 5.3 Pseudocode method update()

Nama Method	update(\$data)
1	
No	Pseudocode
1	Start
2	Get data update[]
3	Set array data = data[]
4	If nama = empty then
5	Output "gagal"
6	Else Output "success"
8	End if
9	End

1

5.1.4 Perancangan Antarmuka

Perancangan antarmuka adalah menggambarkan rancangan dari tampilan sistem yang akan digunakan oleh pengguna. Perancangan antarmuka digunakan untuk mengukur tingkat kemudahan pengguna dalam menggunakan perangkat lunak.

1

5.1.4.1 Perancangan Antarmuka Lihat Analisis

Antarmuka halaman lihat analisis adalah halaman untuk melihat data daftar analisis yang telah dibuat. Halaman ini dapat dilihat oleh aktor PM. Perancangan antarmuka halaman lihat analisis digambarkan pada Gambar 5.5.

Gambar 5.5 Perancangan Antarmuka Lihat Analisis

5.1.4.2 Perancangan Antarmuka Tambah / Edit Analisis

Antarmuka halaman tambah atau *edit* analisis adalah halaman untuk menambahkan data analisis baru atau mengubah data analisis yang telah ada. Halaman ini dapat dilihat oleh aktor PM. Perancangan antarmuka halaman tambah atau *edit* analisis digambarkan pada Gambar 5.6.

Gambar 5.6 Perancangan Antarmuka Tambah/ Edit Analisis

5.1.4.3 Perancangan Antarmuka Lihat Barang

Antarmuka halaman lihat barang adalah halaman untuk melihat daftar data barang yang telah ada atau dibuat. Halaman ini dapat dilihat oleh aktor PM dan Purchasing. Perancangan antarmuka halaman lihat barang digambarkan pada Gambar 5.7.

Gambar 5.7 Perancangan Antarmuka Lihat Barang

5.1.4.4 Perancangan Antarmuka Lihat Laporan

Antarmuka halaman lihat laporan adalah halaman untuk melihat daftar data laporan yang telah ada atau dibuat. Halaman ini dapat dilihat oleh aktor PM dan Pelaksana. Perancangan antarmuka halaman lihat laporan digambarkan pada Gambar 5.8.

Gambar 5.8 Perancangan Antarmuka Lihat Laporan

5.1.4.5 Perancangan Antarmuka Tambah/ Edit Laporan

Antarmuka halaman tambah atau edit laporan adalah halaman untuk menambahkan data laporan baru atau mengubah data laporan yang telah ada. Halaman ini dapat dilihat oleh aktor Pelaksana. Perancangan antarmuka halaman tambah atau edit laporan digambarkan pada Gambar 5.9.

Gambar 5.9 Perancangan Antarmuka Tambah/ Edit Laporan

5.1.4.6 Perancangan Antarmuka Lihat Proyek

Antarmuka halaman lihat proyek adalah halaman untuk melihat daftar data proyek yang telah ada atau dibuat. Halaman ini dapat dilihat oleh aktor PM dan Direktur. Perancangan antarmuka halaman lihat proyek digambarkan pada Gambar 5.10.

Gambar 5.10 Perancangan Antarmuka Lihat Proyek

1

5.1.4.7 Perancangan Antarmuka Lihat RAB

Antarmuka halaman lihat RAB adalah halaman untuk melihat daftar data RAB yang telah ada atau dibuat. Halaman ini dapat dilihat oleh aktor PM dan Direktur. Perancangan antarmuka halaman lihat RAB digambarkan pada Gambar 5.11.

Gambar 5.11 Perancangan Antarmuka Lihat RAB

5.1.4.8 Perancangan Antarmuka Tambah/ Edit RAB

Antarmuka halaman tambah atau edit RAB adalah halaman untuk menambahkan data RAB baru atau mengubah data RAB yang telah ada. Halaman ini dapat dilihat oleh aktor PM. Perancangan antarmuka halaman tambah atau edit RAB digambarkan pada Gambar 5.12.

Gambar 5.12 Perancangan Antarmuka Tambah/ Edit RAB

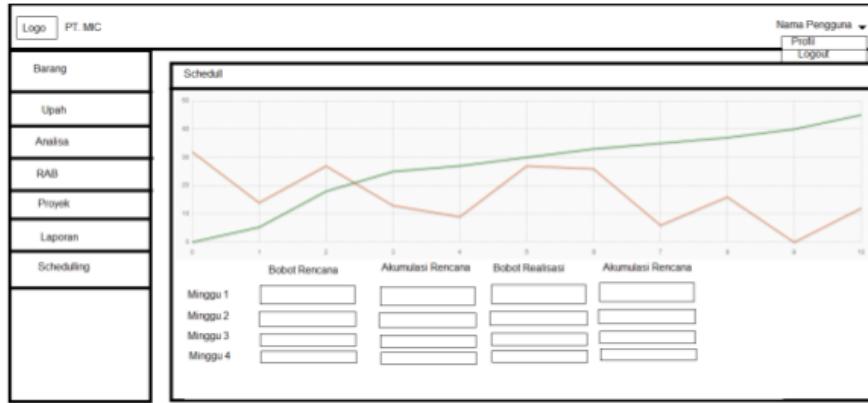
5.1.4.9 Perancangan Antarmuka Lihat Schedule

Antarmuka halaman lihat schedule adalah halaman untuk melihat daftar data schedule yang telah ada atau dibuat. Halaman ini dapat dilihat oleh aktor PM dan Direktur. Perancangan antarmuka halaman lihat schedule digambarkan pada Gambar 5.13.

Gambar 5.13 Perancangan Antarmuka Lihat Schedule

5.1.4.10 Perancangan Antarmuka Tambah atau *Edit Schedule*

Antarmuka halaman tambah atau *edit schedule* adalah halaman untuk menambahkan atau mengubah data *schedule*. Halaman ini dapat dilihat oleh aktor PM. Perancangan antarmuka halaman tambah atau *edit schedule* digambarkan pada Gambar 5.14.



Gambar 5.14 Perancangan Antarmuka Tambah atau *Edit Schedule*

5.1.4.11 Perancangan Antarmuka Lihat Upah

Antarmuka halaman lihat upah adalah halaman untuk melihat daftar data upah yang telah ada atau dibuat. Halaman ini dapat dilihat oleh aktor PM. Perancangan antarmuka halaman lihat upah digambarkan pada Gambar 5.15.

Gambar 5.15 Perancangan Antarmuka Lihat Upah

5.2 Implementasi Sistem

Pada tahap implementasi sistem adalah kelanjutan dari tahap yang sebelumnya telah dilakukan, yaitu tahap rekayasa kebutuhan dan perancangan. Pada tahap ini akan dibahas mengenai spesifikasi sistem, implementasi kode program dan implementasi antarmuka.

5.2.1 Spesifikasi Sistem

Spesifikasi sistem menjelaskan tentang perangkat atau peralatan apa yang kita gunakan untuk melakukan implementasi sistem. Spesifikasi sistem terdapat dua bagian yaitu spesifikasi sistem perangkat keras dan spesifikasi sistem perangkat lunak.

5.2.1.1 Spesifikasi Perangkat Keras

Spesifikasi perangkat keras menjelaskan tentang peralatan perangkat keras yang digunakan untuk melakukan implementasi sistem. Spesifikasi perangkat keras dapat dilihat pada tabel 5.4.

Tabel 5.4 Spesifikasi Perangkat Keras

Nama Komponen	Spesifikasi
System Model	Aspire A315-41
Processor	AMD Ryzen 3 2200U
Memory	8GB
Storage	1TB
Graphics Card	Radeon Vega Mobile Gfx

5.2.1.2 Spesifikasi Perangkat Lunak

Spesifikasi perangkat lunak menjelaskan tentang peralatan perangkat lunak yang digunakan untuk melakukan implementasi sistem. Spesifikasi perangkat lunak dapat dilihat pada tabel 5.5.

Tabel 5.5 Spesifikasi Perangkat Lunak

Nama Komponen	Spesifikasi
Sistem Operasi	Windows 10 Home 64 Bit
Editor Perancangan	Astah UML 8.1.0, Microsoft Paint
Editor Pemrograman	Sublime Text 3
Framework dan Bahasa Pemrograman	Codeigniter, Bootstrap, Datatable, charts.js, Handsontable, fontawesome, PHP, HTML, CSS dan JS
DBMS	MySQL
Database Server	Phpmyadmin
Aplikasi Pendukung	XAMPP
Browser	Google Chrome Desktop versi 77.0.3865.120 dan Mozilla Firefox Desktop versi 68.2.0

5.2.2 Batasan Implementasi

Terdapat beberapa batasan dalam proses implementasi yaitu:

1. Sistem dibangun menggunakan bahasa pemrograman PHP dan JavaScript yang diimplementasikan pada framework CodeIgniter.
2. Sistem menggunakan basis data MySQL.
3. Pembuatan anatarmuka sistem menggunakan HTML, CSS , dan Bootstrap.

5.2.3 Implementasi Kode Program

Implementasi kode program dibuat berdasarkan perancangan komponen yang telah dilakukan sebelumnya. Algoritme yang dibuat dalam bentuk pseudocode akan diubah ke bentuk bahasa pemrograman. Algoritme yang akan ditulis implementasinya yaitu tambah analisis, mengubah analisis dan mengubah upah.

5.2.3.1 Implementasi Kode Program Method simpanAnalisa()

Method simpanAnalisa() adalah Method yang digunakan untuk melakukan penambahan data analisis ke dalam database. Method ini digunakan oleh aktor PM. Kode program Method simpanAnalisa() dijelaskan pada tabel 5.6.

Tabel 5.6 Implementasi Kode Program Method simpanAnalisa()

No	Kode Program
1	<pre> 1 public function simpanAnalisa() { 2 \$namaAnalisa = \$this->input->post('nama'); 3 \$satuanAnalisa = \$this->input->post('satuan'); 4 \$lokasiAnalisa = \$this->input->post('lokasi'); 5 \$barangAnalisa=json_decode(\$this->input->post('barang')); 6 \$upahAnalisa = json_decode(\$this->input->post('upah')); 7 \$total = \$this->input->post('total'); 8 \$this->form_validation->set_rules('nama','Nama analisa','required'); 9 \$data = array(10 'ANALISA_NAMA' => \$namaAnalisa, 11 'SATUAN_NAMA' => \$satuanAnalisa, 12 'LOKASI_UPAH_NAMA' => \$lokasiAnalisa, 13 'TOTAL' => \$total 14); 15 if (\$this->form_validation->run() == FALSE) { 16 \$this->load->view('analisaAdd.php'); 17 } 18 else { 19 \$this->mAnalisa->insert(\$data); 20 \$idAnalisa = \$this->mAnalisa->getRecentID(); 21 foreach (\$barangAnalisa as \$item) { 22 \$data = array(23 'BARANG_ID' => \$item->id, 24 'ANALISA_ID' => \$idAnalisa, 25 'DETAIL_ANALISA_KOEFISIEN' => \$item->koef 26); 27 \$this->mAnalisa->insertdetailanalisa(\$data); 28 } 29 foreach (\$upahAnalisa as \$item) { 30 \$data = array(31 'UPAH_ID' => \$item->id, 32 'ANALISA_ID' => \$idAnalisa, 33 'DETAIL_ANALISA_KOEFISIEN' => \$item->koef 34); 35 \$this->mAnalisa->insertdetailanalisa(\$data); 36 } 37 } 38 echo "success"; 39 } 40 }</pre>

Pada baris ke-1 deklarasi nama fungsi simpanAnalisa(). Baris ke-2 sampai ke-7 inisialisasi data variable namaAnalisa, satuanAnalisa, lokasiAnalisa, barangAnalisa, upahAnalisa dan total. Baris ke-8 sampai ke-9 inisialisasi rules form validation. Baris ke-10 sampai ke-15 memasukkan data namaAnalisa, satuanAnalisa, lokasiAnalisa, dan total ke dalam array data. Baris ke-16 sampai ke-18 melakukan validasi form dan jika salah akan kembali ke halaman tambah analisa. Baris ke-21 memanggil method insert() pada kelas mAnalisa untuk melakukan penyimpanan data array. Baris ke-22 memanggil method getRecentID() yang memberikan data dan disimpan ke dalam variable idAnalisa. Baris ke-22 sampai ke-37 memanggil method insertdetailanalisa() untuk melakukan penyimpanan data array. Baris ke-38

melakukan cetak data "success" yang digunakan untuk kembali ke halaman daftar analisa.

5.2.3.2 Implementasi Kode Program Method updateAnalisa()

Method updateAnalisa() adalah Method yang digunakan untuk melakukan perubahan data analisis dari dalam database. Method ini digunakan oleh aktor PM. Kode program Method updateAnalisa() dijelaskan pada tabel 5.7.

Tabel 5.7 Implementasi Kode Program Method updateAnalisa()

No	Kode Program
1	public function updateAnalisa () {
2	\$IdAnalisa = \$this->input->post('id');
3	\$namaAnalisa = \$this->input->post('nama');
4	\$satuanAnalisa = \$this->input->post('satuan');
5	\$lokasiAnalisa = \$this->input->post('lokasi');
6	\$barangAnalisa=json_decode(\$this->input->post('barang'));
7	\$upahAnalisa = json_decode(\$this->input->post('upah'));
8	\$total = \$this->input->post('total');
9	\$condition = "ANALISA_ID = " . "!" . \$idAnalisa. """;
10	\$this->form_validation->set_rules('nama','Nama analisa','required');
11	\$data = array(
12	'ANALISA_NAMA' => \$namaAnalisa,
13	'SATUAN_NAMA' => \$satuanAnalisa,
14	'LOKASI_UPAH_NAMA' => \$lokasiAnalisa,
15	'TOTAL' => \$total
16);
17	if (\$this->form_validation->run() == FALSE) {
18	\$this->load->view('analisaEdit.php');
19	}
20	else {
21	\$this->mAnalisa->update(\$data, \$condition);
22	\$this->mAnalisa->deleteDetail(\$condition);
23	foreach (\$barangAnalisa as \$item) {
24	\$data = array(
25	'BARANG_ID' => \$item->id,
26	'ANALISA_ID' => \$idAnalisa,
27	'DETAIL_ANALISA_KOEFISIEN' => \$item->koef
28);
29	\$this->mAnalisa->insertdetailanalisa(\$data);
30	}
31	foreach (\$upahAnalisa as \$item) {
32	\$data = array(
33	'UPAH_ID' => \$item->id,
34	'ANALISA_ID' => \$idAnalisa,
35	'DETAIL_ANALISA_KOEFISIEN' => \$item->koef
36);
37	\$this->mAnalisa->insertdetailanalisa(\$data);
38	}
39	echo "success";
40	}
41	}
42	}

Pada baris ke-1 deklarasi nama fungsi updateAnalisa(). Baris ke-2 sampai ke-9 inisialisasi data variable namaAnalisa, satuanAnalisa, lokasiAnalisa, barangAnalisa, upahAnalisa, condition dan total. Baris ke-10 sampai ke-

11 inisialisasi *rules form validation*. Baris ke-12 sampai ke-17 memasukkan data namaAnalisa, satuanAnalisa, lokasiAnalisa, dan total ke dalam array data. Baris ke-18 sampai ke-20 melakukan validasi *form* dan jika salah akan kembali ke halaman *update* analisa. Baris ke-22 memanggil *method update()* pada kelas mAnalisa untuk melakukan penyimpanan data array. Baris ke-23 memanggil *method deleteDetail()* untuk melakukan penghapusan detail analisa sesuai dengan *condition*. Baris ke-24 sampai ke-39 memanggil *method insertdetailanalisa()* untuk melakukan penyimpanan data array. Baris ke-40 melakukan cetak data "success" yang digunakan untuk kembali ke halaman daftar analisa.

5.2.3.3 Implementasi Kode Program *Method update()*

Method update() adalah *Method* yang digunakan untuk melakukan perubahan data upah dari dalam *database*. *Method* ini digunakan oleh aktor PM. Kode program *Method update()* dijelaskan pada tabel 5.8.

Tabel 5.8 Implementasi Kode Program *Method update()*

No	Kode Program
1	public function update() {
2	\$idUpah = \$this->input->post('UPAH_ID');
3	\$namaUpah = \$this->input->post('UPAH_NAMA');
4	\$hargaUpah = \$this->input->post('UPAH_HARGA');
5	\$lokasiUpahId=\$this->input->post('LOKASI_UPAH_ID');
6	\$condition = "ANALISA_ID = " . "" . \$idAnalisa. " ";
7	\$this->form_validation->set_rules('nama','Nama
8	upah','required');
9	\$data = array(
10	'UPAH_NAMA' => \$this->input->post('UPAH_NAMA'),
11	'UPAH_HARGA'=>\$this->input->post('UPAH_HARGA'),
12	'LOKASI_UPAH_ID'=>\$this->input->post('LOKASI_UPAH_ID')
13);
14	if (\$this->form_validation->run() == FALSE) {
15	\$this->load->view('editUpah.php');
16	}
17	else {
18	\$this->mUpah->update(\$data, \$condition);
19	redirect(base_url() . upah);
20	}
21	}

Pada baris ke-1 deklarasi nama fungsi *update ()*. Baris ke-2 sampai ke-6 inisialisasi data variable *idUpah*, *namaUpah*, *hargaUpah*, *lokasiUpahId*, dan *condition*. Baris ke-7 sampai ke-8 inisialisasi *rules form validation*. Baris ke-9 sampai ke-13 memasukkan data *idUpah*, *namaUpah*, *hargaUpah*, *lokasiUpahId*, dan *condition* ke dalam array *data*. Baris ke-14 sampai ke-16 melakukan validasi *form* dan jika salah akan kembali ke halaman *update* upah. Baris ke-18 memanggil *method update()* pada kelas *mUpah* untuk melakukan penyimpanan data array. Baris ke-19 melakukan *redirect* ke halaman daftar upah.

5.2.4 Implementasi Antarmuka

Implementasi antarmuka merupakan hasil dari perancangan antarmuka yang telah dibuat sebelumnya. Halaman antarmuka yang akan dijadikan sampel berjumlah 6 antarmuka. Antarmuka tersebut antara.

1

5.2.4.1 Implementasi Antarmuka Halaman Lihat Analisis

2

Antarmuka halaman daftar analisis digunakan untuk melihat data analisis yang telah tersimpan di dalam database. Halaman daftar analisis berupa tabel yang berisi data nama analisis, satuan, lokasi, harga dan detail. Dimana pada detail terdapat button untuk melakukan update atau hapus data analisis. Terdapat button tambah analisa untuk menjalankan fungsi tambah analisis. Terdapat fungsi search pada tabel. Antarmuka halaman lihat analisis dapat dilihat pada gambar 5.16.

Name	Satuan	Harga	Detail
1 Buah Jambu Bergigi	Pkg	300000	
1 Buah Durian Buluh	kg	100000	
1 Buah Pisang Jengkol	kg	30000	
1 Buah Nanas Asam	kg	10000	
1 Buah Nangka 12"-14"	kg	10000	
1 Buah Pisang Seris Kuning	kg	12000	
1 Buah Pisang Lempis Dl.	kg	8000	
1 Buah Pisang Tempe TU	kg	10000	
1 Buah Ubi	kg	10000	
1 Buah Wawasih	kg	20000	

Gambar 5.16 Implementasi Antarmuka Halaman Daftar Analisis

5.2.4.2 Implementasi Antarmuka Halaman Tambah atau Edit Analisis

Antarmuka halaman tambah atau edit analisis digunakan untuk menambahkan atau mengubah data analisis ke dalam database. Halaman tambah atau edit analisis berupa form dan spreadsheet yang harus diisi terdapat nama analisa, satuan, dan lokasi pada form. Dan pada spreadsheet pada bagian bawah kolom bahan dan upah dapat diklik sehingga muncul halaman pop-up yang berisi data bahan dan upah. Setelah itu kita akan memilih salah satu item pada pop-up tersebut setelah itu nama item yang dipilih akan muncul pada spreadsheet dan kita perlu mengisi nilai koefisien pada spreadsheet sehingga muncul nilai jumlah dan total. Antarmuka halaman tambah atau edit analisis dapat dilihat pada gambar 5.17.

Gambar 5.17 Implementasi Antarmuka Halaman Tambah atau *Edit Analisis*

5.2.4.3 Implementasi Antarmuka Halaman Lihat Schedule

Antarmuka halaman lihat *schedule* digunakan untuk melihat daftar *schedule* yang telah dibuat. Halaman lihat *schedule* berupa tabel yang berisi data nama *schedule*, nama proyek, nama pelaksana, tanggal dan detail. Dimana pada detail terdapat button untuk melakukan *update* atau hapus data *schedule*. Terdapat button tambah *schedule* untuk menjalankan fungsi tambah *schedule*. Terdapat fungsi search pada tabel. Antarmuka halaman lihat *schedule* dapat dilihat pada gambar 5.18.

Gambar 5.18 Implementasi Antarmuka Halaman Lihat *Schedule*

5.2.4.4 Implementasi Antarmuka Halaman Tambah atau *Edit Schedule*

Antarmuka halaman tambah atau *edit schedule* digunakan untuk melakukan menambah atau mengubah data *schedule*. Halaman tambah atau *edit schedule* berupa form dan grafik yang harus diisi terdapat bobot rencana dan bobot realisasi pada form. Antarmuka halaman *schedule* dapat dilihat pada gambar 5.19.



Gambar 5.19 Implementasi Antarmuka Halaman Schedule

5.2.4.5 Implementasi Antarmuka Halaman Lihat Laporan

Antarmuka halaman lihat laporan digunakan untuk melihat daftar laporan yang telah dibuat. Halaman lihat laporan berupa tabel yang berisi data nama laporan, nama proyek, nama pelaksana, tanggal dan detail. Dimana pada detail terdapat button untuk melakukan update atau hapus data laporan. Terdapat button tambah laporan untuk menjalankan fungsi tambah laporan. Terdapat fungsi search pada tabel. Antarmuka halaman lihat laporan dapat dilihat pada gambar 5.20.

The screenshot shows a 'Laporan' page. At the top, there's a navigation bar with the company logo 'PT Matahari Inti Cemerlang', the user role 'Administrator', and a link to 'Home / Laporan'. On the left, a sidebar menu lists 'Borang', 'Usaha', 'Analisa', 'RAB', 'Proyek', 'Laporan', and 'Scheduling', with 'Laporan' currently selected. The main area has a title 'Data Laporan' and a sub-section 'Tambah Laporan'. It displays a table titled 'Show 10 of entries' with columns: No, Nama Laporan, Nama Proyek, Nama Pelaksana, Tanggal, and Detail. A single row is shown with values: 1, tes, tes, tes, 2019-12-03, and a set of edit and delete buttons. Below the table, it says 'Showing 1 to 1 of 1 entries' and has 'Previous' and 'Next' buttons.

Gambar 5.20 Implementasi Antarmuka Halaman Lihat Laporan

5.2.4.6 Implementasi Antarmuka Halaman Tambah atau Edit Laporan

Antarmuka halaman tambah atau *Edit* laporan digunakan untuk menambah atau mengubah data laporan ke dalam *database*. Halaman tambah atau *Edit* laporan berupa form yang harus diisi terdapat nama laporan, nama proyek, nama pelaksana, tanggal, uraian pekerjaan, catatan dan upload foto pada form. Antarmuka halaman tambah laporan dapat dilihat pada gambar 5.21.

Gambar 5.21 Implementasi Antarmuka Halaman Tambah Laporan

5.2.4.7 Implementasi Antarmuka Halaman Lihat Barang

Antarmuka halaman lihat barang digunakan untuk melihat daftar barang yang telah dibuat. Halaman lihat barang berupa tabel yang berisi data aksi, nama barang, satuan, dan harga. Dimana pada aksi terdapat *button* untuk melakukan *update* atau hapus data barang. Terdapat *button* tambah laporan untuk menjalankan fungsi tambah barang. Terdapat fungsi *search* pada tabel. Antarmuka halaman lihat barang dapat dilihat pada gambar 5.22.

Aksi	Nama	Satuan	Harga
	Paku Pasang	M3	140000
	Batu kali	m3	150000
	Stenografi 1-2	m3	240000
	Stenografi 2-3	m3	250000
	Batu Marmer	m3	550
	Siklu corak	m3	75000
	Siklu Platit Doser	m3	80000
	Kayut Ayahan Pintu	m	40000
	Besi Beton Platuk da. Simes	m	23000
	Besi Beton Platuk da. Simes	m	37000

Gambar 5.22 Implementasi Antarmuka Halaman Lihat Barang

5.2.4.8 Implementasi Antarmuka Halaman Lihat Proyek

Antarmuka halaman lihat proyek digunakan untuk melihat daftar proyek yang telah dibuat. Halaman lihat proyek berupa tabel yang berisi data kode SPK, nama proyek, nama pemilik, lokasi dan detail. Dimana pada detail terdapat *button* untuk melakukan *update* atau hapus data proyek. Terdapat *button* tambah proyek untuk menjalankan fungsi tambah proyek. Terdapat fungsi *search* pada tabel. Antarmuka halaman lihat proyek dapat dilihat pada gambar 5.23.

No	Kode SPK	Nama Proyek	Nama Pemilik	Lokasi	Detail
1	TES88881	DOME	PT Mahabhi Infra Construeng	JAKARTA	[Detail] [Edit] [Delete]
2	TES88883	THEME PARK	PT Mahabhi Infra Construeng	JAKARTA	[Detail] [Edit] [Delete]
3	TES88884	GROSIR	PT Mahabhi Infra Construeng	JAKARTA	[Detail] [Edit] [Delete]
4	TES88885	HOTEL	PT Mahabhi Infra Construeng	JAKARTA	[Detail] [Edit] [Delete]
5	TES88886	CAROUSEL	PT Mahabhi Infra Construeng	JAKARTA	[Detail] [Edit] [Delete]
6	TES88887	OFFICE	PT Mahabhi Infra Construeng	JAKARTA	[Detail] [Edit] [Delete]
7	TES88888	Indo1	PT Mahabhi Infra Construeng	JAKARTA	[Detail] [Edit] [Delete]

Gambar 5.23 Implementasi Antarmuka Halaman Lihat Proyek

5.2.4.9 Implementasi Antarmuka Halaman Lihat RAB

Antarmuka halaman lihat rab digunakan untuk melihat daftar rab yang telah dibuat. Halaman lihat rab berupa tabel yang berisi data nama rab, nama pemilik proyek, lokasi, total harga, dan detail. Dimana pada detail terdapat *button* untuk melakukan *update* atau *hapus* data rab. Terdapat *button* tambah RAB untuk menjalankan fungsi tambah rab. Terdapat fungsi *search* pada tabel. Antarmuka halaman lihat RAB dapat dilihat pada gambar 5.24.

Nama RAB	Nama Pemilik Project	Lokasi	TOTAL	Detail
test1	PT Mahabhi Infra Construeng	JAKARTA	42200	[Detail] [Edit] [Delete]
test2	PT Mahabhi Infra Construeng	JAKARTA	1214200	[Detail] [Edit] [Delete]
test3	PT Mahabhi Infra Construeng	JAKARTA	1004200	[Detail] [Edit] [Delete]
test4	PT Mahabhi Infra Construeng	JAKARTA	1210000	[Detail] [Edit] [Delete]

Gambar 5.24 Implementasi Antarmuka Halaman Lihat RAB

5.2.4.10 Implementasi Antarmuka Halaman Tambah atau Edit RAB

Antarmuka halaman tambah atau *edit* rab digunakan untuk menambahkan atau mengubah data rab ke dalam *database*. Halaman tambah atau *edit* rab berupa form dan *spreadsheet* yang harus diisi terdapat nama rab, dan nama proyek pada form. Dan pada *spreadsheet* pada bagian bawah kolom pekerjaan persiapan, pekerjaan tanah dan struktur, pekerjaan arsitek, pekerjaan mekanikal, elektrikal, dan plumbing dan pekerjaan lain-lain dapat di klik sehingga muncul halaman *pop-up* yang berisi data analisa. Setelah itu kita akan memilih salah satu *item* pada *pop-up* tersebut setelah itu nama item yang dipilih akan muncul pada *spreadsheet* dan kita perlu mengisi nilai koefisien pada *spreadsheet* sehingga

muncul nilai jumlah dan total. Antarmuka halaman tambah atau *edit* analisis dapat dilihat pada gambar 5.25.

The screenshot shows a web-based application interface for managing RAB (Request for Quotation) data. On the left, there's a sidebar with navigation links: Analisa (selected), RAB, Proyek, Laporan, and Monitoring. The main area is titled 'Tambah Data RAB' (Add RAB Data). It contains a table with columns: Proyek, Volume, Satuan, Harga Satuan, and Jumlah. There are several rows of data entries, each with a green checkmark icon and a red delete icon. At the bottom right of the table is a green 'Simpan' (Save) button.

Gambar 5.25 Implementasi Antarmuka Halaman Tambah atau Edit RAB

5.2.4.11 Implementasi Antarmuka Halaman Lihat Upah

Antarmuka halaman lihat upah digunakan untuk melihat daftar upah yang telah dibuat. Halaman lihat upah berupa tabel yang berisi data aksi, nama upah, harga, dan lokasi. Dimana pada aksi terdapat button untuk melakukan update atau hapus data upah. Terdapat button tambah upah untuk menjalankan fungsi tambah upah. Terdapat fungsi search pada tabel. Antarmuka halaman lihat upah dapat dilihat pada gambar 5.26.

The screenshot shows a web-based application interface for managing wage data. On the left, there's a sidebar with navigation links: Analisa (selected), RAB, Proyek, Laporan, and Monitoring. The main area is titled 'Master Upah' (Upah Master). It contains a table with columns: Aksi, Nama, Harga, and Lokasi. The table lists 163 entries, showing names like 'Wanita', 'Tulang Ibu', 'Hewati Tulang Ibu', 'PENSIRUK', 'TUNANG GULU', 'TUNANG BATU', 'TUNANG KAYU', 'TUNANG CAT', 'KEMBALI TUNANGAN', and 'MANDOR GUL'. Each row has a green 'Edit' icon and a red 'Delete' icon. At the top right is a search bar. At the bottom, there are buttons for 'Previous' and 'Next' pages, and a total count of 'Showing 1 to 10 of 163 entries'.

Gambar 5.26 Implementasi Antarmuka Halaman Lihat Upah

BAB 6 PENGUJIAN

6.1 Pengujian Unit

Pada pengujian unit, pengujian perangkat lunak dilakukan di setiap unit yang ada pada sistem untuk dilakukan pengujian. Tujuan dari melakukan pengujian unit untuk memeriksa bagian kode di dalam sistem untuk memastikan tidak ada nya bug atau error. Method yang digunakan dalam melakukan pengujian unit yaitu method simpanAnalisa(), insert(), dan update().

6.1.1 Pengujian Unit Method simpanAnalisa() Klas cAnalisa

Method simpanAnalisa() digunakan untuk melakukan penambahan data analisis yang dilakukan oleh PM. Method ini terletak pada klas cAnalisa. pseudocode method simpanAnalisa() dapat dilihat pada tabel 6.1.

1. Pseudocode method simpanAnalisa()

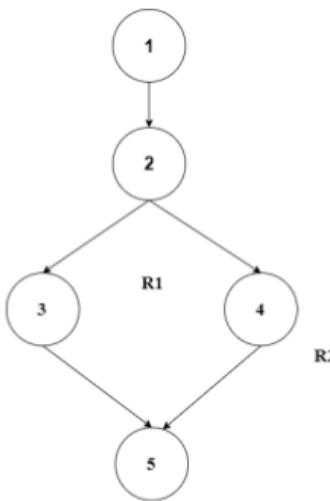
Tabel 6.1 Pseudocode method simpanAnalisa() Klas cAnalisa

Nama Method	simpanAnalisa(\$data,\$barang, \$upah)	
No	Pseudocode	Node
1	Start	1
2	Get data simpananalisa[]	
3	Set array data[]	
4	If namaAnalisa = empty then	2
5	Output = "gagal"	3
6	Else Set array data = barangAnalisa	4
7	Set array data = upahAnalisa	
10	Output "success"	
11	End if	5
12	End	
1		

2. Basis Path Testing

a. Flow Graph

Flow Graph method simpanAnalisa() dapat dilihat pada gambar 6.1.



Gambar 6.1 Flow Graph Method simpanAnalisa()

1 b. Cyclomatic Complexity

- $V(G) = \text{jumlah region} = 2$
- $V(G) = (\text{jumlah edge} - \text{jumlah node}) + 2 = (5 - 5) + 2 = 2$
- $V(G) = \text{jumlah predicate node} + 1 = 1 + 1 = 2$

c. Independent Path

- Jalur 1 = 1-2-3-5
- Jalur 2 = 1-2-4-5

3. Test Case

Berdasarkan hasil dari *flow graph* yang telah dilakukan sebelumnya, di dapatkan dua jalur independen yang menjadi kasus uji untuk *method* simpanAnalisa(). Hasil dari *test case* *method* simpanAnalisa() dapat dilihat pada tabel 6.2.

Tabel 6.2 Hasil Test Case method simpanAnalisa()

No. Jalu r	Prosedur Uji	Expected Result	Result	Status
1	Klas <i>driver</i> memanggil <i>method</i> simpanAnalisa() dengan parameter <code>\$data['ANALISA_NAMA']='', 'SATUAN_NAMA']='cm', 'LOKASI_UPAH_NAMA'= 'MALANG', 'TOTAL'= '123456'],</code>	Nama analisa kosong. Sistem menampilkan pesan "gagal"	Nama analisa kosong. Sistem menampilkan pesan "gagal"	valid

	<pre>\$barang['BARANG_ID'='1', 'ANALISA_ID'='200', 'DETAIL_ANALISA_KOEFISIEN = '1'], \$upah['UPAH_ID'='1', 'ANALISA_ID'='200', 'DETAIL_ANALISA_KOEFISIEN = '1']</pre>			
2	<p>Klas <i>driver</i> memanggil <i>method</i> <i>simpanAnalisa()</i> dengan parameter</p> <pre>\$data['ANALISA_NAMA'='analisa', 'SATUAN_NAMA'='cm', 'LOKASI_UPAH_NAMA'='MALANG', 'TOTAL'='123456'], \$barang['BARANG_ID'='1', 'ANALISA_ID'='200', 'DETAIL_ANALISA_KOEFISIEN = '1'], \$upah['UPAH_ID'='1', 'ANALISA_ID'='200', 'DETAIL_ANALISA_KOEFISIEN = '1']</pre>	<p>Analisa berhasil ditambahkan, Sistem menampilkan pesan "success"</p>	<p>Analisa berhasil ditambahkan, Sistem menampilkan pesan "success"</p>	<i>valid</i>

1

6.1.2 Pengujian Unit *Method insert () Klas mAnalisa*

Method insert () digunakan untuk melakukan penambahan data analisis ke dalam sistem yang dilakukan oleh PM. *Method* ini berada pada klas *mAnalisa*. *Pseudocode* *method insert ()* dapat dilihat pada tabel 6.3.

1. *Pseudocode method updateAnalisa()*

Tabel 6.3 Pseudocode method insert () klas mAnalisa

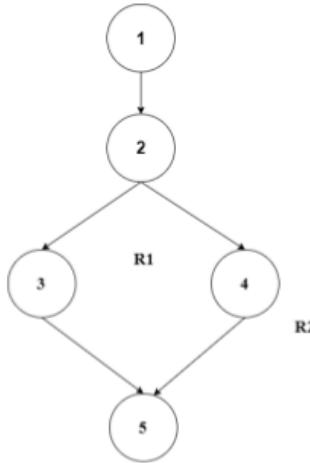
Nama Method		insert (\$data)	
No	Pseudocode		Node
1	Start		1
2	Get data insert[]		
3	Set table = "master_analisa"		
4	Set array data = data[]		
5	Set ql = get data analisa		
6	If \$ql->num_rows() > 0 then		
7	Output "gagal"		
8	Else Call method insert dengan array data		
9	Output "success"		

10	End If	5
11	End	

2. Basis Path Testing

a. Flow Graph

Flow Graph method insert() dapat dilihat pada gambar 6.2.



Gambar 6.2 Flow Graph Method insert()

b. Cyclomatic Complexity

- $V(G) = \text{jumlah region} = 2$
- $V(G) = (\text{jumlah edge} - \text{jumlah node}) + 2 = (5 - 5) + 2 = 5$
- $V(G) = \text{jumlah predicate node} + 1 = 0 + 1 = 1$

c. Independent Path

- Jalur 1 = 1-2-3-5
- Jalur 2 = 1-2-4-5

3. Test Case

Berdasarkan hasil dari flow graph yang telah dilakukan sebelumnya, di dapatkan dua jalur independen yang menjadi kasus uji untuk method insert(). Hasil dari test case method insert() dapat dilihat pada tabel 6.4.

Tabel 6.4 Hasil Test Case method insert()

No. Jalur	Prosedur Uji	Expected Result	Result	Status
1	Klas driver memanggil method insert() dengan parameter \$data['ANALISA_NAMA']=	Nama analisa telah ada di dalam sistem, Sistem	Nama analisa telah ada di dalam sistem,	valid

	'nama', 'SATUAN_NAMA'= 'cm', - 'LOKASI_UPAH_NAMA'= 'MALANG', 'TOTAL'= '123456']	menampilkan pesan "gagal"	Sistem menampilkan pesan "gagal"	
2	Klas <i>driver</i> memanggil <i>method insert()</i> dengan parameter \$data['ANALISA_NAMA']= 'nama analisa', 'SATUAN_NAMA'='cm', 'LOKASI_UPAH_NAMA'= 'MALANG', 'TOTAL'= '123456']	Analisa berhasil ditambahkan, Sistem menampilkan pesan "success"	Analisa berhasil ditambahkan, , Sistem menampilkan pesan "success"	valid

6.1.3 Pengujian Unit *Method update ()*

Method update () digunakan untuk melakukan perubahan data upah yang dilakukan oleh PM. *Pseudocode method update ()* dapat dilihat pada tabel 6.5.

1. *Pseudocode method update ()*

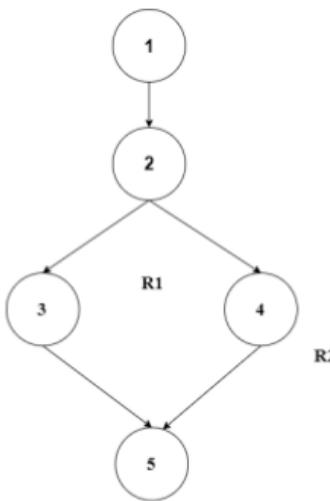
Table 6.5 *Pseudocode method update ()*

Nama Method		update(\$data)
No	Pseudocode	Node
1	Start	1
2	Get data update[]	
3	Set array data = data[]	
4	If nama = empty then	2
5	Output "gagal"	3
6	Else Output "success"	4
8	End if	5
9	End	

2. Basis Path Testing

a. *Flow Graph*

Flow Graph method update () dapat dilihat pada gambar 6.3.



Gambar 6.3 Flow Graph Method `update()`

1

b. *Cyclomatic Complexity*

- $V(G) = \text{jumlah region} = 2$
- $V(G) = (\text{jumlah edge} - \text{jumlah node}) + 2 = (5 - 5) + 2 = 2$
- $V(G) = \text{jumlah predicate node} + 1 = 1 + 1 = 2$

c. *Independent Path*

- Jalur 1 = 1-2-3-5
- Jalur 2 = 1-2-4-5

3. *Test Case*

1

Berdasarkan hasil dari *flow graph* yang telah dilakukan sebelumnya, di dapatkan dua jalur independen yang menjadi kasus uji untuk *method update()*. Hasil dari *test case method update()* dapat dilihat pada tabel 6.6.

Tabel 6.6 Hasil Test Case `method update()`

No. Jalur	Prosedur Uji	Expected Result	Result	Status
1	Klas <i>driver</i> memanggil <i>method update()</i> dengan parameter <code>\$data['UPAH_NAMA']=''</code> , <code>'UPAH_HARGA'='123456'</code> , <code>'LOKASI_UPAH_ID'='1'</code>	Nama upah kosong. Sistem menampilkan pesan "gagal"	Nama upah kosong. Sistem menampilkan pesan "gagal"	1 <i>valid</i>
2	Klas <i>driver</i> memanggil <i>method update()</i> dengan parameter <code>\$data['UPAH_NAMA']='nama'</code> , <code>'UPAH_HARGA'='123456'</code> , <code>'LOKASI_UPAH_ID'='1'</code>	Upah berhasil ditambahkan, Sistem menampilkan	Upah berhasil ditambahkan, Sistem menampilkan	<i>valid</i>

		pesan "success"	pesan "success"	
--	--	--------------------	--------------------	--

1

6.2 Pengujian Integrasi

Pengujian integrasi dilakukan untuk menguji hubungan setiap kelas yang ada di dalam sistem. Pengujian ini dilakukan dengan teknik pengujian white-box menggunakan metode *basis path testing*. Method yang digunakan dalam melakukan pengujian integrasi antara method `simpanAnalisa()` kelas `cAnalisa` dan `insert()` kelas `mAnalisa`.

1

6.2.1 Pengujian Integrasi Method `simpanAnalisa()` kelas `cAnalisa` dan method `insert()` kelas `mAnalisa`

Kedua method ini digunakan untuk melakukan penambahan data analisis yang dilakukan oleh PM. Pseudocode pengujian integrasi ini dapat dilihat pada tabel 6.7.

1

1. Pseudocode Method `simpanAnalisa()` kelas `cAnalisa` dan method `insert()` kelas `mAnalisa`

Tabel 6.7 Pseudocode Method `simpanAnalisa()` kelas `cAnalisa` dan method `insert()` kelas `mAnalisa`

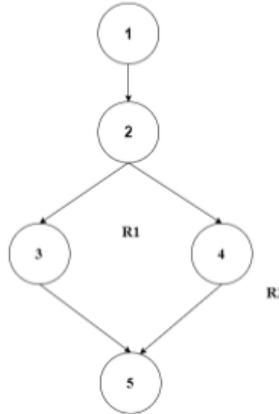
No	Pseudocode	Node
1	Start	1
2	Set namaAnalisa, satuanAnalisa, lokasiAnalisa, barangAnalisa, upahAnalisa, total, set_rules	
3	Set array data	
4	If namaAnalisa = empty then	2
5	Output "gagal"	3
6	Else Set table = master_analisa	4
7	Set ql = get data analisa	
8	If \$ql->num_rows() > 0 then	
9	Output "gagal"	5
10	Else Call method insert dengan array data	6
11	Output "success"	
12	Set query = nilai max column id dari table master_analisa	
13	Set idAnalisa = getRecentID	
14	Set table = detail_analisa	
15	Call method insertdetailanalisa dengan parameter data[barangAnalisa]	

16	Call method insertdetailanalisa dengan parameter data[upahAnalisa]	
17	End if	7
18	End if	8
19	End	
1		

2. Basis Path Testing

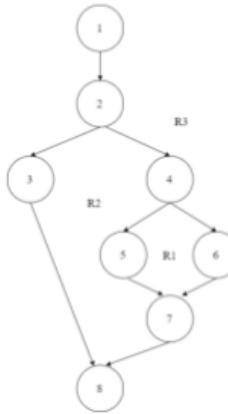
a. Flow Graph

Flow Graph Method simpanAnalisa() klas cAnalisa dan insert() klas mAnalisa dapat dilihat pada gambar 6.4.



Gambar 6.4 Flow Graph Method simpanAnalisa() klas cAnalisa

Dibawah ini adalah **flow graph** dari **method** simpanAnalisa() klas cAnalisa yang digabung dengan **method** insert() klas mAnalisa dapat dilihat pada gambar 6.5



Gambar 6.5 Flow Graph Method simpanAnalisa() klas cAnalisa dan method insert() klas mAnalisa

1

b. *Cyclomatic Complexity*

- $V(G) = \text{jumlah region} = 3$
- $V(G) = (\text{jumlah edge} - \text{jumlah node}) + 2 = (9 - 8) + 2 = 3$
- $V(G) = \text{jumlah predicate node} + 1 = 2 + 1 = 3$

c. *Independent Path*

- Jalur 1 = 1-2-3-8
- Jalur 2 = 1-2-4-5-7-8
- Jalur 3 = 1-2-4-6-7-8

3. *Test Case*

Berdasarkan hasil dari *flow graph* yang telah dilakukan sebelumnya, di dapatkan tiga jalur independen yang menjadi kasus uji. Hasil dari test *case Method simpanAnalisa()* klas *cAnalisa* dan *insert()* klas *mAnalisa* dapat dilihat pada tabel 6.8.

Tabel 6.8 Hasil Test Case Method *simpanAnalisa()* klas *cAnalisa* dan *insert()* klas *mAnalisa*

No. Jalur	Prosedur Uji	Expected Result	Result	Status
1	Klas <i>driver</i> memanggil <i>method simpanAnalisa()</i> dengan data <i>namaAnalisa=''</i> , <i>satuanAnalisa='satuan'</i> , <i>lokasiAnalisa='lokasi'</i> , <i>total='1234'</i> , <i>barangId='1'</i> , <i>upahId='1'</i> , <i>koefisien='1'</i>	Nama analisa kosong. Sistem menampilkan pesan "gagal"	Nama analisa kosong. Sistem menampilkan pesan "gagal"	valid
2	Klas <i>driver</i> memanggil <i>method simpanAnalisa()</i> dengan data <i>namaAnalisa='nama'</i> , <i>satuanAnalisa='satuan'</i> , <i>lokasiAnalisa='lokasi'</i> , <i>total='1234'</i> , <i>barangId='1'</i> , <i>upahId='1'</i> , <i>koefisien='1'</i>	Nama analisa telah ada di dalam sistem, Sistem menampilkan pesan "gagal"	Nama analisa telah ada di dalam sistem, Sistem menampilkan pesan "gagal"	valid
3	Klas <i>driver</i> memanggil <i>method simpanAnalisa()</i> dengan data <i>namaAnalisa='nama baru'</i> , <i>satuanAnalisa='satuan'</i> , <i>lokasiAnalisa='lokasi'</i> , <i>total='1234'</i> , <i>barangId='1'</i> , <i>upahId='1'</i> , <i>koefisien='1'</i>	Analisa berhasil ditambahkan, Sistem menampilkan pesan "success"	Analisa berhasil ditambahkan, Sistem menampilkan pesan "success"	valid

6.3 Pengujian Validasi

Pengujian validasi merupakan tahap dimana sistem diuji apakah semua fungsionalitas yang ada di dalam sistem sudah sesuai dengan *use case scenario* yang sudah dibuat sebelumnya. Pengujian validasi dilakukan dengan menggunakan metode *black-box* dimana pengujian dilakukan hanya dengan melihat hasil keluaran tanpa menelusuri kode program.

1

6.3.1 Pengujian Validasi Masuk

Kasus uji dan hasil pengujian validasi masuk dapat dilihat pada tabel 6.9.

Tabel 6.9 Pengujian Validasi Masuk

No	Test Case	Test Procedure	Expected	Status
1	Berhasil masuk	1. Buka halaman masuk 2. <i>Input email</i> = "faris@yahoo.com", <i>password</i> = "faris123" 3. Menekan tombol <i>login</i>	Berhasil masuk ke dalam sistem.	valid
2	Mengosongkan kolom pada saat <i>login</i>	1. . Buka halaman masuk 2. <i>Input email</i> = "faris@yahoo.com", <i>password</i> = "" 3. Menekan tombol <i>login</i>	Menampilkan pesan "Data tidak boleh kosong"	valid
3	Gagal masuk	1. . Buka halaman masuk 2. <i>Input email</i> = "faris@yahoo.com", <i>password</i> = "lupa123" 3. Menekan tombol <i>login</i>	Menampilkan "Data tidak ditemukan"	valid

6.3.2 Pengujian Validasi Menambah Pengguna

Kasus uji dan hasil pengujian validasi menambah pengguna dapat dilihat pada tabel 6.10.

Tabel 6.10 Pengujian Validasi Menambah Pengguna

No	Test Case	Test Procedure	Expected	Status
----	-----------	----------------	----------	--------

1	Berhasil menambah pengguna	1. Buka halaman daftar pengguna 2. Menekan tombol tambah pengguna 3. Input nama="Faris", jabatan="PM", email="faris@yahoo.com", password="faris123" 3. Menekan tombol daftar	Pengguna baru berhasil ditambahkan ke dalam sistem	<i>valid</i>
2	Mengosongkan kolom pada saat daftar	1. Buka halaman daftar pengguna 2. Menekan tombol tambah pengguna 3. Input nama="Faris", jabatan="PM", email="faris@yahoo.com", password="" 3. Menekan tombol daftar	Menampilkan pesan "Data tidak boleh kosong"	<i>valid</i>
3	Gagal menambah pengguna	1. Buka halaman daftar pengguna 2. Menekan tombol tambah pengguna 3. Input nama="admin", jabatan="admin", email="admin@yahoo.com", password="admin" 3. Menekan tombol daftar	Menampilkan pesan "Username telah terdaftar"	<i>valid</i>

6.3.3 Pengujian Validasi Keluar

Kasus uji dan hasil pengujian validasi keluar dapat dilihat pada tabel 6.11.

Tabel 6.11 Pengujian Validasi Keluar

No	Test Case	Test Procedure	Expected	Status
1	Berhasil keluar	1. Telah masuk ke dalam sistem 2. Menekan tombol "keluar"	Berhasil keluar dari sistem	<i>valid</i>

6.3.4 Pengujian Validasi Melihat Daftar Pengguna

Kasus uji dan hasil pengujian validasi daftar pengguna dapat dilihat pada tabel 6.12.

Tabel 6.12 Pengujian Validasi Melihat Daftar Pengguna

No	Test Case	Test Procedure	Expected	Status
1	Melihat daftar pengguna	1. Membuka halaman daftar pengguna	Menampilkan daftar pengguna	valid

6.3.5 Pengujian Validais Mengubah Pengguna

Kasus uji dan hasil pengujian validasi mengubah pengguna dapat dilihat pada tabel 6.13.

Tabel 6.13 Pengujian Validais Mengubah Pengguna

No	Test Case	Test Procedure	Expected	Status
1	Mengubah data pengguna	1. Membuka halaman daftar pengguna 2. Menekan tombol <i>edit</i> pada pengguna yang dipilih 3. <i>Input</i> nama="admin123", jabatan="admin", email="admin@yahoo.com", password="admin123" 4. menekan tombol <i>update</i>	Berhasil mengubah data pengguna dari dalam sistem	valid

6.3.6 Pengujian Validasi Menghapus Pengguna

Kasus uji dan hasil pengujian validasi menghapus pengguna dapat dilihat pada tabel 6.14.

Tabel 6.14 Pengujian Validasi Menghapus Pengguna

No	Test Case	Test Procedure	Expected	Status
1	Menghapus data pengguna	1. Membuka halaman daftar pengguna 2. Menekan tombol <i>delete</i> pada pengguna yang dipilih	Berhasil menghapus data pengguna dari dalam sistem	valid

6.3.7 Pengujian Validasi Melihat Daftar Barang

8

Kasus uji dan hasil pengujian validasi melihat daftar barang dapat dilihat pada tabel 6.15.

Tabel 6.15 Pengujian Validasi Melihat Daftar Barang

No	Test Case	Test Procedure	Expected	Status
1	Melihat daftar barang	1. Membuka halaman daftar barang	Menampilkan daftar barang	valid

6.3.8 Pengujian Validasi Menambah Barang

1

Kasus uji dan hasil pengujian validasi menambah barang dapat dilihat pada tabel 6.16.

Table 6.16 Pengujian Validasi Menambah Barang

No	Test Case	Test Procedure	Expected	Status
1	Menambah data barang	1. Membuka halaman daftar barang 2. Menekan tombol tambah barang 3. Input nama="kayu", satuan="pcs", harga="100000" 4. Menekan tombol simpan	Berhasil menambah data barang ke dalam sistem	valid

6.3.9 Pengujian Validasi Mengubah Barang

8

Kasus uji dan hasil pengujian validasi mengubah barang dapat dilihat pada tabel 6.17.

Tabel 6.17 Pengujian Validasi Mengubah Barang

No	Test Case	Test Procedure	Expected	Status
1	Mengubah data barang	1. Membuka halaman daftar barang 2. Menekan tombol edit pada data yang dipilih 3. Input nama="kayu", satuan="pcs", harga="100000" 4. Menekan tombol simpan	Berhasil mengubah data barang ke dalam sistem	valid

6.3.10 Pengujian Validasi Menghapus Barang

Kasus uji dan hasil pengujian validasi menghapus barang dapat dilihat pada tabel 6.18.

Tabel 6.18 Pengujian Validasi Menghapus Barang

No	Test Case	Test Procedure	Expected	Status
1	Menghapus data barang	1. Membuka halaman daftar barang 2. Menekan tombol <i>delete</i> pada data yang dipilih	Berhasil menghapus data barang ke dalam sistem	valid

6.3.11 Pengujian Validasi Melihat Daftar Laporan

Kasus uji dan hasil pengujian validasi melihat daftar laporan dapat dilihat pada tabel 6.19.

Tabel 6.19 Pengujian Validasi Melihat Daftar Laporan

No	Test Case	Test Procedure	Expected	Status
1	Melihat daftar laporan	1. Membuka halaman daftar laporan	Menampilkan data daftar laporan	valid

6.3.12 Pengujian Validasi Menambah Laporan

Kasus uji dan hasil pengujian validasi menambah laporan dapat dilihat pada tabel 6.20.

Tabel 6.20 Pengujian Validasi Menambah Laporan

No	Test Case	Test Procedure	Expected	Status
1	Menambah data laporan	1. Membuka halaman daftar laporan 2. Menekan tombol tambah laporan 3. Input nama laporan ="pekerjaan pondasi", nama proyek="DOME", nama pelaksana="Faris", Tanggal="10/12/2019",	Berhasil menambahkan data baru laporan	valid

		uraian pekerjaan="", catatan="", Upload foto="", 4. Menekan tombol simpan		
--	--	--	--	--

6.3.13 Pengujian validasi Mengubah Laporan

Kasus uji dan hasil pengujian validasi mengubah laporan dapat dilihat pada tabel 6.21.

Tabel 6.21 Pengujian validasi Mengubah Laporan

No	Test Case	Test Procedure	Expected	Status
1	Mengubah data laporan	1. Membuka halaman daftar laporan 2. Menekan tombol edit pada data yang dipilih 3. Input nama laporan ="pekerjaan pondasi", nama proyek="DOME", nama pelaksana="Faris", Tanggal="10/12/2019", uraian pekerjaan="", catatan="", Upload foto="", 4. Menekan tombol simpan	Berhasil Mengubah data laporan	valid

6.3.14 Pengujian Validasi Menghapus Laporan

Kasus uji dan hasil pengujian validasi menghapus laporan dapat dilihat pada tabel 6.22.

Tabel 6.22 Pengujian Validasi Menghapus Laporan

No	Test Case	Test Procedure	Expected	Status
1	Menghapus data laporan	1. Membuka halaman daftar laporan 2. Menekan tombol delete pada data yang dipilih	Berhasil menghapus data laporan	valid

6.3.15 Pengujian Validasi Melihat Upah

Kasus uji dan hasil pengujian validasi melihat upah dapat dilihat pada tabel 6.23.

Tabel 6.23 Pengujian Validasi Melihat Upah

No	Test Case	Test Procedure	Expected	Status
1	Melihat data upah	1. Membuka halaman upah	Menampilkan data upah dari dalam sistem	valid

6.3.16 Pengujian Validasi Menambah Upah

Kasus uji dan hasil pengujian validasi menambah upah dapat dilihat pada tabel 6.24.

Tabel 6.24 Pengujian Validasi Menambah Upah

No	Test Case	Test Procedure	Expected	Status
1	Menambah data upah	1. Membuka halaman upah 2. Menekan tombol tambah upah 3. Input nama="listrik", harga="75000", lokasi"JAKARTA" 4. Menekan tombol simpan	Berhasil menambah data upah	valid

6.3.17 Pengujian Validasi Mengubah Upah

Kasus uji dan hasil pengujian validasi mengubah upah dapat dilihat pada tabel 6.25.

Tabel 6.25 Pengujian Validasi Mengubah Upah

No	Test Case	Test Procedure	Expected	Status
1	Mengubah data upah	1. Membuka halaman upah 2. Menekan tombol edit pada data yang dipilih 3. Input nama="listrik", harga="75000", lokasi"JAKARTA" 4. Menekan tombol simpan	Berhasil mengubah data upah	valid

6.3.18 Pengujian Validasi Menghapus Upah

Kasus uji dan hasil pengujian validasi menghapus upah dapat dilihat pada tabel 6.26.

Tabel 6.26 Pengujian Validasi Menghapus Upah

No	Test Case	Test Procedure	Expected	Status
1	Menghapus data upah	1. Membuka halaman upah 2. Menekan tombol <i>delete</i> pada data yang dipilih	Berhasil menghapus data upah	valid

6.3.19 Pengujian Validasi Melihat Analisis

Kasus uji dan hasil pengujian validasi melihat analisis dapat dilihat pada tabel 6.27.

Tabel 6.27 Pengujian Validasi Melihat Analisis

No	Test Case	Test Procedure	Expected	Status
1	Melihat data analisis	1. Membuka halaman analisis	Menampilkan data analisis dari dalam sistem	valid

6.3.20 Pengujian Validasi Menambah Analisis

Kasus uji dan hasil pengujian validasi menambah analisis dapat dilihat pada tabel 6.28.

Tabel 6.28 Pengujian Validasi Menambah Analisis

No	Test Case	Test Procedure	Expected	Status
1	Menambah data analisis	1. Membuka halaman analisis 2. Menekan tombol tambah analisis 3. Input nama="pekerjaan pondasi", satuan="m", lokasi="JAKARTA", uraian bahan="batu bata", koefisien bahan="3", uraian upah="listrik", koefisien upah="10" 4. Menekan tombol simpan	Berhasil menambah data analisis	valid

6.3.21 Pengujian Validasi Mengubah Analisis

3

Kasus uji dan hasil pengujian validasi mengubah analisis dapat dilihat pada tabel 6.29.

Tabel 6.29 Pengujian Validasi Mengubah Analisis

No	Test Case	Test Procedure	Expected	Status
1	Mengubah data analisis	1. Membuka halaman analisis 2. Menekan tombol edit pada data yang dipilih 3. Input nama="pekerjaan pondasi", satuan="m", lokasi="JAKARTA", uraian bahan="batu bata", koefisien bahan="3", uraian upah="listrik", koefisien upah="10" 4. Menekan tombol simpan	Berhasil mengubah data analisis	valid

6.3.22 Pengujian Validasi Menghapus Analisis

3

Kasus uji dan hasil pengujian validasi menghapus analisis dapat dilihat pada tabel 6.30.

Tabel 6.30 Pengujian Validasi Menghapus Analisis

No	Test Case	Test Procedure	Expected	Status
1	Menghapus data analisis	1. Membuka halaman analisis 2. Menekan tombol delete pada data yang dipilih	Berhasil menghapus data analisis	valid

6.3.23 Pengujian Validasi Melihat RAB

1

Kasus uji dan hasil pengujian validasi melihat RAB dapat dilihat pada tabel 6.31.

Tabel 6.31 Pengujian Validasi Melihat RAB

No	Test Case	Test Procedure	Expected	Status
1	Melihat data RAB	1. Membuka halaman RAB	Menampilkan data RAB dari dalam sistem	valid

6.3.24 Pengujian Validasi Menambah RAB

Kasus uji dan hasil pengujian validasi menambah RAB dapat dilihat pada tabel 6.32.

Tabel 6.32 Pengujian Validasi Menambah RAB

No	Test Case	Test Procedure	Expected	Status
1	Menambah data RAB	<ol style="list-style-type: none">1. Membuka halaman RAB2. Menekan tombol tambah RAB3. Input nama="ruko blok A", lokasi="JAKARTA", uraian persiapan ="batu bata", koefisien persiapan ="3", uraian tanah ="bongkat muat", koefisien tanah ="10", uraian arsitek ="bongkat muat", koefisien arsitek ="10", uraian MEP ="bongkat muat", koefisien MEP ="10", uraian lain-lain ="bongkat muat", koefisien lain-lain ="10",4. Menekan tombol simpan	Berhasil menambah data RAB	valid

6.3.25 Pengujian Validasi Mengubah RAB

Kasus uji dan hasil pengujian validasi mengubah RAB dapat dilihat pada tabel 6.33.

Tabel 6.33 Pengujian Validasi Mengubah RAB

No	Test Case	Test Procedure	Expected	Status
1	Mengubah data RAB	<ol style="list-style-type: none">1. Membuka halaman RAB2. Menekan tombol edit pada data yang dipilih3. Input nama="ruko blok A", lokasi="JAKARTA", uraian persiapan ="batu bata", koefisien persiapan ="3", uraian tanah ="bongkat muat", koefisien	Berhasil mengubah data RAB	valid

		<p>tanah = "10", uraian arsitek = "bongkat muat", koefisien arsitek = "10", uraian MEP = "bongkat muat", koefisien MEP = "10", uraian lain-lain = "bongkat muat", koefisien lain-lain = "10",</p> <p>4. Menekan tombol simpan</p>		
--	--	---	--	--

6.3.26 Pengujian Validasi Menghapus RAB

Kasus uji dan hasil pengujian validasi menghapus RAB dapat dilihat pada tabel 6.34.

Tabel 6.34 Pengujian Validasi Menghapus RAB

No	Test Case	Test Procedure	Expected	Status
1	Menghapus data RAB	<p>1. Membuka halaman RAB</p> <p>2. Menekan tombol <i>delete</i> pada data yang dipilih</p>	Berhasil menghapus data RAB	<i>valid</i>

6.3.27 Pengujian Validasi Melihat Proyek

Kasus uji dan hasil pengujian validasi melihat proyek dapat dilihat pada tabel 6.35.

Tabel 6.35 Pengujian Validasi Melihat Proyek

No	Test Case	Test Procedure	Expected	Status
1	Melihat data proyek	1. Membuka halaman melihat proyek	Menampilkan data proyek dari dalam sistem	<i>valid</i>

6.3.28 Pengujian Validasi Menambah Proyek

Kasus uji dan hasil pengujian validasi menambah proyek dapat dilihat pada tabel 6.36.

Tabel 6.36 Pengujian Validasi Menambah Proyek

No	Test Case	Test Procedure	Expected	Status
1	Menambah data proyek	1. Membuka halaman lihat proyek	Berhasil menambah data proyek	<i>valid</i>

		<p>2. Menekan tombol tambah proyek</p> <p>3. <i>Input</i> nama="grisenda", no. spk=" 58/SPK/VII/2019", deskripsi = "", pemilik project ="PT. Gatra", lokasi="JAKARTA"</p> <p>4. Menekan tombol simpan</p>		
--	--	---	--	--

1 6.3.29 Pengujian Validasi Mengubah Proyek

Kasus uji dan hasil pengujian validasi mengubah proyek dapat dilihat pada tabel 6.37.

Tabel 6.37 Pengujian Validasi Mengubah Proyek

No	Test Case	Test Procedure	Expected	Status
1	Mengubah data proyek	<p>1. Membuka halaman lihat proyek</p> <p>2. Menekan tombol edit pada data yang dipilih</p> <p>3. <i>Input</i> nama="grisenda", no. spk=" 58/SPK/VII/2019", deskripsi = "", pemilik project ="PT. Gatra", lokasi="JAKARTA"</p> <p>4. Menekan tombol simpan</p>	Berhasil mengubah data proyek	valid

6.3.30 Pengujian Validasi Menghapus Proyek

Kasus uji dan hasil pengujian validasi menghapus proyek dapat dilihat pada tabel 6.38.

Tabel 6.38 Pengujian Validasi Menghapus Proyek

No	Test Case	Test Procedure	Expected	Status
1	Menghapus data proyek	<p>1. Membuka halaman lihat proyek</p> <p>2. Menekan tombol delete pada data yang dipilih</p>	Berhasil menghapus data proyek	valid

6.3.31 Pengujian Validasi Melihat Time Scheduling

10

Kasus uji dan hasil pengujian validasi melihat time scheduling dapat dilihat pada tabel 6.39.

Tabel 6.39 Pengujian Validasi Melihat Time Scheduling

No	Test Case	Test Procedure	Expected	Status
1	Melihat data time scheduling	1. Membuka halaman melihat time scheduling 2. Menekan tombol tambah schedule 3. Input waktu pengerjaan="3", nama proyek="DOME", bobot rencana ="1", bobot realisasi ="1" 4. Menekan tombol simpan	Menampilkan data melihat time scheduling dari dalam sistem	valid

6.3.32 Pengujian Validasi Menambah Time Scheduling

11

Kasus uji dan hasil pengujian validasi menambah time scheduling dapat dilihat pada tabel 6.40.

Tabel 6.40 Pengujian Validasi Menambah Time Scheduling

No	Test Case	Test Procedure	Expected	Status
1	Menambah data time scheduling	1. Membuka halaman lihat time scheduling 2. Menekan tombol tambah schedule 3. Input waktu pengerjaan="3", nama proyek="DOME", bobot rencana ="1", bobot realisasi ="1" 4. Menekan tombol simpan	Berhasil menambah data time scheduling	valid

6.3.33 Pengujian Validasi Mengubah Time Scheduling

10

Kasus uji dan hasil pengujian validasi mengubah time scheduling dapat dilihat pada tabel 6.41.

Tabel 6.41 Pengujian Validasi Mengubah Time Scheduling

No	Test Case	Test Procedure	Expected	Status
1	Mengubah data time scheduling	1. Membuka halaman lihat time scheduling 2. Menekan tombol edit pada data yang dipilih	Berhasil mengubah data time scheduling	valid

		<p>3. Input waktu pengerjaan="3", proyek="DOME", rencana ="1", realisasi ="1"</p> <p>4. Menekan tombol simpan</p>		
--	--	---	--	--

6.3.34 Pengujian Validasi Menghapus Time Scheduling

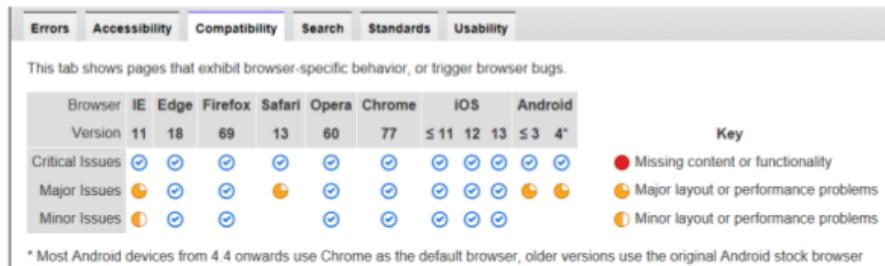
Kasus uji dan hasil pengujian validasi menghapus *time scheduling* dapat dilihat pada tabel 6.42.

Tabel 6.42 Pengujian Validasi Menghapus *Time Scheduling*

No	Test Case	Test Procedure	Expected	Status
1	Menghapus data <i>time scheduling</i>	<p>1. Membuka halaman lihat <i>time scheduling</i></p> <p>2. Menekan tombol <i>delete</i> pada data yang dipilih</p>	Berhasil menghapus data <i>time scheduling</i>	valid

6.4 Pengujian Compability

Untuk mengetahui apakah sistem yang telah kita buat dapat berjalan dengan baik di berbagai *browser* dan *device* maka perlu untuk melakukan pengujian *compatibility*. Pengujian ini dilakukan dengan bantuan aplikasi sortsite, dimana kita dapat menjalankan sistem kita pada aplikasi tersebut. Hasil pengujian *compatibility* melalui aplikasi sortsite didapat hasil seperti pada Gambar 6.5.



Gambar 6.5 Pengujian Compability

1 BAB 7 PENUTUP

7.1 Kesimpulan

1 Berdasarkan dari hasil tahapan-tahapan yang telah dilakukan sebelumnya, mulai dari rekaya kebutuhan hingga ke pengujian sistem. Maka didapatkan beberapa kesimpulan yaitu :

1. Hasil analisis kebutuhan sistem yang didapatkan dari elisitasi kebutuhan sistem dengan menggunakan teknik wawancara menghasilkan 34 kebutuhan sistem dan 6 aktor. Aktor yang terkait adalah pengguna, admni, *purchasing*, PM, pelaksana dan direktur.
2. Pada tahap perancangan menghasilkan pemodelan *sequence diagram*, pemodelan *class diagram*, perancangan komponen dan perancangan antarmuka. Pada pemodelan *sequence diagram*, pemodelan *class diagram* dan perancangan komponen diambil tiga sample yang digunakan yaitu manambah analisa, mengubah analisa dan mengubah upah.
3. Pada tahap implementasi sistem, implementasi kode program dengan menggunakan *framework Codeigniter* dengan bahasa pemrograman PHP . pada imlementasi basis data menggunakan Mysql. Implementasi antarmuka dengan menggunakan HTML, CSS, Javascript, Bootstrap, Datatables, dan Handsontable
4. Pada tahap pengujian sistem, sistem diuji dengan menggunakan 3 jenis pengujian yaitu pengujian unit, pengujian validasi, dan pengujian compatibility. Pada pengujian unit dilakukan dengan menggunakan metode *white-box* serta menggunakan tiga sample uji yang menunjukkan hasil *valid*. Pada pengujian validasi dilakukan dengan menggunakan metode *black-box* serta menggunakan tiga sample uji dan menunjukkan hasil *valid*. Pada pengujian compatibility dilakukan untuk menguji kebutuhan non fungsional dibantu dengan menggunakan aplikasi sortsite, dimana pada aplikasi ini sistem dijalankan di berbagai *browser* dan menghasilkan.

7.2 Saran

Berdasarkan kesimpulan tersebut, terdapat saran untuk pengembangan selanjutnya yaitu:

1. Pada fitur *time schedule*, tampilan dapat dikembangkan menjadi lebih menarik dan dapat ditambahkan fitur pembagian *timeline* kerja.
2. Sistem dapat ditambahkan fitur manajemen keluar dan masuk barang pada suatu kegiatan proyek.

PEMBANGUNAN SISTEM MANAJEMEN PROYEK BERBASIS WEB (STUDI KASUS :PT. MITRA INTI CEMERLANG)

ORIGINALITY REPORT

37 %
SIMILARITY INDEX

19%
INTERNET SOURCES

0%
PUBLICATIONS

32%
STUDENT PAPERS

PRIMARY SOURCES

- | | | | |
|--|----------|---|----------------|
| | 1 | Submitted to Universitas Brawijaya | 27 % |
| | | Student Paper | |
| | 2 | www.scribd.com | 1 % |
| | | Internet Source | |
| | 3 | docobook.com | 1 % |
| | | Internet Source | |
| | 4 | adamfr97.blogspot.com | 1 % |
| | | Internet Source | |
| | 5 | media.neliti.com | 1 % |
| | | Internet Source | |
| | 6 | repository.usu.ac.id | 1 % |
| | | Internet Source | |
| | 7 | es.scribd.com | <1 % |
| | | Internet Source | |
| | 8 | id.123dok.com | <1 % |
| | | Internet Source | |
| | 9 | www.collis.id | |

Internet Source

<1 %

10

[id.scribd.com](#)

Internet Source

<1 %

11

[pt.scribd.com](#)

Internet Source

<1 %

12

[www.slideshare.net](#)

Internet Source

<1 %

13

[core.ac.uk](#)

Internet Source

<1 %

14

[library.binus.ac.id](#)

Internet Source

<1 %

15

[docplayer.info](#)

Internet Source

<1 %

16

[eprints.umm.ac.id](#)

Internet Source

<1 %

17

[www.tutorialkampus.com](#)

Internet Source

<1 %

18

[digilib.unila.ac.id](#)

Internet Source

<1 %

19

[sir.stikom.edu](#)

Internet Source

<1 %

20

[febiiainpky.blogspot.com](#)

Internet Source

<1 %

21	vdocuments.site Internet Source	<1 %
22	kitabisa72.blogspot.com Internet Source	<1 %
23	widuri.raharja.info Internet Source	<1 %
24	aimos.ugm.ac.id Internet Source	<1 %
25	etheses.uin-malang.ac.id Internet Source	<1 %
26	www.informasiproyek.com Internet Source	<1 %
27	www.arsitag.com Internet Source	<1 %
28	fadilajunitan.blogspot.com Internet Source	<1 %
29	system.hideandseek.co.jp Internet Source	<1 %
30	thesis.binus.ac.id Internet Source	<1 %
31	eprints.ums.ac.id Internet Source	<1 %
32	Submitted to University of Westminster	

-
- 33 Rangga Ramadhani, Damris Muhammad, Dyah Kumala Sari,. "Analisis Keuntungan Sistem Pembayaran Monthly Payment Dan Progress Payment Pada Proyek Pembangunan (Studi Kasus : Pembangunan Saluran Drainase Sungai Dingin dan Pembangunan Gedung Unit Pelayanan Teknis (UPT), Muara Bungo)", Jurnal Engineering, 2019
Publication <1 %
- 34 j-ptiik.ub.ac.id Internet Source <1 %
- 35 jakartarentofficespace.blogspot.com Internet Source <1 %
- 36 caheastborn92.blogspot.com Internet Source <1 %
- 37 eprints.akakom.ac.id Internet Source <1 %
- 38 masykurali19.blogspot.com Internet Source <1 %
- 39 repository.uinjkt.ac.id Internet Source <1 %
- 40 lppm.atmaluhur.ac.id Internet Source <1 %

41	www.irwanhendrasaputra.web.id	<1 %
Internet Source		
42	jurnal.uns.ac.id	<1 %
Internet Source		
43	text-id.123dok.com	<1 %
Internet Source		
44	elib.unikom.ac.id	<1 %
Internet Source		
45	docslide.us	<1 %
Internet Source		
46	kinetik.umm.ac.id	<1 %
Internet Source		
47	yolandhaartaulisitompul97.blogspot.com	<1 %
Internet Source		
48	sttgarut.ac.id	<1 %
Internet Source		
49	edoc.site	<1 %
Internet Source		
50	ikanuraei.blogspot.com	<1 %
Internet Source		

Exclude quotes

Off

Exclude matches

Off

Exclude bibliography Off