

KATA PENGANTAR

Segala puji bagi Allah SWT yang telah memberikan kemudahan sehingga dapat menyelesaikan laporan Internship I ini, tanpa pertolongan-Nya mungkin penulis tidak akan sanggup menyelesaikannya dengan baik. Shalawat dan salam semoga terlimpah curahkan kepada Nabi Muhammad SAW beserta sahabat dan keluarga Beliau.

Laporan ini disusun untuk memenuhi kelulusan matakuliah Internship I pada Program Studi DIV Teknik Informatika. Proses Internship I ini juga tidak terlepas dari bantuan berbagai pihak. Oleh karena itu, pada kata pengantar ini penulis menyampaikan terimakasih kepada :

1. M. Yusril Helmi Setyawan, S.Kom., M.Kom. selaku Ketua Program Studi DIV Teknik Informatika Tahun Akademik 2019/2020.
2. Nisa Hanum Harani, S.Kom, M.T. selaku Koordinator Internship I Tahun Akademik 2019/2020.
3. Cahyo Prianto, S.Pd., M.T. selaku Dosen Wali dan Pembimbing Internal yang telah memberikan pengarahan dan masukan dalam pembuatan Internship I ini.
4. Orang tua dan keluarga yang telah memberikan bantuan dukungan material dan moral.

Penulis telah membuat laporan ini dengan sebaik-baiknya, diharapkan memberikan kritik dan saran dari semua pihak yang bersifat membangun, terimakasih.

Bandung, 17 Januari 2020

Penulis

1. Pendahuluan

Perkembangan teknologi informasi telah membawa dampak dalam kehidupan salah satunya dapat mempengaruhi efektivitas operasional pada perusahaan/ organisasi. Sebuah teknologi informasi yang diterapkan dimanfaatkan sebagai salah satu solusi untuk meminimalisir tingkat kesalahan pada pengguna, baik pelayanan kepada konsumen maupun dalam mengelola distribusi informasi dan data kepada sesama karyawan perusahaan. Pengembangan perangkat lunak yang tepat sasaran hendaknya disesuaikan dengan segala kebutuhan administrasi yang dibutuhkan sehari-hari, sehingga pada proses input akhirnya segala keputusan dan kebijakan yang akan ditentukan dapat menunjang perkembangan perusahaan/ organisasi.

Berbagai proyek seperti produksi, konstruksi bangunan, pengembangan sistem, dan proyek penelitian dapat dimodelkan sebagai jaringan, yang dapat ditangani dengan teknik manajemen yang canggih. Dalam banyak kasus, menyelesaikan proyek dengan batas waktu adalah persyaratan yang paling penting. Sebagai contoh, memenuhi tanggal pengiriman produk, yang ditetapkan sebagai batas waktu, adalah kendala penting bagi banyak produsen.

PT. Javan Cipta Solusi merupakan perusahaan yang bergerak di bidang Dukungan dan Pengembangan Perangkat Lunak Sumber Terbuka sejak tahun 2008. Berdasarkan hasil pengamatan dokumen laporan proyek PT. Javan Cipta Solusi, dalam mengerjakan setiap proyeknya PT. Javan Cipta Solusi membutuhkan manajer proyek yang bertugas untuk mengatur jalannya proyek. Proses manajemen proyek dibagi dalam 4 tahap yaitu perencanaan (planning), organisasi (organizing), pelaksanaan (actuating) dan pengendalian (controlling), dimana masing-masing tahap memiliki keterkaitan satu dengan yang lain. Pada tahap perencanaan dan pengaturan, seorang manajer proyek

di PT. Javan Cipta Solusi bertanggung jawab dalam melakukan beberapa kegiatan seperti penyusunan kegiatan, memperkirakan durasi proyek, dan menentukan sumber daya apa saja yang dibutuhkan. Masalah yang ada pada tahap ini berkaitan dengan waktu dan mutu, manajer proyek tidak memiliki alat bantu khusus dalam menyusun kegiatan, sehingga umumnya dibuat berdasarkan perkiraan, proses ini membutuhkan waktu yang relatif lama dan kualitas perencanaannya tergantung dari pengalaman manajer proyek yang menanganinya. Pada tahap eksekusi, seorang manajer proyek menjalankan setiap aktivitas yang sudah didefinisikan pada tahap perencanaan dan pengaturan, proses pelaksanaan proyek disesuaikan dengan rencana yang dibuat mulai dari susunan kegiatan, dan sumber daya lain yang dibutuhkan. Permasalahan yang terjadi pada tahap ini berkaitan dengan mutu, aktivitas dan penggunaan sumber daya pada proyek dituntut untuk sesuai dengan perencanaan sebelumnya demi menghindari terjadinya ketidaksesuaian. Semakin besar tingkat ketidaksesuaian antara rencana dan realisasi proyek dapat beresiko menimbulkan kerugian bagi perusahaan dan pemangku kepentingan yang terlibat dalam proyek tersebut. Tahap pengendalian berfungsi untuk meninjau kembali proses yang telah berjalan dan mengevaluasi rencana yang ada sebagai bentuk pengendalian, sistem pelaporannya menggunakan berbagai macam template dokumen elektronik yang telah dibuat sebelumnya oleh perusahaan, selanjutnya template tersebut diisi oleh pemangku kepentingan terkait dan menyerahkan laporan ke perusahaan secara berkala. Permasalahan yang terjadi pada tahap ini berkaitan dengan waktu dan mutu. Data laporan saat ini relatif tidak akurat karena bukan data yang bersifat real-time.

Dari permasalahan tersebut, maka dibuatlah “Sistem Manajemen Proyek Menggunakan Critical Path Method Untuk Memprediksi Durasi Total Pengerjaan Proyek di PT. Javan Cipta Solusi” dengan tujuan untuk membantu

proses perencanaan proyek agar sesuai dengan realisasi sehingga tidak menimbulkan kerugian pada perusahaan dan para pemangku kepentingan.

Aplikasi ini dibangun dengan menggunakan bahasa pemrograman PHP (Hypertext Pre Processor), PHP dengan menggunakan framework CodeIgniter dan algoritma CPM (Critical Path Method).

2. Dasar Pemrograman Web

2.1 Pengertian Pemrograman Web



Pemrograman web diambil dari 2 suku kata yaitu pemrograman dan web. Pemrograman yang dalam bahasa English adalah programming dan diartikan proses, cara, perbuatan program. Definisi Web adalah jaringan komputer yang terdiri dari kumpulan situs internet yang menawarkan teks dan grafik dan suara dan sumber daya animasi melalui protokol transfer hypertext.

Orang banyak mengenal web dengan istilah WWW (world wide web), World Wide Web adalah layanan internet yang paling populer saat ini internet mulai dikenal dan digunakan secara luas setelah adanya layanan WWW .

WWW adalah halaman-halaman website yang dapat saling terkoneksi satu dengan lainnya (hyperlink) yang membentuk samudra belantara informasi . WWW berjalan dengan protokol HyperText Transfer Protokol (HTTP). Halaman Web merupakan file teks murni (plain text) yang berisi sintaks-sintaks HTML yang dapat dibuka/ dilihat/ diterjemahkan dengan Internet Browser . Sintaks HTML mampu memuat konten text, gambar, audio, video dan animasi. Kini internet identik dengan web, karena kepopuleran web sebagai standar interface pada layanan-layanan yang ada di internet. Banyak keuntungan yang diberikan oleh Aplikasi berbasis Web dari pada aplikasi berbasis desktop, sehingga aplikasi berbasis web telah diadopsi oleh perusahaan sebagai bagian dari strategi teknologi informasinya, karena beberapa alasan :

- Akses informasi mudah,
- Setup server lebih mudah
- Informasi mudah didistribusikan
- Bebas platform, informasi dapat di sajikan oleh browser web pada sistem operasi mana saja karena adanya standar dokumen berbagai tipe data dapat disajikan

HTTP (HyperTextTransfer Protocol) adalah protokol yang dipergunakan untuk mentransfer dokumen dalam World Wide Web(WWW). Protokol ini adalah protokol ringan, tidak berstatus dan generik yang dapat dipergunakan berbagai macam tipe dokumen. Pengembangan HTTP dikoordinasi oleh Konsorsium World Wide Web(W3C) dan grup bekerja Internet Engineering Task Force(IETF), bekerja dalam publikasi satu seri RFC, yang paling terkenal RFC 2616, yang menjelaskan HTTP/1,1, versi HTTP yang digunakan umum sekarang ini. HTTP adalah sebuah protokol

meminta/menjawab antara client dan server. Sebuah client HTTP seperti web browser, biasanya memulai permintaan dengan membuat hubungan TCP/IP ke port tertentu di tuan rumah yang jauh (biasanya port 80). Sebuah server HTTP yang mendengarkan di port tersebut menunggu client mengirim kode permintaan (request), seperti "GET / HTTP/1.1" (yang akan meminta halaman yang sudah ditentukan), diikuti dengan pesan MIME yang memiliki beberapa informasi kode kepala yang menjelaskan aspek dari permintaan tersebut, diikuti dengan badan dari data tertentu. Beberapa kepala (header) juga bebas ditulis atau tidak, sementara lainnya (seperti tuan rumah) diperlukan oleh protokol HTTP/1.1. Begitu menerima kode permintaan (dan pesan, bila ada), server mengirim kembali kode jawaban, seperti "200 OK", dan sebuah pesan yang diminta, atau sebuah pesan error atau pesan lainnya.

Sejarah Protokol HTTP pertama kali dipergunakan dalam WWW pada tahun 1990. Pada saat tersebut yang dipakai adalah protokol HTTP versi 0.9. Versi 0.9 ini adalah protokol transfer dokumen secara mentah, maksudnya adalah data dokumen dikirim sesuai dengan isi dari dokumen tersebut tanpa memandang tipe dari dokumen. Kemudian pada tahun 1996 protokol HTTP diperbaiki menjadi HTTP versi 1.0. Perubahan ini untuk mengakomodasi tipe-tipe dokumen yang hendak dikirim beserta encoding yang dipergunakan dalam pengiriman data dokumen. Sesuai dengan perkembangan infrastruktur internet maka pada tahun 1999 dikeluarkan HTTP versi 1.1 untuk mengakomodasi proxy, cache dan koneksi yang persisten.

2.2 Pengertian HTML



HTML merupakan turunan atau pengembangan dari SGML (Standar Generalized Markup Language). HTML sendiri dikembangkan oleh Tim Berners-Lee sewaktu masih bekerja di CERN yang pertama kali dipopulerkan oleh browser Mosaic yang dikembangkan NCSA. Selama awal tahun 1990an, HTML semakin memiliki perkembangan yang sangat cepat. Akan tetapi pengembangan resmi HTML baru dikeluarkan oleh Internet Engineering Task Force (IETF) yang dikeluarkan pada tahun 1995. HTML2 yang dikembangkan ini merupakan turunan dari HTML+ pada tahun 1993. HTML3 yang juga dirilis pada tahun 1995 mempunyai kemampuan yang jauh lebih bagus dari versi sebelumnya. Merupakan hasil dari usaha yang dikembangkan oleh World Wide Web Consortium's (W3C) yang kemudian menghasilkan HTML3 di tahun 1996 dan dirilis HTML4 pada akhir tahun tersebut yaitu 1997 dan 1998.

Secara Umum Pemrograman Web dibagi menjadi 2 bagian yaitu :

1. Client Side Scripting (CSS)



Client Side Scripting digunakan ketika browser (pengguna) klien memiliki semua kode dan halaman tersebut diubah berdasarkan informasi klien (pengguna). Browser Web mengeksekusi skrip sisi klien yang terletak di dalam komputer pengguna. Skrip sisi klien juga dikenal sebagai skrip tertanam (karena mereka sering disematkan dalam dokumen HTML atau XHTML).

Browser mendapatkan halaman yang dikirim oleh server & mengeksekusi skrip sisi klien. Skrip sisi klien tidak dapat digunakan untuk bergabung dengan database di server web. Skrip sisi klien tidak bisa mendapatkan sistem file yang terletak di server web. Catatan dan pengaturan yang bersifat lokal di komputer klien (pengguna) dapat didekati menggunakan bahasa skrip sisi Klien. Secara umum diamati bahwa respons dari skrip sisi klien lebih cepat bila dibandingkan dengan bahasa skrip sisi server saat skrip disiapkan di komputer lokal.

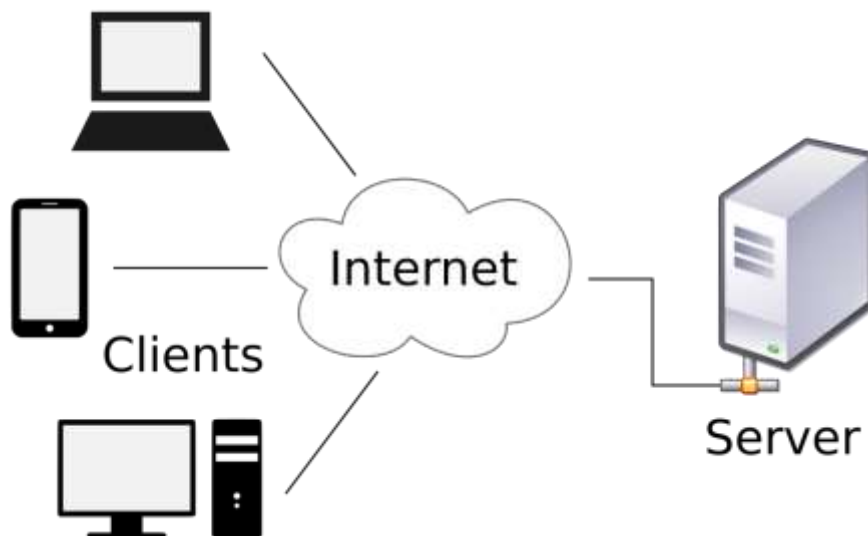
Contoh Script Paling Populer:

1.Java Script

2.XML

3.CSS

2. Server Side Scripting (SSS)



Server Side Scripting adalah teknik yang digunakan dalam pengembangan web yang melibatkan penggunaan skrip pada server web yang menghasilkan respons yang disesuaikan untuk permintaan setiap pengguna (klien) ke situs web. Alternatifnya adalah untuk server web itu sendiri untuk memberikan halaman web statis. Skrip dapat ditulis dalam salah satu dari sejumlah bahasa skrip sisi server yang tersedia (lihat di bawah). Skrip sisi server dibedakan dari skrip sisi klien di mana skrip tertanam, seperti JavaScript, dijalankan sisi klien dalam browser web, tetapi kedua teknik ini sering digunakan bersama.

Server Side Scripting sering digunakan untuk menyediakan antarmuka yang disesuaikan untuk pengguna. Skrip ini dapat mengumpulkan

karakteristik klien untuk digunakan dalam menyesuaikan respons berdasarkan karakteristik tersebut, persyaratan pengguna, hak akses, dll. Skrip sisi server juga memungkinkan pemilik situs web untuk menyembunyikan kode sumber yang menghasilkan antarmuka, sedangkan dengan sisi klien scripting, pengguna memiliki akses ke semua kode yang diterima oleh klien. Sisi buruk dari penggunaan skrip sisi server adalah bahwa klien harus membuat permintaan lebih lanjut melalui jaringan ke server untuk menunjukkan informasi baru kepada pengguna melalui browser web. Permintaan ini dapat memperlambat pengalaman bagi pengguna, menempatkan lebih banyak beban di server, dan mencegah penggunaan aplikasi saat pengguna terputus dari server.

Ketika server menyajikan data dengan cara yang umum digunakan, misalnya sesuai dengan protokol HTTP atau FTP, pengguna dapat memilih sejumlah program klien (sebagian besar browser web modern dapat meminta dan menerima data menggunakan kedua protokol tersebut). Dalam hal aplikasi yang lebih khusus, pemrogram dapat menulis sendiri server, klien, dan protokol komunikasi mereka, yang hanya dapat digunakan satu sama lain.

2.3 Pengertian PHP



PHP adalah singkatan dari “PHP: Hypertext Preprocessor”, yaitu bahasa pemrograman disisi server yang digunakan secara luas untuk penanganan pembuatan dan pengembangan sebuah situs web dan bisa digunakan bersamaan dengan HTML. Ketika Anda mengakses sebuah URL, maka web browser akan melakukan request ke sebuah web server.

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

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

Pada November 1997, dirilis PHP/FI 2.0. Pada rilis ini, interpreter

PHP sudah diimplementasikan dalam program C. Dalam rilis ini disertakan juga modul-modul ekstensi yang meningkatkan kemampuan PHP/FI secara signifikan.

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

Pada pertengahan tahun 1999, Zend merilis interpreter PHP baru dan rilis tersebut dikenal dengan PHP 4.0. PHP 4.0 adalah versi PHP yang paling banyak dipakai pada awal abad ke-21. Versi ini banyak dipakai disebabkan kemampuannya untuk membangun aplikasi web kompleks tetapi tetap memiliki kecepatan dan stabilitas yang tinggi.

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

Beberapa kelebihan PHP dari bahasa pemrograman web, antara lain:

- Bahasa pemrograman PHP adalah sebuah bahasa script yang tidak melakukan sebuah kompilasi dalam penggunaannya.
- Web Server yang mendukung PHP dapat ditemukan di mana - mana dari mulai apache, IIS, Lighttpd, hingga Xitami dengan konfigurasi yang relatif mudah.
- Dalam sisi pengembangan lebih mudah, karena banyaknya milis - milis dan

developer yang siap membantu dalam pengembangan.

- Dalam sisi pemahaman, PHP adalah bahasa scripting yang paling mudah karena memiliki referensi yang banyak.
- PHP adalah bahasa open source yang dapat digunakan di berbagai mesin (Linux, Unix, Macintosh, Windows) dan dapat dijalankan secara runtime melalui console serta juga dapat menjalankan perintah-perintah system.

PHP memiliki 8 tipe data, yaitu:

1. Boolean
2. Integer
3. Float/ Double
4. String
5. Array
6. Object
7. Resource
8. NULL

2.4 Pengertian MySql



MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (bahasa Inggris: database management system) atau DBMS yang multialur, multipengguna, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis di bawah lisensi GNU General Public License (GPL), tetapi mereka juga menjual di bawah lisensi komersial untuk kasus-kasus di mana penggunaannya tidak cocok dengan penggunaan GPL.

Tidak sama dengan proyek-proyek seperti Apache, di mana perangkat lunak dikembangkan oleh komunitas umum, dan hak cipta untuk kode sumber dimiliki oleh penulisnya masing-masing, MySQL dimiliki dan disponsori oleh sebuah perusahaan komersial Swedia MySQL AB, di mana memegang hak cipta hampir atas semua kode sumbernya. Kedua orang Swedia dan satu orang Finlandia yang mendirikan MySQL AB adalah: David Axmark, Allan Larsson, dan Michael "Monty" Widenius.

MySQL adalah sebuah implementasi dari sistem manajemen basis

data relasional (RDBMS) yang didistribusikan secara gratis di bawah lisensi GPL (General Public License). Setiap pengguna dapat secara bebas menggunakan MySQL, tetapi dengan batasan perangkat lunak tersebut tidak boleh dijadikan produk turunan yang bersifat komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam basisdata yang telah ada sebelumnya; SQL (Structured Query Language). SQL adalah sebuah konsep pengoperasian basisdata, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis.

Kehandalan suatu sistem basisdata (DBMS) dapat diketahui dari cara kerja pengoptimasi-nya dalam melakukan proses perintah-perintah SQL yang dibuat oleh pengguna maupun program-program aplikasi yang memanfaatkannya. Sebagai peladen basis data, MySQL mendukung operasi basisdata transaksional maupun operasi basisdata non-transaksional. Pada modus operasi non-transaksional, MySQL dapat dikatakan unggul dalam hal unjuk kerja dibandingkan perangkat lunak peladen basisdata kompetitor lainnya. Namun pada modus non-transaksional tidak ada jaminan atas reliabilitas terhadap data yang tersimpan, karenanya modus non-transaksional hanya cocok untuk jenis aplikasi yang tidak membutuhkan reliabilitas data seperti aplikasi blogging berbasis web (wordpress), CMS, dan sejenisnya. Untuk kebutuhan sistem yang ditujukan untuk bisnis sangat disarankan untuk menggunakan modus basisdata transaksional, hanya saja sebagai konsekuensinya unjuk kerja MySQL pada modus transaksional tidak secepat unjuk kerja pada modus non-transaksional.

MySQL pada awalnya diciptakan pada tahun 1979, oleh Michael "Monty" Widenius, seorang programmer komputer asal Swedia. Monty mengembangkan sebuah sistem database sederhana yang dinamakan

UNIREG yang menggunakan koneksi low-level ISAM database engine dengan indexing. Pada saat itu Monty bekerja pada perusahaan bernama TcX di Swedia.

TcX pada tahun 1994 mulai mengembangkan aplikasi berbasis web, dan berencana menggunakan UNIREG sebagai sistem database. Namun sayangnya, UNIREG dianggap tidak cocok untuk database yang dinamis seperti web.

TcX kemudian mencoba mencari alternatif sistem database lainnya, salah satunya adalah mSQL (miniSQL). Namun mSQL versi 1 ini juga memiliki kekurangan, yaitu tidak mendukung indexing, sehingga performanya tidak terlalu bagus.

Dengan tujuan memperbaiki performa mSQL, Monty mencoba menghubungi David Hughes (programmer yang mengembangkan mSQL) untuk menanyakan apakah ia tertarik mengembangkan sebuah konektor di mSQL yang dapat dihubungkan dengan UNIREG ISAM sehingga mendukung indexing. Namun saat itu Hughes menolak, dengan alasan sedang mengembangkan teknologi indexing yang independen untuk mSQL versi 2.

Dikarenakan penolakan tersebut, David Hughes, TcX (dan juga Monty) akhirnya memutuskan untuk merancang dan mengembangkan sendiri konsep sistem database baru. Sistem ini merupakan gabungan dari UNIREG dan mSQL (yang source codenya dapat bebas digunakan). Sehingga pada May 1995, sebuah RDBMS baru, yang dinamakan MySQL dirilis.

David Axmark dari Detron HB, rekanan TcX mengusulkan agar MySQL di 'jual' dengan model bisnis baru. Ia mengusulkan agar MySQL dikembangkan dan dirilis dengan gratis. Pendapatan perusahaan selanjutnya

di dapat dari menjual jasa “support” untuk perusahaan yang ingin mengimplementasikan MySQL. Konsep bisnis ini sekarang dikenal dengan istilah Open Source.

Pada tahun 1995 itu juga, TcX berubah nama menjadi MySQL AB, dengan Michael Widenius, David Axmark dan Allan Larsson sebagai pendirinya. Titel “AB” di belakang MySQL, adalah singkatan dari “Aktiebolag”, istilah PT (Perseroan Terbatas) bagi perusahaan Swedia.

MySQL memiliki beberapa keistimewaan, antara lain:

1. Portabilitas. MySQL dapat berjalan stabil pada berbagai sistem operasi seperti Windows, Linux, FreeBSD, Mac Os X Server, Solaris, Amiga, dan masih banyak lagi.
2. Perangkat lunak sumber terbuka. MySQL didistribusikan sebagai perangkat lunak sumber terbuka, di bawah lisensi GPL sehingga dapat digunakan secara gratis.
3. Multi-user. MySQL dapat digunakan oleh beberapa pengguna dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.
4. 'Performance tuning', MySQL memiliki kecepatan yang menakjubkan dalam menangani query sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.
5. Ragam tipe data. MySQL memiliki ragam tipe data yang sangat kaya, seperti signed / unsigned integer, float, double, char, text, date, timestamp, dan lain-lain.
6. Perintah dan Fungsi. MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah Select dan Where dalam perintah (query).
7. Keamanan. MySQL memiliki beberapa lapisan keamanan seperti level

subnetmask, nama host, dan izin akses user dengan sistem perizinan yang mendetail serta sandi terenkripsi.

8. Skalabilitas dan Pembatasan. MySQL mampu menangani basis data dalam skala besar, dengan jumlah rekaman (records) lebih dari 50 juta dan 60 ribu tabel serta 5 miliar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.
9. Konektivitas. MySQL dapat melakukan koneksi dengan klien menggunakan protokol TCP/IP, Unix soket (UNIX), atau Named Pipes (NT).
10. Lokalisasi. MySQL dapat mendeteksi pesan kesalahan pada klien dengan menggunakan lebih dari dua puluh bahasa. Meski pun demikian, bahasa Indonesia belum termasuk di dalamnya.
11. Antar Muka. MySQL memiliki antar muka (interface) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (Application Programming Interface).
12. Klien dan Peralatan. MySQL dilengkapi dengan berbagai peralatan (tool) yang dapat digunakan untuk administrasi basis data, dan pada setiap peralatan yang ada disertakan petunjuk online.
13. Struktur tabel. MySQL memiliki struktur tabel yang lebih fleksibel dalam menangani ALTER TABLE, dibandingkan basis data lainnya semacam PostgreSQL ataupun Oracle.

Berikut Fitur serta kapabilitas yang dimiliki oleh MySQL:

1. Unjuk kerja yang tinggi dalam memproses query sederhana, dalam arti dapat memproses lebih banyak SQL per satuan waktu.

2. Memiliki lebih banyak tipe data seperti: signed/unsigned integer yang memiliki panjang data sebesar 1,2,3,4 dan 8 byte, FLOAT, DOUBLE, CHAR, VARCHAR, TEXT, BLOB, DATE, TIME, DATETIME, TIMESTAMP, YEAR, SET dan tipe ENUM.
3. Mendukung field yang dijadikan Index, dengan maksimal 32 index dalam satu tabel. *
4. MYSQL memiliki beberapa lapisan keamanan, seperti subnetmask, nama host, dan izin akses user dengan sistem perijinan yang mendetail serta sandi/password terenkripsi.
5. Konektivitas , MySQL dapat melakukan koneksi dengan klien menggunakan protokol TCP/IP, Unix socket (UNIX), atau Named Pipes(NT).
6. Multi-user. MySQL dapat digunakan oleh beberapa pengguna dalam waktu yang bersamaan tanpa mengalami masalah atau konflik
7. Command and function, MySQL memiliki fungsi dan operator secara penuh yang mendukung perintah select dan where dalam query.
8. Structure Table, MySQL memiliki struktur tabel yang lebih fleksibel dalam menangani ALTER TABLE dibandingkan DBMS lainnya.
9. Mendukung penuh terhadap kalimat SQL GROUP BY dan ORDER BY. Mendukung terhadap fungsi penuh (COUNT(), COUNT(), DISTINCT() AVG(), STD(), SUM(), MAX() dan MIN())

2.5 Pengertian CodeIgniter



CodeIgniter merupakan aplikasi sumber terbuka yang berupa kerangka kerja PHP dengan model MVC (Model, View, Controller) untuk membangun website dinamis dengan menggunakan PHP. CodeIgniter memudahkan pengembang web untuk membuat aplikasi web dengan cepat mudah dibandingkan dengan membuatnya dari awal. CodeIgniter dirilis pertama kali pada 28 Februari 2006.

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

Ada beberapa alasan mengapa menggunakan Kerangka kerja:

- Mempercepat dan mempermudah pembangunan sebuah aplikasi web.
- Relatif memudahkan dalam proses maintenance karena sudah ada pola tertentu dalam sebuah kerangka kerja (dengan syarat programmer

mengikuti pola standar yang ada)

- Umumnya kerangka kerja menyediakan fasilitas-fasilitas yang umum dipakai sehingga kita tidak perlu membangun dari awal (misalnya validasi, ORM, pagination, multiple database, scaffolding, pengaturan session, error handling, dll)
- Lebih bebas dalam pengembangan jika dibandingkan CMS

Model View Controller merupakan suatu konsep yang cukup populer dalam pembangunan aplikasi web, berawal pada bahasa pemrograman Small Talk, MVC memisahkan pengembangan aplikasi berdasarkan komponen utama yang membangun sebuah aplikasi seperti manipulasi data, user interface, dan bagian yang menjadi kontrol aplikasi. Terdapat 3 jenis komponen yang membangun suatu MVC pattern dalam suatu aplikasi yaitu:

1. View, merupakan bagian yang menangani presentation logic. Pada suatu aplikasi web bagian ini biasanya berupa file template HTML, yang diatur oleh controller. View berfungsi untuk menerima dan merepresentasikan data kepada user. Bagian ini tidak memiliki akses langsung terhadap bagian model.
2. Model, biasanya berhubungan langsung dengan database untuk memanipulasi data (insert, update, delete, search), menangani validasi dari bagian controller, tetapi tidak dapat berhubungan langsung dengan bagian view.
3. Controller, merupakan bagian yang mengatur hubungan antara bagian model dan bagian view, controller berfungsi untuk menerima request dan data dari user kemudian menentukan apa yang akan diproses oleh aplikasi.

Dengan menggunakan prinsip MVC suatu aplikasi dapat dikembangkan sesuai dengan kemampuan developernya, yaitu programmer

yang menangani bagian model dan controller, sedangkan designer yang menangani bagian view, sehingga penggunaan arsitektur MVC dapat meningkatkan maintainability dan organisasi kode. Walaupun demikian dibutuhkan komunikasi yang baik antara programmer dan designer dalam menangani variabel-variabel yang akan ditampilkan.

Ada beberapa kelebihan CodeIgniter (CI) dibandingkan dengan Kerangka kerja PHP lain,

- Performa sangat cepat: salah satu alasan tidak menggunakan kerangka kerja adalah karena eksekusinya yang lebih lambat daripada PHP from the scratch, tapi Codeigniter sangat cepat bahkan mungkin bisa dibilang codeigniter merupakan kerangka kerja yang paling cepat dibanding kerangka kerja yang lain.
- Konfigurasi yang sangat minim (nearly zero configuration): tentu saja untuk menyesuaikan dengan database dan keleluasaan routing tetap diizinkan melakukan konfigurasi dengan mengubah beberapa file konfigurasi seperti database.php atau autoload.php, tetapi untuk menggunakan codeigniter dengan setting standard, anda hanya perlu mengubah sedikit saja file pada folder config.
- Banyak komunitas: dengan banyaknya komunitas CI ini, memudahkan kita untuk berinteraksi dengan yang lain, baik itu bertanya atau teknologi terbaru.
- Dokumentasi yang sangat lengkap: Setiap paket instalasi codeigniter sudah disertai user guide yang sangat bagus dan lengkap untuk dijadikan permulaan, bahasanya pun mudah dipahami.

2.6 Pengertian CPM



Critical Path Method (CPM) adalah alogaritma berbasis matematika untuk menjadwalkan sekelompok aktivitas proyek. CPM merupakan salah satu peralatan terpenting untuk manajemen proyek. Critical Path Method dikembangkan tahun 1950-an oleh Morgan R. Walker dari DuPont dan James E. Kelley, Jr. dari Remington Rand. Keduanya bekerjasama mengembangkan CPM pada tahun 1989. Di saat yang hampir bersamaan, Booz Allen Hamilton dan angkatan laut AS juga mengembangkan Program Evaluation and Review Technique.

CPM merupakan model kegiatan proyek yang digambarkan dalam bentuk jaringan. Kegiatan yang digambarkan sebagai titik pada jaringan dan peristiwa yang menandakan awal atau akhir dari kegiatan digambarkan sebagai busur atau garis antara titik. Komponen-komponen dalam metode CPM adalah:

- a. Diagram Network
- b. Hubungan antar simbol dan urutan kegiatan
- c. Jalur kritis
- d. Tenggang waktu kegiatan

e. Limit jadwal kegiatan

Secara teoritis yaitu perhitungan dengan menggunakan metode Critical Path Methode (CPM) berdasarkan data pembangunan. Langkah-langkah dalam perencanaan proyek menggunakan metode CPM :

1. Tentukan rincian kegiatan. Dari rincian kegiatan yang harus dilakukan dalam sebuah proyek, tambahkan informasi durasi dan identifikasikan prasyarat kegiatan sebelumnya yang harus terselesaikan terlebih dahulu.
2. Tentukan urutan kegiatan dan gambarkan dalam bentuk jaringan. Beberapa kegiatan akan dapat dimulai dengan sangat tergantung pada penyelesaian kegiatan lain. Relasi antar kegiatan ini harus diidentifikasi dan digambarkan secara berurutan dalam bentuk titik dan busur.
3. Susun perkiraan waktu penyelesaian untuk masing-masing kegiatan. Waktu yang dibutuhkan untuk menyelesaikan setiap kegiatan dapat diestimasi dengan menggunakan pengalaman masa lalu atau perkiraan dari para praktisi. CPM tidak memperhitungkan variasi waktu penyelesaian, sehingga hanya satu perkiraan yang akan digunakan untuk memperkirakan waktu setiap kegiatan.
4. Identifikasi jalur kritis (jalan terpanjang melalui jaringan). Jalur kritis adalah jalur yang memiliki durasi terpanjang yang melalui jaringan. Arti penting dari jalur kritis adalah bahwa jika kegiatan yang terletak pada jalur kritis tersebut tertunda, maka waktu penyelesaian proyek secara keseluruhan otomatis juga akan tertunda.
5. Pada jalur selain jalur kritis, akan ditemui waktu longgar/waktu toleransi

(slack time) yaitu sejumlah waktu sebuah kegiatan dapat ditunda tanpa menunda penyelesaian proyek secara keseluruhan.

6. Update Diagram CPM. Pada saat proyek berlangsung, waktu penyelesaian kegiatan dapat diperbarui sesuai dengan diperolehnya informasi dan asumsi baru. Sebuah jalur kritis baru mungkin akan muncul, dan perubahan bentuk jaringan sangat mungkin harus dilakukan.

Keterbatasan CPM adalah digunakannya satu angka perkiraan waktu penyelesaian bagi setiap kegiatan. Jika memang dibutuhkan perencanaan proyek yang lebih kompleks, metode PERT dengan tiga varian waktu perkiraan akan dapat memberikan alternatif perkiraan waktu penyelesaian proyek yang lebih terbuka.

Dalam proses identifikasi critical path method, dikenal beberapa terminology dan rumus-rumus perhitungan sebagai berikut:

1. ES (*Earliest Start Time*)

Waktu mulai paling awal suatu kegiatan. Bila waktu kegiatan dinyatakan atau berlangsung dalam jam, maka waktu ini adalah jam paling awal kegiatan dimulai.

2. EF (*Earliest Finish Time*)

Waktu selesai paling awal suatu kegiatan. Bila hanya ada satu kegiatan terdahulu, maka EF suatu kegiatan terdahulu merupakan ES kegiatan berikutnya.

3. LS (*Latest Allowable Start Time*)

Waktu paling akhir kegiatan boleh mulai yaitu waktu paling akhir kegiatan boleh dimulai tanpa memperlambat proyek secara keseluruhan.

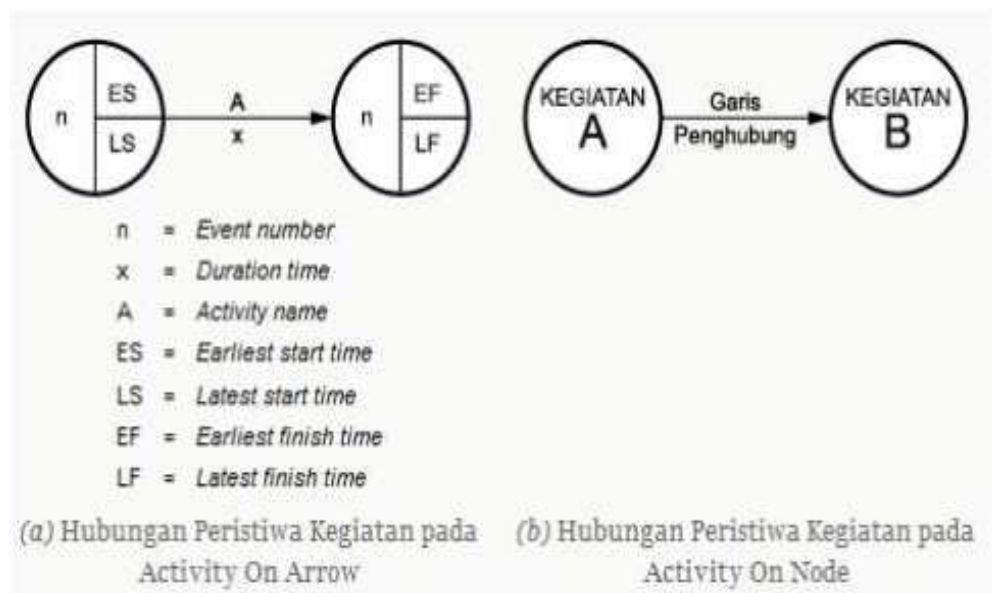
4. LF (*Latest Allowable Finish Time*)

Waktu paling akhir kegiatan boleh selesai (*Latest Allowable Finish Time*) tanpa memperlambat penyelesaian proyek.

5. D (*Duration*)

Adalah kurun waktu suatu kegiatan. Umumnya dengan satuan waktu hari, minggu, bulan dan lain lain.

Di dalam teknik CPM terdapat 2 cara untuk menggambarkan diagram aliran kerja yaitu metode menggambarkan kegiatan pada node disebut metode diagram AON (*Activity On Node*) sedangkan metode kegiatan menggambarkan pada garis disebut metode diagram AOA (*Activity On Arrow*) dapat dilihat perbedaannya dari gambar 2.1, sebagai berikut :



Dimana diagram AOA menunjukan waktu kegiatan ada pada garis sedangkan diagram AON menunjukan kegiatan pada titik. Ketentuan lain dari Critical Path adalah :

1. Jalur kritis juga diperkenankan melalui *dummy*.

2. Jalur kritis tidak perlu hanya terdiri dari satu jalur, tetapi boleh terdiri dari atau lebih jalur [15].

Waktu penyelesaian satu kegiatan tidak boleh melebihi waktu yang sudah ditentukan, karena keterlambatan kegiatan kritis dapat mengganggu (memperpanjang) waktu penyelesaian seluruh proyek. Langkah-langkah pembuatan CPM (*Critical Path Method*), yaitu :

1. Pahami urutan dari masing-masing kegiatan atau pekerjaan tersebut dan ketergantungannya antar masing-masing kegiatan/pekerjaan yang bersangkutan.
2. Rangkaikan satu jaringan atau persyaratan seperti yang dijelaskan.
3. Ingat kegiatan mana yang harus mendahului kegiatan lain dan mana yang merupakan lanjutan dari kegiatan sebelumnya.
4. Jika jumlah kegiatannya banyak mencapai ratusan kegiatan maka untuk memudahkan penyusunan CPM bisa dikerjakan dengan mengikuti urutan pekerjaan dari masing-masing kelompok pekerjaan.
5. CPM dari *work item group* yang sudah jadi lalu di gabungkan dengan work item pekerjaan yang dibuat sendiri [15].

Algoritma metode jalur kritis atau critical path method (CPM) memiliki rumus sebagai berikut.

Perhitungan Maju

Digunakan untuk mengetahui waktu paling awal peristiwa (*Node/Event*) dapat terjadi atau *Erliest Time of Occurance* (TE). Dalam perhitungan ini terdapat [16] :

- Waktu Mulai Paling Awal (ES).

$$ES(j) = \{ES(i) + D(ij)\}$$

- Waktu Selesai Paling Awal (EF).

$$EF(i,j) = \{ES(i) + D(I,j)\}$$

Perhitungan Mundur

Digunakan untuk mengetahui waktu paling akhir peristiwa boleh terjadi atau *Latest Allowable Event / Occurance Time* (TL). Dalam perhitungan ini terdapat:

- Waktu paling akhir kegiatan dimulai (LS).

$$LS(i) = \{LF(i) - D(ij)\}$$

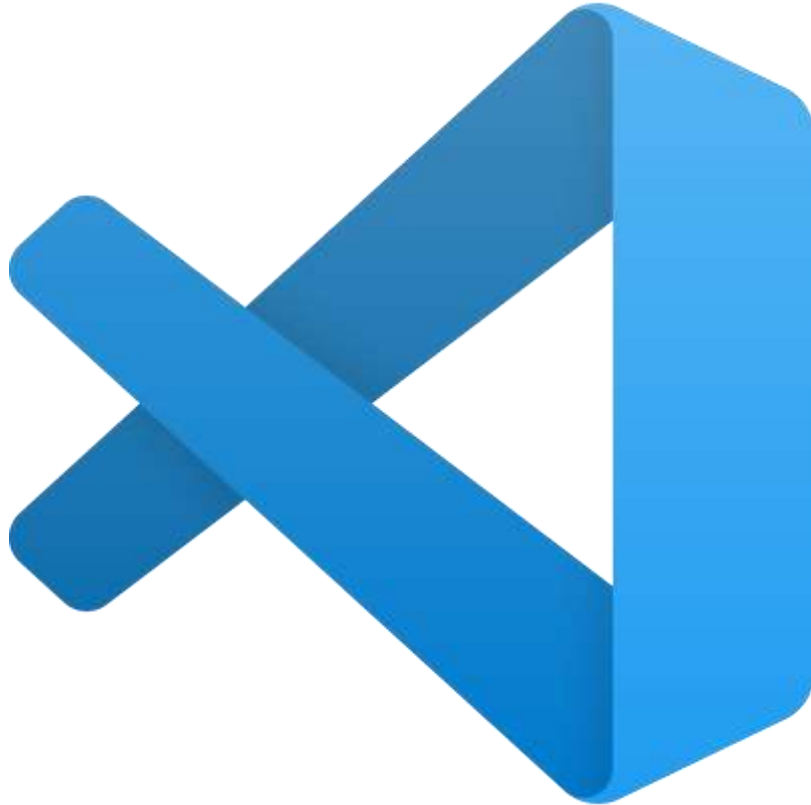
- Waktu Paling Akhir Kegiatan Selesai (LF).

$$LF(I,j) = LS(I,j) + D(I,j)\}$$

3. Tools yang Digunakan

3.1 Pengenalan Tools yang Digunakan

3.1.1 Visual Studio Code



Microsoft Visual Studio by merupakan sebuah perangkat lunak lengkap (suite) yang dapat digunakan untuk melakukan pengembangan aplikasi, baik itu aplikasi bisnis, aplikasi personal, ataupun komponen aplikasinya, dalam bentuk aplikasi console, aplikasi Windows, ataupun aplikasi Web. Visual Studio mencakup kompiler, SDK, Integrated Development Environment (IDE), dan dokumentasi (umumnya berupa MSDN Library). Kompiler yang dimasukkan ke dalam paket Visual Studio antara lain Visual C++, Visual C#, Visual Basic, Visual Basic .NET, Visual InterDev, Visual J++, Visual J#, Visual FoxPro, dan Visual SourceSafe.

Microsoft Visual Studio dapat digunakan untuk mengembangkan aplikasi dalam native code (dalam bentuk bahasa mesin yang berjalan di atas Windows) ataupun managed code (dalam bentuk Microsoft Intermediate Language di atas .NET Framework). Selain itu, Visual Studio juga dapat digunakan untuk mengembangkan aplikasi Silverlight, aplikasi Windows Mobile (yang berjalan di atas .NET Compact Framework).

3.1.2 Xampp

XAMPP adalah perangkat lunak bebas, yang mendukung banyak system operasi, merupakan kompilasi dari beberapa program. XAMPP merupakan tool yang menyediakan paket perangkat lunak ke dalam satu buah paket. Dengan menginstall XAMPP maka tidak perlu lagi melakukan instalasi dan konfigurasi web server Apache, PHP dan MySQL secara manual. XAMPP akan menginstallasi dan mengkonfigurasikannya secara otomatis untuk anda atau auto konfigurasi.

XAMPP merupakan salah satu paket installasi Apache, PHP dan MySQL instant yang dapat kita gunakan untuk membantu proses installasi ketiga produk tersebut. Selain paket installasi instant XAMPP versi 1.6.4 juga memberikan fasilitas pilihan penggunaan PHP4 atau PHP5. Untuk berpindah versi PHP yang ingin digunakan juga sangat mudah dilakukan dengan menggunakan bantuan PHP-Switch yang telah disertakan oleh XAMPP, dan yang terpenting XAMPP bersifat free atau gratis untuk digunakan.

XAMPP merupakan tool yang menyediakan paket perangkat lunak ke dalam satu buah paket. Dengan menginstall XAMPP maka tidak perlu lagi melakukan instalasi dan konfigurasi web server Apache, PHP dan MySQL secara manual. XAMPP akan menginstallasi dan mengkonfigurasikannya secara otomatis.

XAMPP merupakan pengembangan dari LAMP (Linux Apache, MySQL, PHP and PERL), XAMPP ini merupakan project non-profit yang dikembangkan oleh Apache Friends yang didirikan Kai 'Oswald' Seidler dan Kay Vogelgesang pada tahun 2002, project mereka ini bertujuan mempromosikan penggunaan Apache web server.

Fungsi XAMPP sendiri adalah sebagai server yang berdiri sendiri (localhost), yang terdiri beberapa program antara lain : Apache HTTP Server, MySQL database, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl. Nama XAMPP sendiri merupakan singkatan dari X (empat sistem operasi apapun), Apache, MySQL, PHP dan Perl.

Program ini tersedia dalam GNU General Public License dan bebas, merupakan web server yang mudah untuk digunakan yang dapat menampilkan halaman web yang dinamis. Untuk mendapatkannya XAMPP anda dapat mendownload langsung dari web resminya. Dan berikut beberapa definisi program lainnya yang terdapat dalam XAMPP.

Berikut ini terdapat beberapa fitur-fitur xampp, terdiri atas:

1. Apache

Apache adalah perangkat lunak sumber terbuka yang menjadi alternatif dari server web Netscape. Server HTTP Apache atau Server Web/WWW Apache merupakan server web yang dapat dijalankan di banyak sistem operasi yang berguna untuk melayani dan memfungsikan situs web. Apache dikembangkan oleh komunitas terbuka yang terdiri dari pengembang-pengembang dibawah naungan Apache Software Foundation.

2. MySQL

MySQL adalah singkatan "My Structured Query Language". Program ini berjalan sebagai server menyediakan multi-user mengakses ke sejumlah

database. MySQL umumnya digunakan oleh perangkat lunak bebas yang memerlukan fitur penuh sistem manajemen database, seperti WordPress, phpBB dan perangkat lunak lain yang dibangun pada perangkat lunak LAMP. Ia juga digunakan dalam skala sangat tinggi World Wide Web, termasuk produk-produk Google dan Facebook.

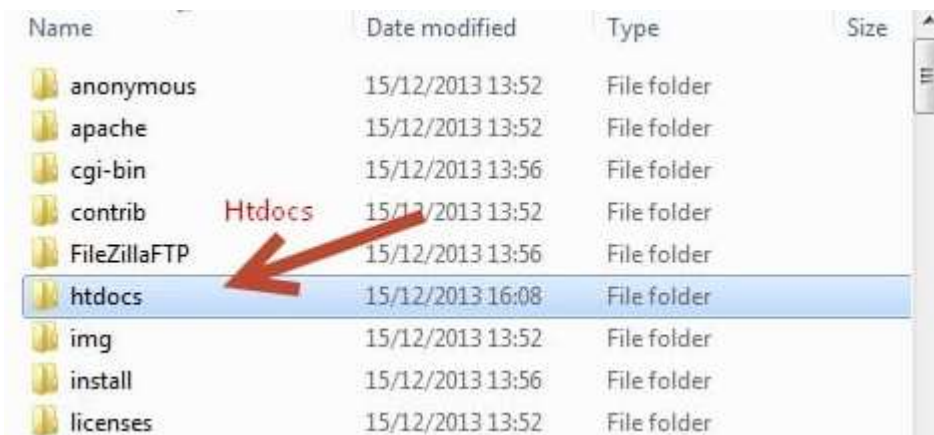
3. PHP

PHP adalah bahasa pemrograman script yang banyak dipakai untuk memprogram situs web dinamis, walaupun tidak tertutup kemungkinan digunakan untuk pemakaian lain. Contoh terkenal dari aplikasi PHP adalah phpBB dan MediaWiki (software di belakang Wikipedia). Contoh terkenal dari aplikasi PHP adalah phpBB dan MediaWiki (software di belakang Wikipedia).

Bagian-Bagian XAMPP

Berikut ini terdapat beberapa bagian-bagian XAMPP, terdiri atas:

- Htdocs

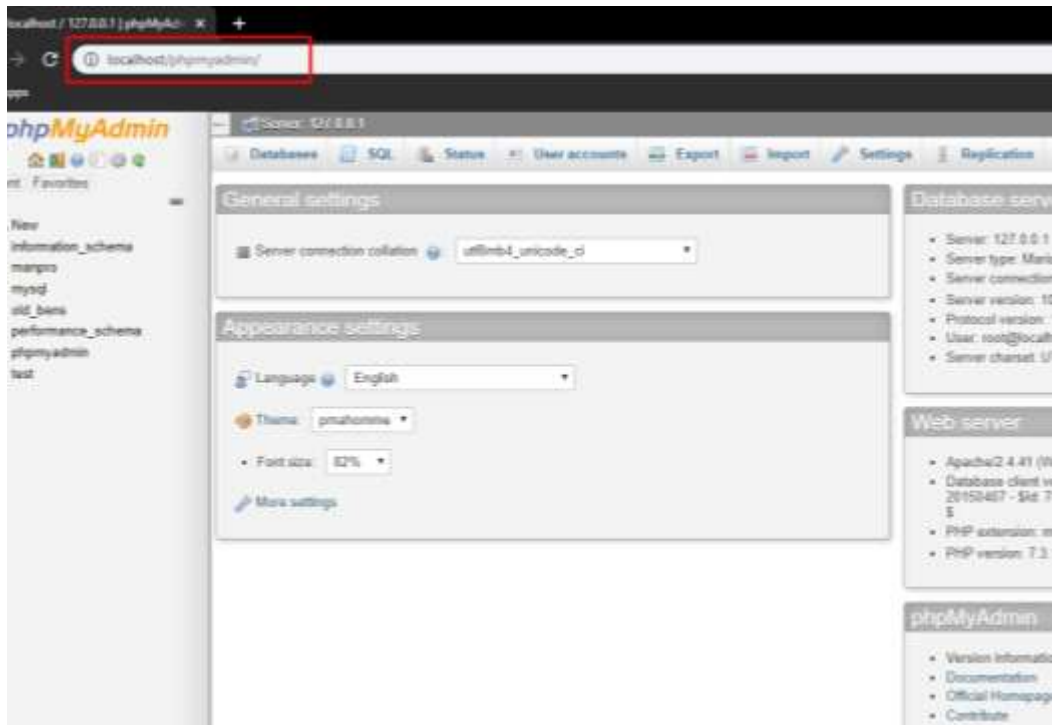


Name	Date modified	Type	Size
anonymous	15/12/2013 13:52	File folder	
apache	15/12/2013 13:52	File folder	
cgi-bin	15/12/2013 13:56	File folder	
contrib	15/12/2013 13:52	File folder	
FileZillaFTP	15/12/2013 13:56	File folder	
htdocs	15/12/2013 16:08	File folder	
img	15/12/2013 13:52	File folder	
install	15/12/2013 13:56	File folder	
licenses	15/12/2013 13:52	File folder	

Htdocs adalah sebuah folder yang digunakan sebagai tempat penyimpanan berkas seperti PHP, HTML, dan script lain yang digunakan

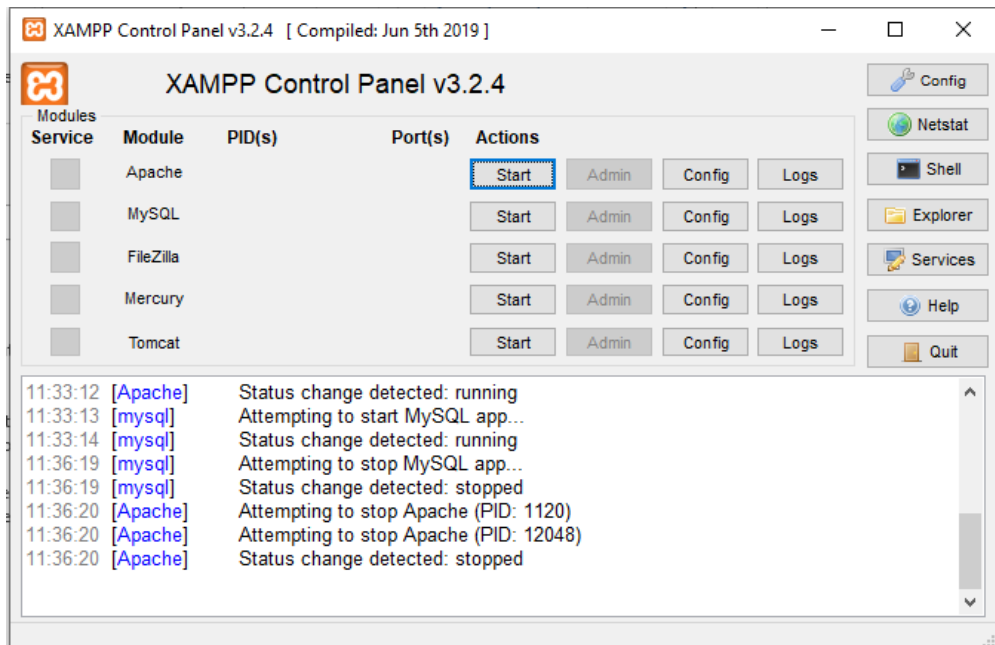
dalam sebuah halaman website. Secara kapasitas penyimpanan, XAMPP tergantung dari seberapa besar kapasitas hardisk di laptop atau komputer anda. Sedangkan bila menggunakan hosting online, maka tergantung pilihan waktu membeli sebuah hosting.

- phpMyadmin



phpMyadmin adalah sebuah tempat yang digunakan untuk mengelola database MySQL yang berada di komputer atau laptop. Untuk mengakses phpMyadmin yakni dengan membuka browser internet (Mozilla atau chrome) lalu ketikkan alamat <http://localhost/phpMyadmin> maka akan muncul tampilannya.

- Control Panel



Control Panel adalah sebuah layanan untuk mengelola XAMPP baik itu mengontrol (start atau stop XAMPP) serta layanan service lainnya. Secara online di dalam hosting atau VPS dikenal CPanel

Kelebihan dan Kekurangan XAMPP

Berikut ini terdapat beberapa kelebihan dan kekurangan XAMPP, terdiri atas:

Kelebihan :

1. Database Storage Engine ini banyak digunakan oleh programmer apalagi oleh web developer karena sifatnya yang free. Untuk yang expert sudah ada yang bayar.
2. Kemampuannya sudah bisa diandalkan, mempunyai kapasitas yang cukup mumpuni sekitar 60.000 tabel dengan jumlah record mencapai 5.000.000.000 bahkan untuk yang terbaru sudah lebih.
3. Keamanan datanya cukup aman walaupun tidak sehebat Postgre apalagi Oracle.

4. Engine ini multiplatform sehingga mampu diaplikasikan di berbagai sistem operasi. My Sql cocok diaplikasikan diaplikasi kelas kecil dan menengah.
5. Kelebihan paling utama engine ini adalah kecepatannya.

Kekurangan :

1. Tidak cocok untuk menangani data dengan jumlah yang besar, baik untuk menyimpan data maupun untuk memproses data.
2. Memiliki keterbatasan kemampuan kinerja pada server ketika data yang disimpan telah melebihi batas maksimal kemampuan daya tampung server karena tidak menerapkan konsep Technology Cluste.

XAMPP adalah singkatan yang setiap huruf adalah :

- **X** : Program ini dapat dijalankan di banyak sistem operasi, seperti Windows, Linux, Mac OS, dan Solaris.
- **A : Apache**, server aplikasi Web. Apache tugas utama adalah untuk menghasilkan halaman web yang benar kepada pengguna terhadap kode PHP yang sudah dituliskan oleh pembuat halaman web. jika perlu kode PHP juga berdasarkan yang tertulis, dapat database diakses dulu (misalnya MySQL) untuk mendukung halaman web yang dihasilkan.
- **M : MySQL**, server aplikasi database. Pertumbuhannya disebut SQL singkatan dari Structured Query Language. SQL merupakan bahasa terstruktur yang difungsikan untuk mengolah database. MySQL dapat digunakan untuk membuat dan mengelola database dan isinya. Bisa juga memanfaatkan MySQL guna untuk menambahkan, mengubah, dan menghapus data dalam database.
- **P : PHP**, bahasa pemrograman web. Bahasa pemrograman PHP adalah bahasa pemrograman untuk membuat web yang server-side scripting. PHP digunakan untuk membuat halaman web dinamis. Sistem manajemen database yang sering digunakan dengan PHP adalah MySQL. namun PHP juga mendukung Pengelolaan sistem database Oracle, Microsoft Access, Interbase, d-base, PostgreSQL, dan sebagainya.

- **P : Perl**, bahasa pemrograman untuk semua tujuan, pertama kali dikembangkan oleh Larry Wall, mesin Unix. Perl dirilis pertama kali tanggal 18 Desember 1987 yang ditandai dengan keluarnya Perl 1. Pada versi-versi selanjutnya, Perl juga tersedia untuk berbagai sistem operasi Unix (SunOS, Linux, BSD, HP-UX), juga tersedia untuk sistem operasi seperti DOS, Windows, PowerPC, BeOS, VMS, EBCDIC, dan PocketP

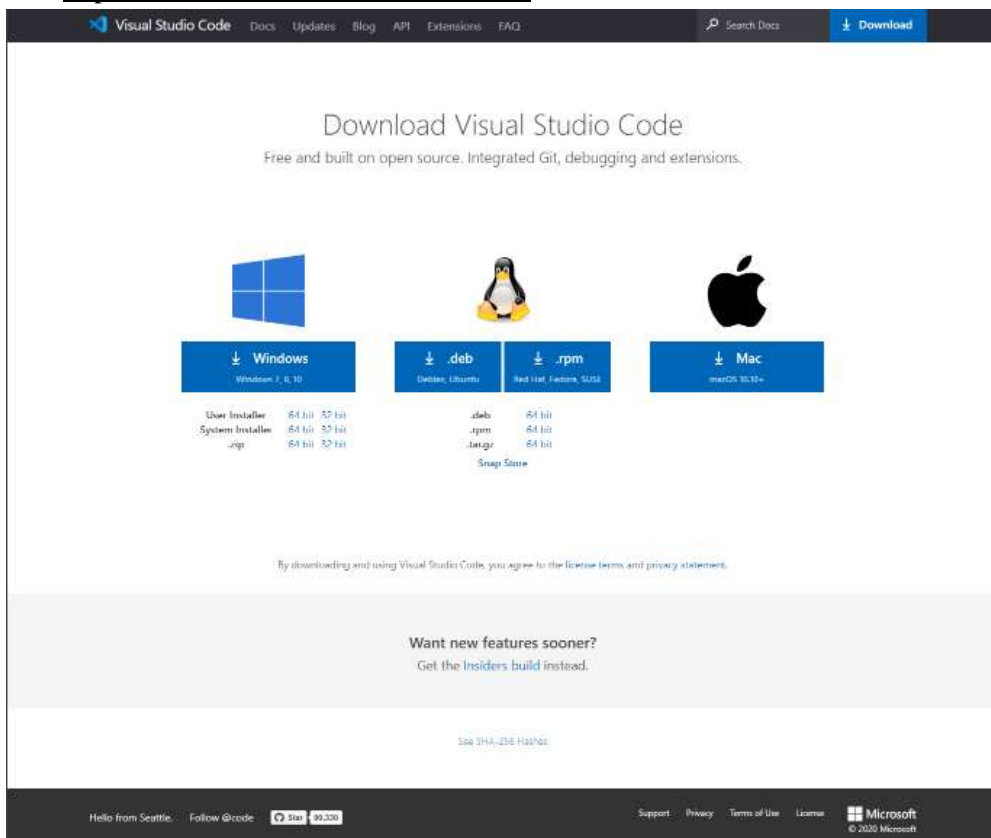
3.2 Instalasi Tools yang Digunakan

3.2.1 Instalasi Visual Studio Code

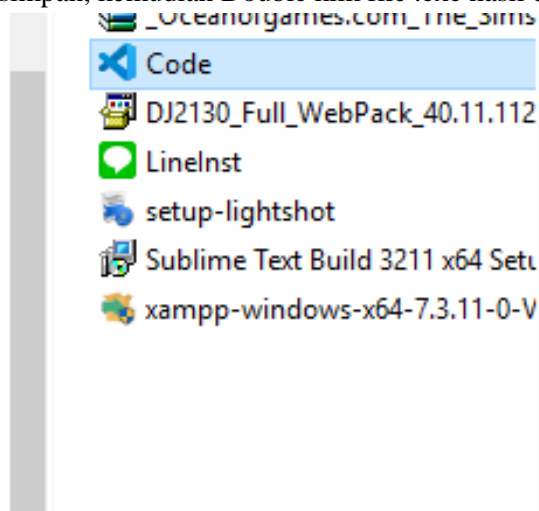
Berikut adalah langkah-langkah untuk melakukan instalasi Visual Studio Code:

1. Tahapan yang pertama untuk melakukan instalasi maka download dahulu file .exe pada link berikut :

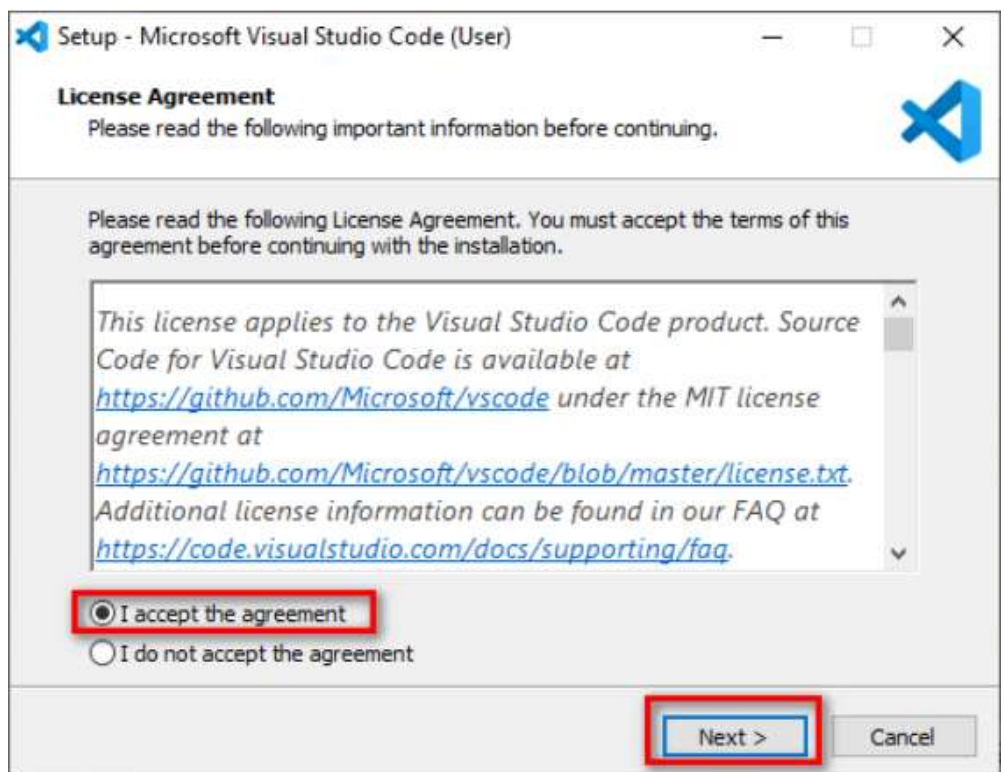
<https://code.visualstudio.com/download>



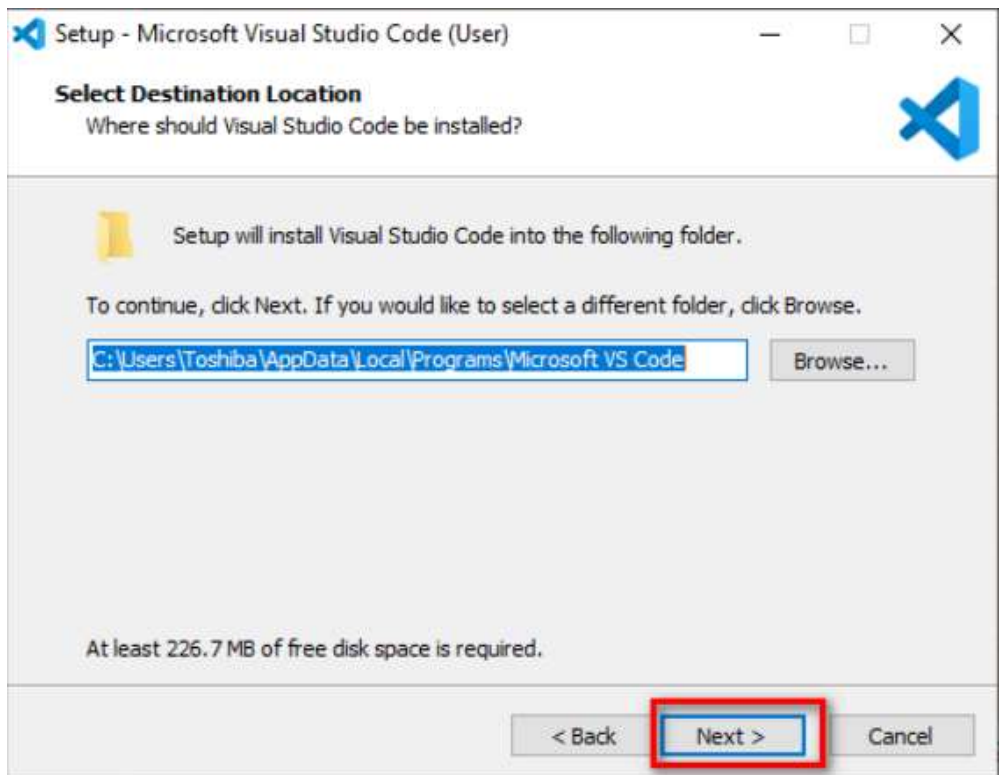
2. Tahapan yang kedua setelah selesai mendownload maka buka directory dimana file .exe tersimpan, kemudian Double klik file .exe hasil download tadi :



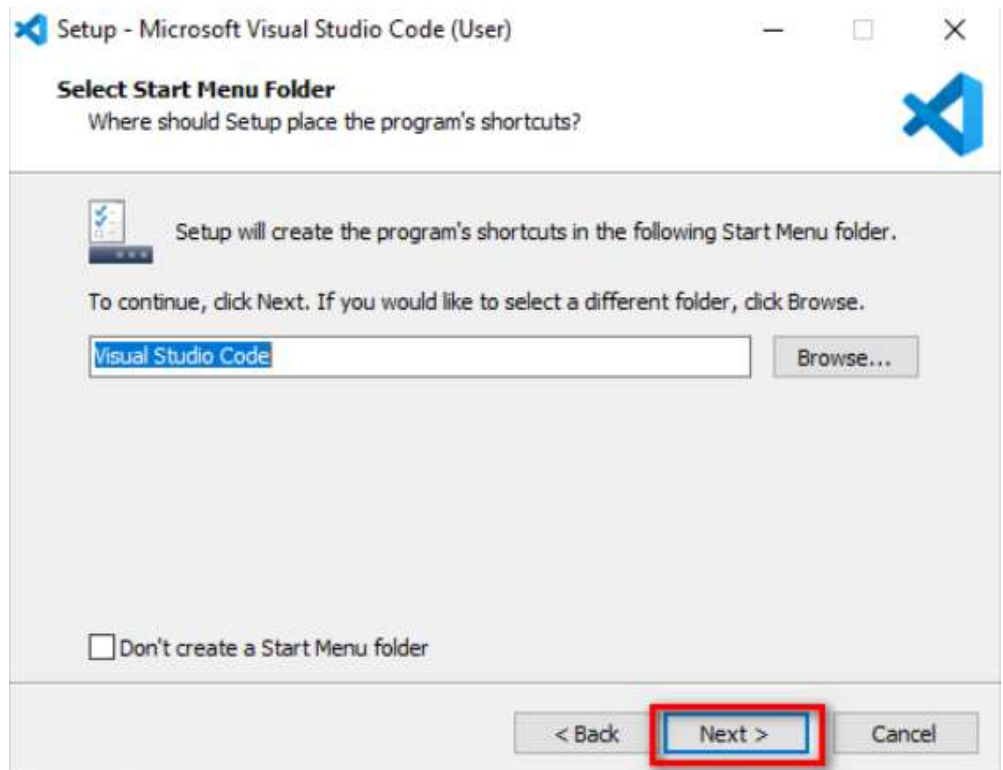
3. Tahapan ketiga jika sudah double klik file .exe maka akan muncul tampilan berikut kemudian klik next :



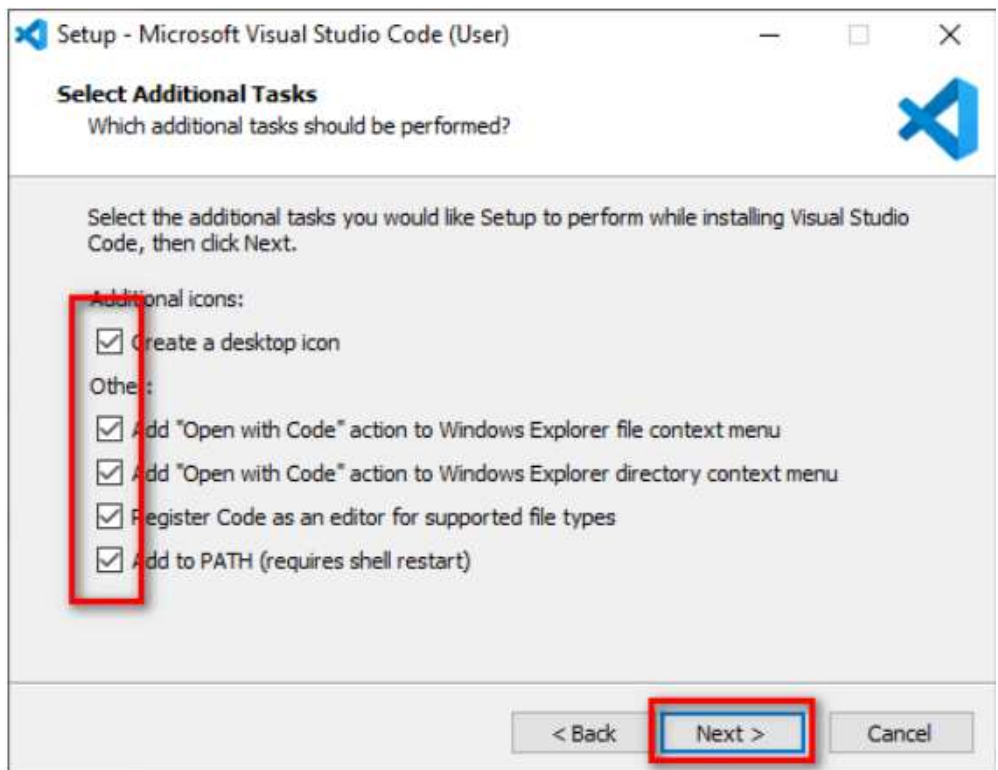
4. Untuk Select Destination Location bisa di biarkan saja jika lokasi instalasi tidak akan di rubah. Klik Next.



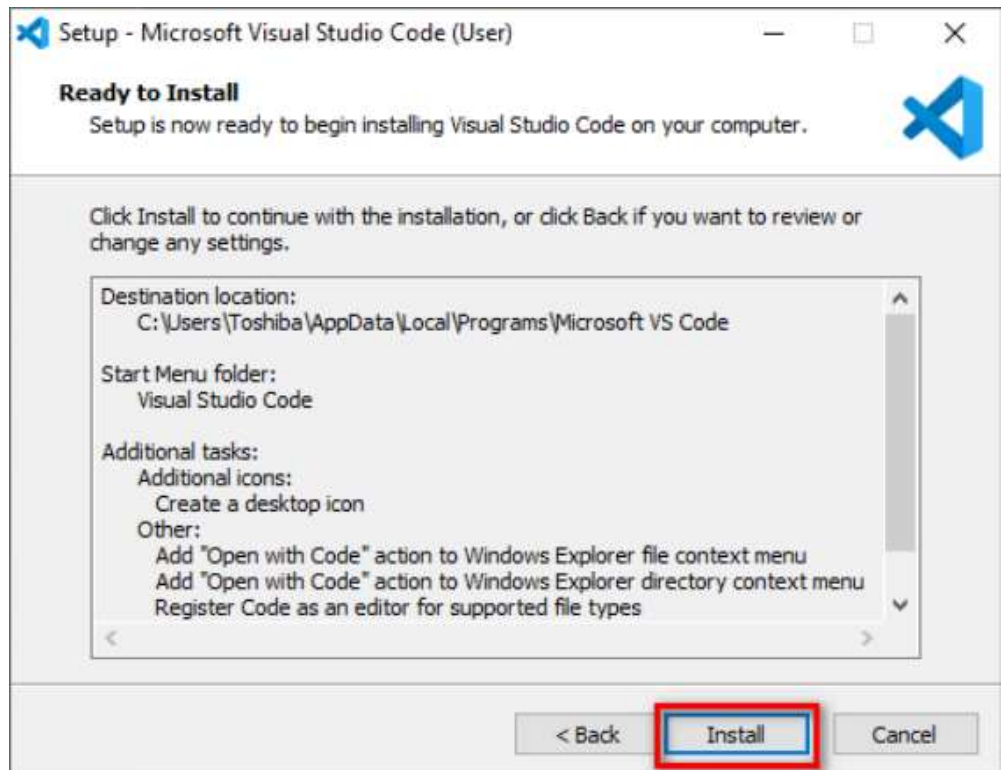
5. Klik Next lagi jika tidak akan merubah Start Menu Folder.



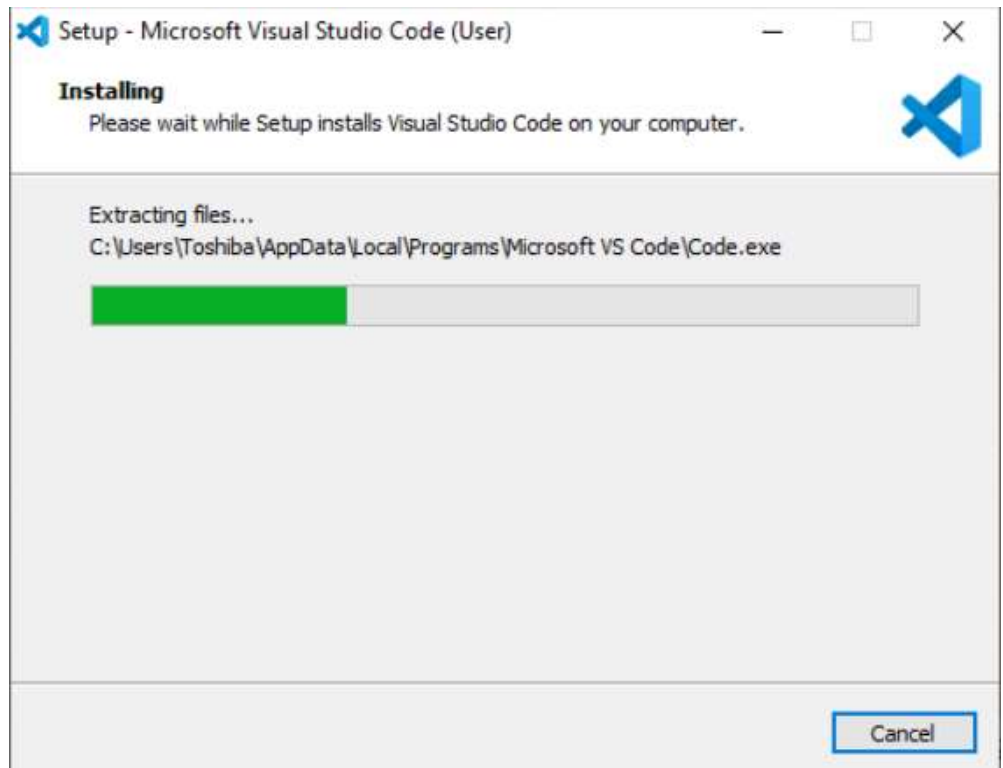
6. Di bagian Select Additional Tasks centang semua. Kemudian Next.



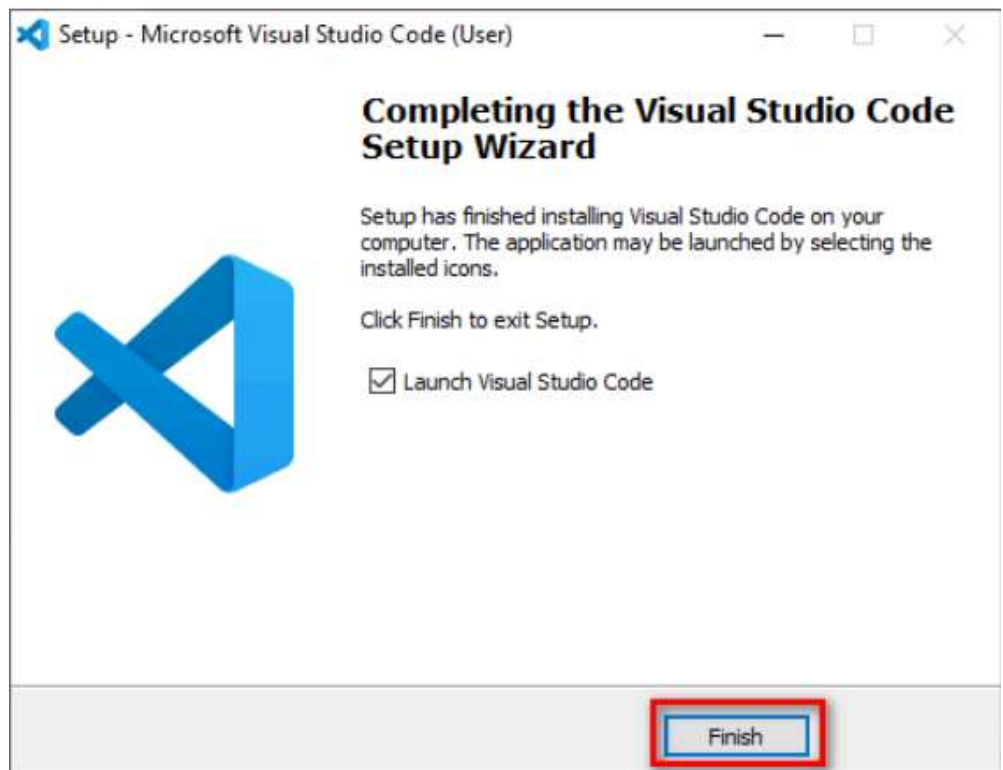
7. Lalu klik Install untuk memulai proses installasi.



8. Tunggu sampai proses instalasi selesai.



9. Setelah selesai klik Finish.



3.2.2 Instalasi XAMPP

1. Download aplikasi XAMPP terbaru
<https://www.apachefriends.org/download.html> . Pilih salah satu



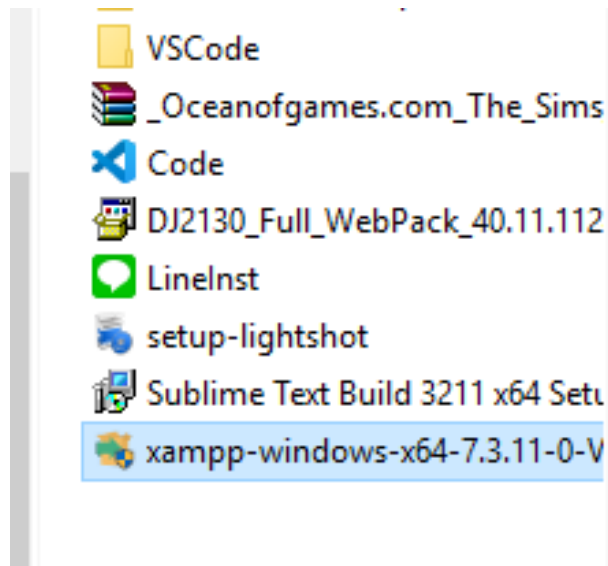
XAMPP for Windows 5.6.36, 7.0.30, 7.1.18 & 7.2.6

Version	Checksum	Size
5.6.36 / PHP 5.6.36 What's Included?	md5 sha1	Download (32 bit) 110 Mb
7.0.30 / PHP 7.0.30 What's Included?	md5 sha1	Download (32 bit) 121 Mb
7.1.18 / PHP 7.1.18 What's Included?	md5 sha1	Download (32 bit) 121 Mb
7.2.6 / PHP 7.2.6 What's Included?	md5 sha1	Download (32 bit) 123 Mb

[Requirements](#) [Add-ons](#) [More Downloads »](#)

Windows XP or 2003 are not supported. You can download a compatible version of XAMPP for these platforms [here](#).

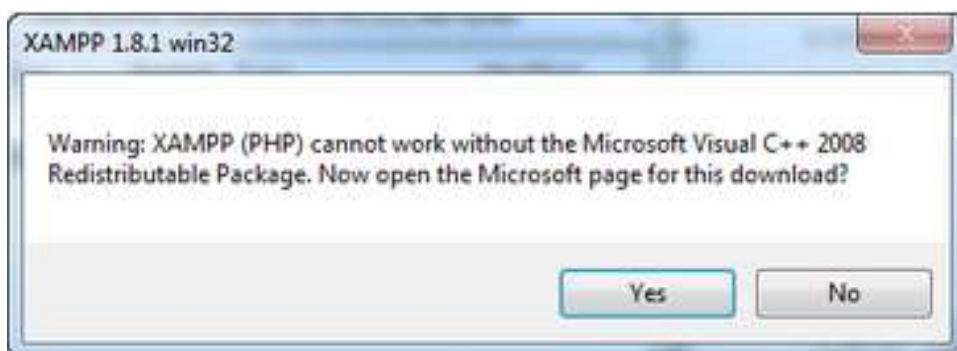
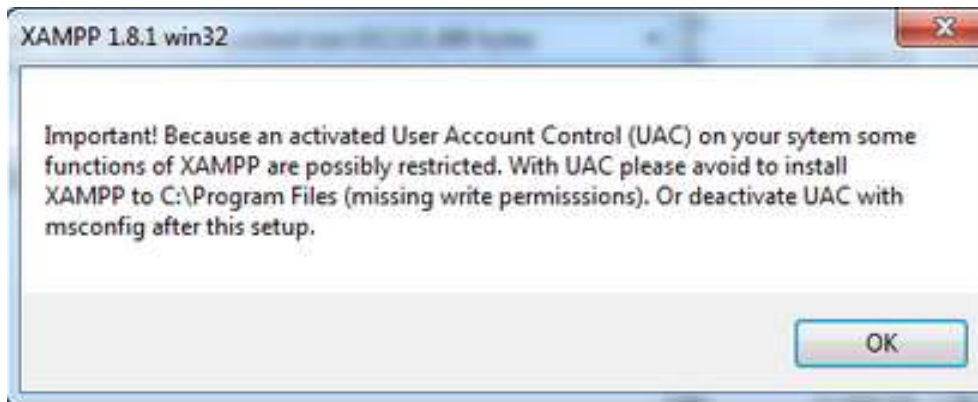
2. Tahapan yang kedua setelah selesai mendownload maka buka directory dimana file .exe tersimpan, kemudian Double klik file .exe hasil download tadi :



3. Dobel klik file XAMPP yang baru saja Anda download, nanti selanjutnya akan muncul jendela “installer language” seperti di bawah ini:



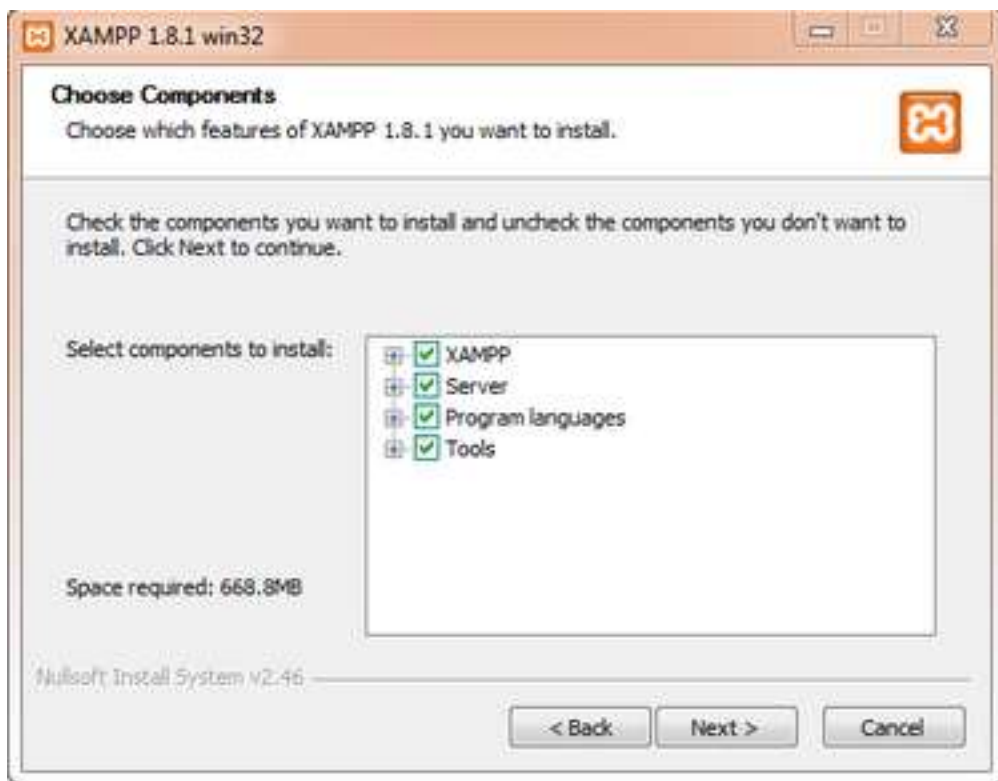
4. Selanjutnya pilih bahasa. Pilih yang Bahasa Inggris (English). Klik OK.
5. Kadang pada proses ini muncul pesan error. Jika ada, abaikan saja dan lanjutkan dengan klik OK dan YES.



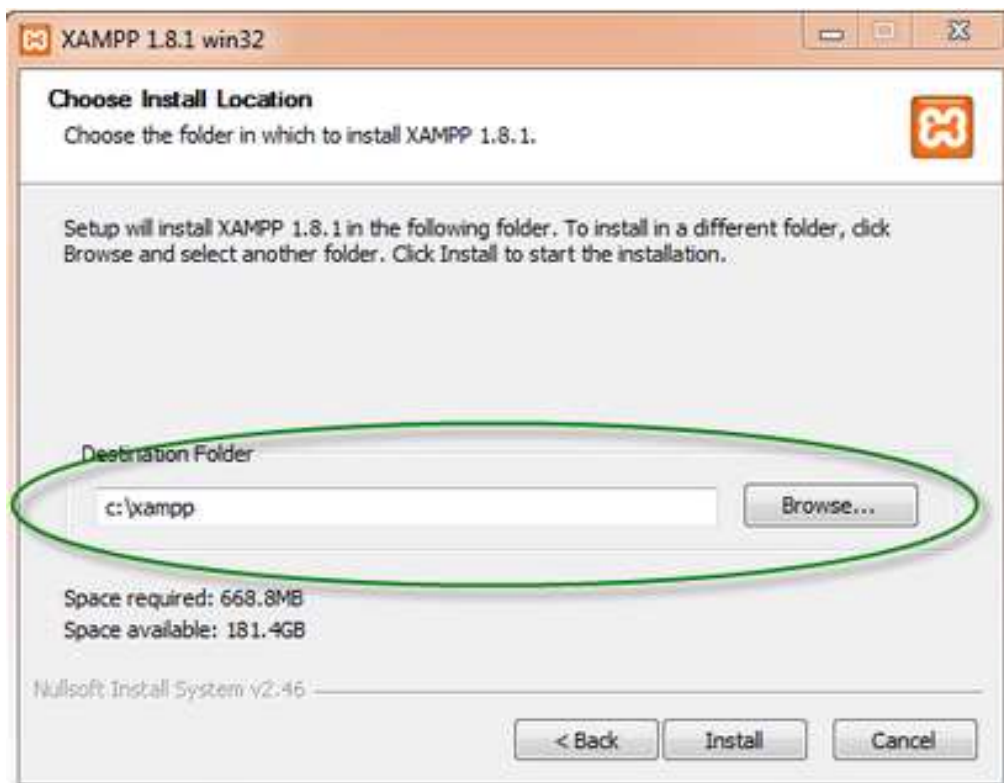
6. Berikutnya akan muncul jendela yang isinya meminta Anda menutup semua aplikasi yang sedang berjalan. Jika semua aplikasi sudah ditutup, maka klik tombol **Next.**



7. Selanjutnya Anda akan diminta untuk memilih aplikasi yang mau diinstal. Centang saja semua pilihan dan klik tombol **Next**.



8. Kemudian Anda akan diminta untuk menentukan lokasi folder penyimpanan file-file dan folder XAMPP. Secara default akan diarahkan ke lokasi **c:\xampp**. Namun jika Anda ingin menyimpannya di folder lain bisa klik **browse** dan tentukan secara manual folder yang ingin digunakan. Jika sudah selesai, lanjutkan dan klik tombol **Install**.



9. Tunggu beberapa menit hingga proses instalasi selesai. Jika sudah muncul jendela seperti di bawah ini, klik tombol **Finish** untuk menyelesaikannya.



10. Berikutnya, akan muncul jendela dialog seperti gambar di bawah ini yang menanyakan Anda apakah mau langsung menjalankan aplikasi XAMPP atau tidak. Jika ya, maka klik YES.

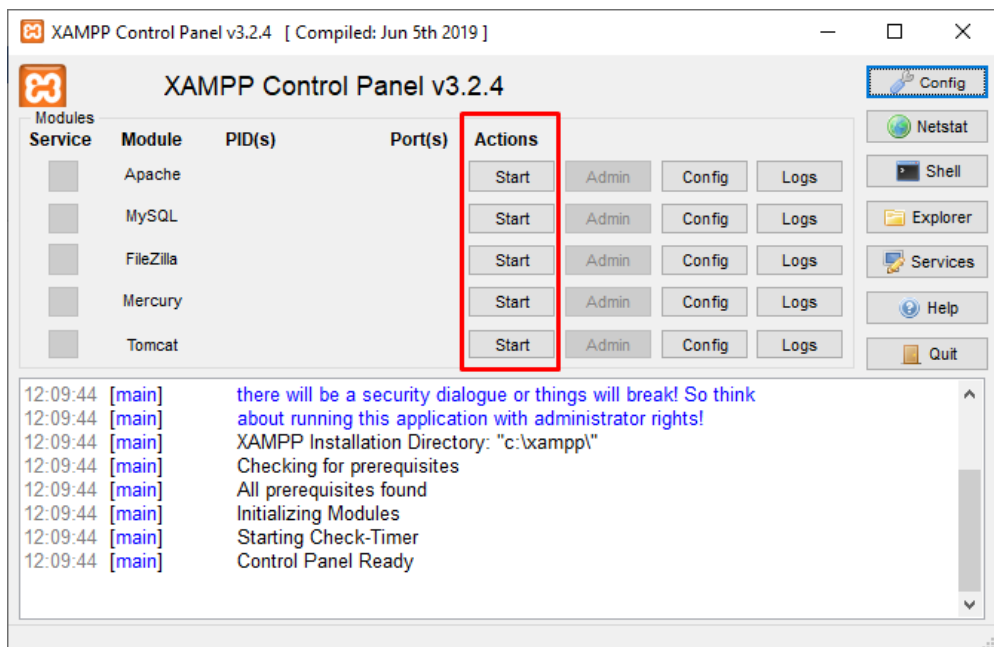


3.2.3 Menjalankan Aplikasi XAMPP

1. Bukalah aplikasi XAMPP, bisa melalui **Start Menu** atau **Desktop**, dan klik icon XAMPP. Atau, jika Anda membukanya begitu proses instalasi selesai maka klik **Yes** seperti yang terlihat pada gambar di atas.



2. Setelah terbuka, silahkan klik tombol **Start** pada kolom **Action** sehingga tombol tersebut berubah menjadi **Stop**. Dengan mengklik tombol tersebut, artinya itulah aplikasi yang dijalankan. Biasanya jika saya menggunakan XAMPP, yang saya start hanyalah aplikasi Apache dan MySQL, karena saya tidak memerlukan aplikasi seperti Filezilla, dan lain-lain.



5.1 Analisis

Analisis adalah tahap yang sangat penting karena suatu kesalahan dalam tahap ini akan mempengaruhi pada tahap berikutnya. Penelitian juga membuktikan bahwa kesalahan yang diperbaiki setelah tahap analisis akan memakan biaya yang lebih besar dari pada jika di perbaiki saat dilakukan analisis.

Analisis ini akan dilakukan terhadap Sistem Manajemen Proyek Menggunakan Critical Path Method Untuk Memprediksi Durasi Total Pengerjaan Proyek di PT. Javan Cipta Solusi. Aplikasi ini dibuat menggunakan UML yang memberikan gambaran tentang proses yang terjadi dalam aplikasi yang dapat berupa suatu proses untuk mendapatkan suatu informasi yang dicari.

5.1.1 Analisis Sistem yang Sedang Berjalan

5.1.1.1. Analisis Prosedur

A. Analisis Prosedur yang Sedang Berjalan

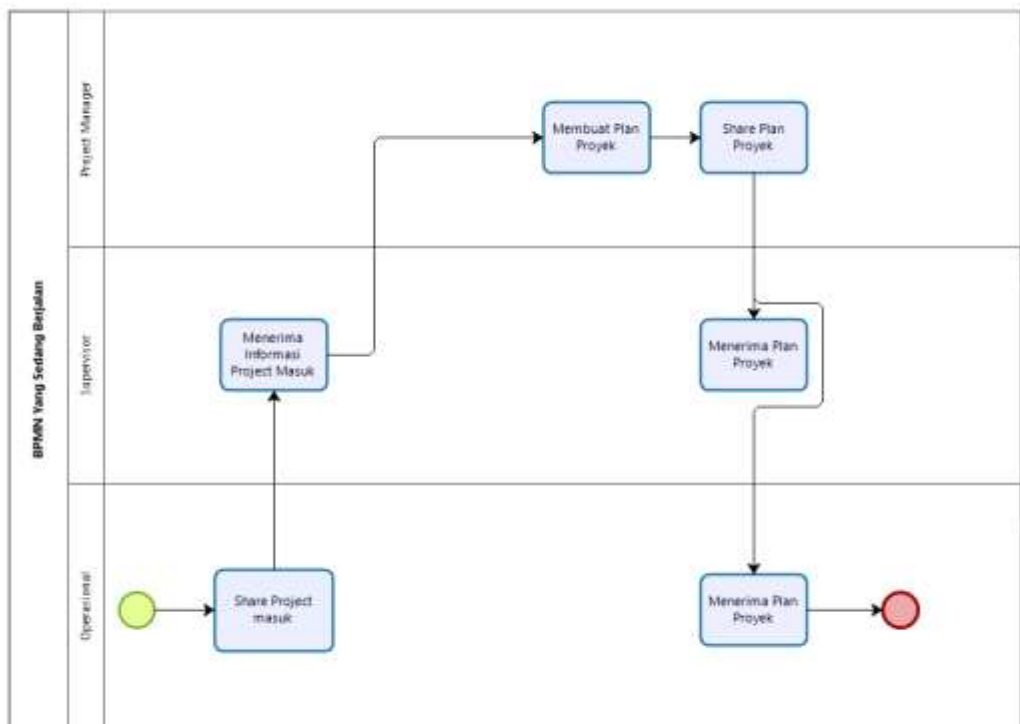
Prosedur yang ada pada perencanaan proyek yang sedang berjalan, yaitu sebagai berikut:

1. Operasional memberitahukan informasi terkait proyek yang masuk ke Supervisor.
2. Supervisor menerima informasi proyek masuk.
3. Supervisor akan memberitahukan informasi proyek dan SDM ke *Project Manager*.
4. Ketika *Project Manager* sudah mendapatkkn informasi tersebut, selanjutnya membuat plan proyek, pada pembuatan plan proyek ini

Project Manager dalam menentukan total durasi waktu hanya menambahkan semua durasi pada setiap modul, jadi pengerjaan proyek kurang bias dioptimalkan.

5. Setelah selesai membuat plan, *Project Manager* akan memberikan informasi terkait plan project tersebut.

Adapun BPMN prosedur penjadwalan proyek yang sedang berjalan, yaitu:



Gambar 5. 1BPMN yang Sedang Berjalan

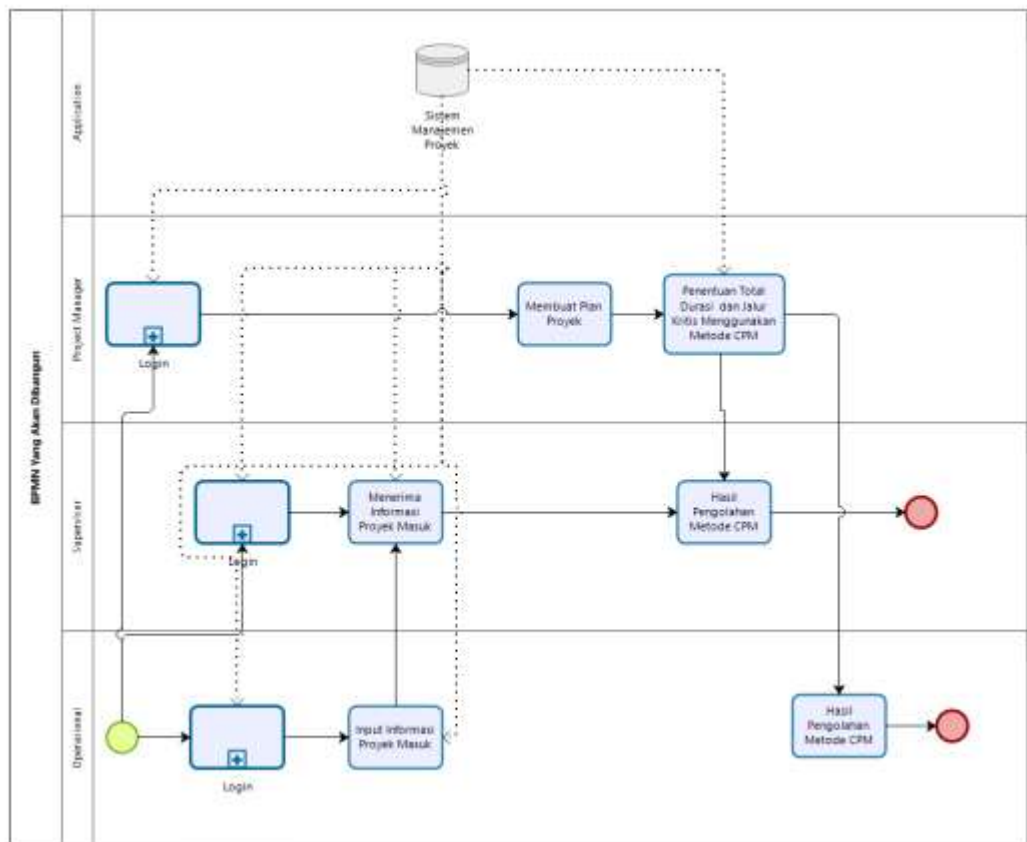
5.1.2 Analisis Sistem yang akan Dibangun

A. Analisis Sistem yang akan Dibangun pada Prosedur Perencanaan Proyek

Prosedur yang ada pada perencanaan proyek yang akan dibangun, yaitu sebagai berikut:

1. Operasional, Supervisor, dan *Project Manager* melakukan login.
2. Operasional input data proyek yang masuk.
3. Supervisor menerima notifikasi.
4. Setelah itu maka *Project Manager* akan membuat plan proyek dan melakukan perhitungan CPM untuk menentukan total durasi proyek dan jalur kritis untuk mengoptimalkan pengerjaan proyek.
5. Setelah sudah melakukan perhitungan maka Supervisor dan Operasional menerima update terkait proyek.

Adapun BPMN prosedur penjadwalan proyek yang akan dibangun adalah sebagai berikut:



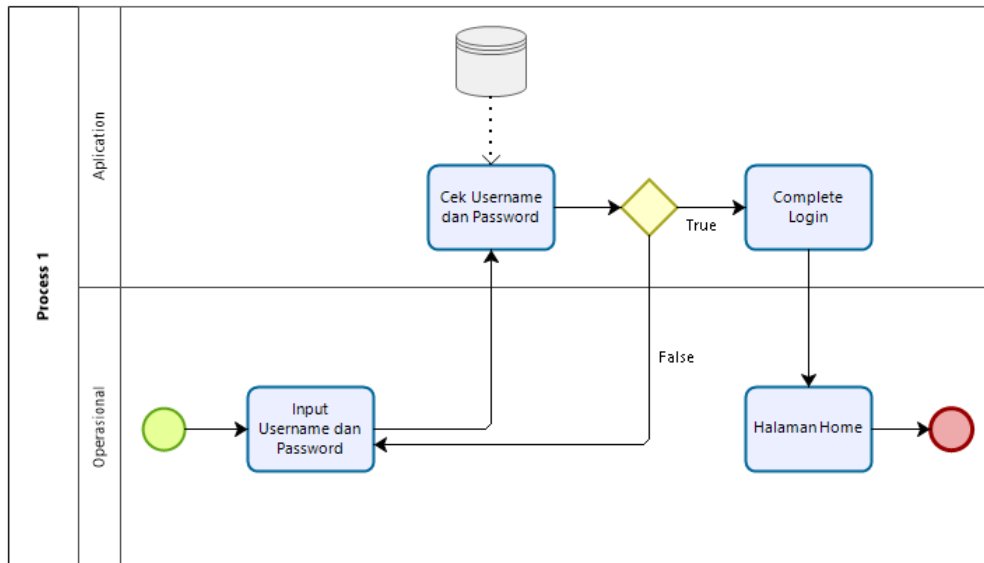
Gambar 5. 2 BPMN yang akan Dibangun

B. Analisis Login Operasional pada Sistem

Prosedur yang ada pada perencanaan login Operasional, yaitu sebagai berikut:

1. Operasional akses web.
2. Operasional menginputkan *username* dan *password*.
3. Sistem akan cek *username* dan *password*, jika sesuai dengan yang ada di database maka login berhasil dan dialihkan ke halaman home, dan ketika *username* dan *password* tidak sesuai dengan database maka login gagal dan Operasional mengulangi input *username* dan *password*.

Adapun BPMN prosedur *login* Operasional yang akan dibangun adalah sebagai berikut:



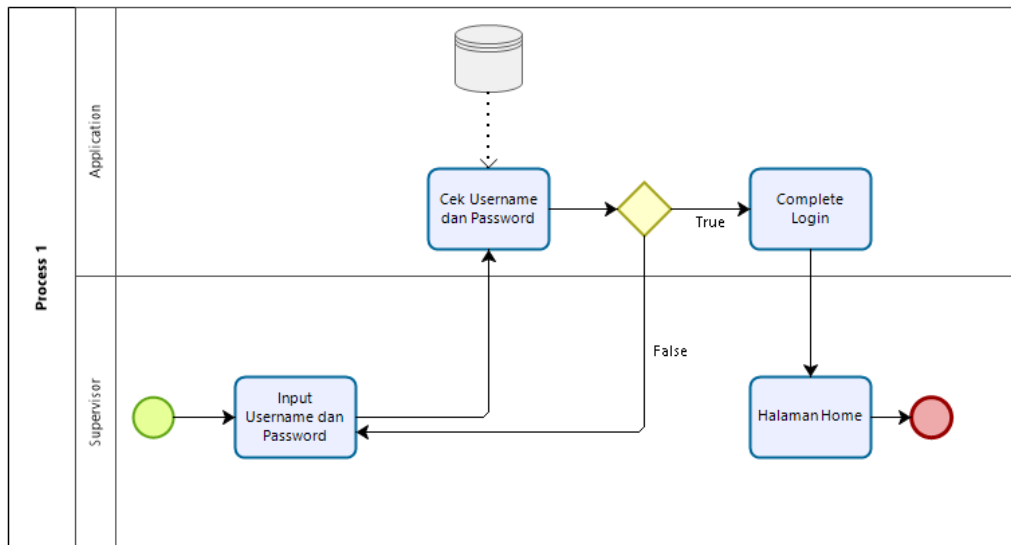
Gambar 5. 3 BPMN Login Operasional

C. Analisis Login Supervisor pada Sistem

Prosedur yang ada pada perencanaan login Supervisor, yaitu sebagai berikut:

1. Supervisor akses web.
2. Supervisor menginputkan *username* dan *password*.
3. Sistem akan cek *username* dan *password*, jika sesuai dengan yang ada di database maka login berhasil dan dialihkan ke halaman home, dan ketika *username* dan *password* tidak sesuai dengan database maka login gagal dan Supervisor mengulangi input *username* dan *password*.

Adapun BPMN prosedur *login* Supervisor yang akan dibangun adalah sebagai berikut:



Gambar 5. 4 BPMN Login Operasional

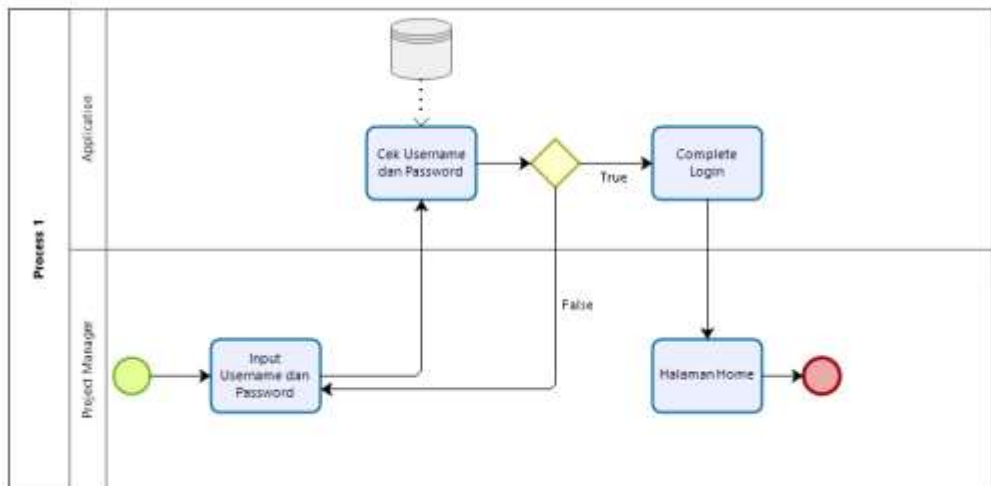
D. Analisis Login *Project Manager* pada Sistem

Prosedur yang ada pada perencanaan login Operasional, yaitu sebagai berikut:

1. *Project Manager* akses web.
2. *Project Manager* menginputkan *username* dan *password*.
3. Sistem akan cek *username* dan *password*, jika sesuai dengan yang ada di database maka login berhasil dan dialihkan ke halaman home, dan ketika *username* dan *password* tidak sesuai dengan database maka login gagal

dan *Project Manager* mengulangi input *username* dan *password*.

Adapun BPMN prosedur *login Project Manager* yang akan dibangun adalah sebagai berikut:



Gambar 5. 5 BPMN Login Operasional

5.1.2.1 Kebutuhan Fungsional (Functional Requirements)

Analisis kebutuhan fungsional merupakan suatu kebutuhan yang berhubungan dengan kebutuhan sistem yang akan dibuat. Di mana menjabarkan mengenai fungsi-fungsi yang dapat mendukung jalannya sistem.

Setiap proses memiliki representasi masing-masing pada sebuah tabel atau data yang terdapat pada *database* yang telah dirancang sebelumnya dan setiap proses berhubungan langsung dengan entitas atau *user*.

5.1.2.2 Kebutuhan Non-Fungsional (Non-Functional Requirements)

Analisis kebutuhan non fungsional dilakukan untuk mengetahui spesifikasi kebutuhan untuk sistem. Spesifikasi kebutuhan melibatkan analisis perangkat keras, analisis perangkat lunak, dan analisis pengguna.

A. Kebutuhan Perangkat Keras Saat Pembuatan Aplikasi

Tabel 5. 1 Kebutuhan Perangkat Keras

No	Nama Perangkat	Spesifikasi	Keterangan
1.	<i>Hardisk</i>	100 GB	Media untuk menyimpan data aplikasi yang dibuat
2.	<i>RAM</i>	8 GB	Mempercepat pemrosesan data pada PC
5.	<i>Processor</i>	<i>Intel ® Core™ i5-6200U @2.30GHz</i>	Untuk kecepatan transfer data dari sistem yang sangat bergantung pada kecepatan prosesor komputer

B. Kebutuhan Perangkat Lunak Saat Pembuatan Aplikasi

Tabel 5. 2 Kebutuhan Perangkat Lunak

No	Tools / Software	Fungsi	Keterangan
1.	<i>Windows 10</i>	Sistem Operasi	Kompatibel di semua perangkat
2.	<i>XAMPP 5.6</i>	<i>Server</i> yang berdiri sendiri (<i>localhost</i>)	Dapat mengakses <i>web</i> dinamis tanpa koneksi internet dan <i>hosting</i>

4.	<i>Microsoft Office Visio 2016</i>	<i>Software Pendukung</i>	<i>Tools</i> nya sangat mudah dipahami, dan cara penggunaannya yang mudah
5.	<i>PHP</i>	<i>Bahasa Pemrograman yang digunakan</i>	Bersifat <i>open source</i> dan dapat membuat <i>web</i> menjadi dinamis
6.	<i>Sublime Text</i>	<i>Editor teks</i>	Memiliki banyak <i>fiture</i> , <i>cross-platform</i> , mudah, dan sederhana, dan mendukung banyak Bahasa pemrograman dan <i>markup</i>
7.	<i>CodeIgniter</i>	<i>Framework</i>	Memberikan standar <i>coding</i> sehingga memudahkan untuk mempelajari kembali sistem aplikasi yang dibangun
8.	<i>Google Chrome</i>	<i>Browser</i>	Memiliki proses <i>loading</i> pada tab yang terpisah
9.	<i>Star UML</i>	<i>Desain UML</i>	Mudah dalam menghubungkan antar koneksi pada suatu kelas atau aktivitas.

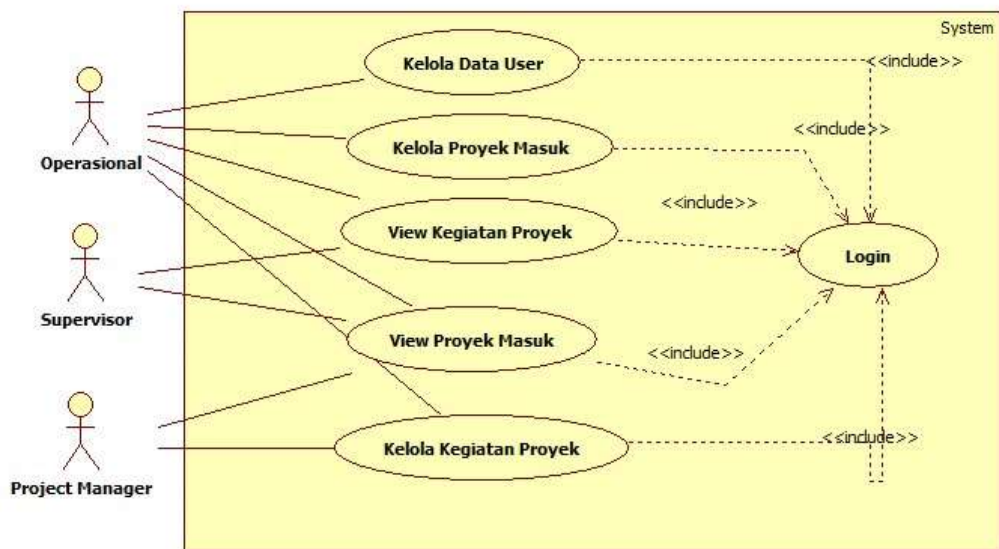
C. Analisis Pengguna

Aplikasi yang akan dibuat ini digunakan untuk lingkungan PT. Javan Cipta Solusi. Pengguna aplikasi yang terlibat adalah Operasional, Supervisor, dan *Project Manager*.

5.2 Perancangan

5.2.1 Use Case Diagram

Use Case Diagram merupakan diagram utama yang sangat penting untuk mengorganisasi dan memodelkan suatu sistem yang dibutuhkan serta diharapkan pengguna.



Gambar 5. 6 Use Case Diagram

5.2.1.1 Definisi Aktor

Tabel 5. 3 Definisi Aktor

No	Aktor	Keterangan
1.	Operasional	Operasional dapat melakukan akses ke semua menu, yaitu dapat melakukan pengelolaan data seperti kelola data user, kelola proyek masuk, kelola kegiatan proyek. Operasional juga dapat melihat kegiatan proyek dan proyek masuk.
2.	Supervisor	Supervisor dapat mengelola SDM, melihat proyek masuk, dan melihat kegiatan proyek.
3.	<i>Project Manager</i>	<i>Project Manager</i> dapat melihat proyek masuk dan mengelola kegiatan proyek.

5.2.1.2 Definisi Use Case

Use case yang ada dalam diagram didefinisikan pada tabel di bawah ini:

Tabel 5. 4 Definisi Use Case

No.	No. Definisi	Use Case	Deskripsi
1.	UC1	<i>Login Operasional</i>	Melakukan proses <i>login</i>
2.	UC2	<i>Login Supervisor</i>	Melakukan proses <i>login</i>
3.	UC3	<i>Login Project Manager</i>	Melakukan proses <i>login</i>
4.	UC4	<i>Kelola Data User</i>	Merupakan serangkaian aktivitas yang dilakukan untuk mengelola data <i>user</i> .
5.	UC5	Kelola Proyek Masuk	Merupakan serangkaian aktivitas yang dilakukan untuk mengelola data proyek masuk.
6.	UC6	View Kegiatan Proyek	Merupakan aktifitas yang dilakukan untuk melihat data kegiatan proyek.
7.	UC7	View Proyek Masuk	Merupakan aktifitas yang dilakukan untuk melihat data proyek yang masuk.
8.	UC8	Kelola Kegiatan Proyek	Merupakan serangkaian aktivitas yang dilakukan untuk mengelola data kegiatan proyek.

5.2.1.3 Skenario Use Case

Skenario *Use Case* pada Sistem Manajemen Proyek Menggunakan Critical Path Method Untuk Memprediksi Durasi Total Pengerjaan Proyek di PT. Javan Cipta Solusi di antaranya:

1. Skenario *Use Case Login Operasional*

Skenario dalam diagram *use case Login*, ada pada tabel seperti berikut ini:

Tabel 5. 5 Skenario Use Case Login Admin

Identifikasi	
Nomor	UC1
Nama	<i>Login Operasional</i>
Tujuan	Melakukan <i>Login</i> untuk masuk halaman utama.
Deskripsi	<i>Login</i> dengan memasukkan <i>Username</i> dan <i>Password</i>
Aktor	Operasional
Skenario Utama	
Kondisi Awal	Menampilkan <i>Form Login</i> .
Aksi Aktor	Reaksi Sistem
1. Memasukkan <i>username</i> dan <i>password</i> .	2. <i>Form Login</i> akan menampilkan <i>textbox username, password</i> dan untuk <i>password</i> ditampilkan dalam bentuk kode ● pada layar untuk jaminan keamanan.
3. Aktor melakukan konfirmasi persetujuan terhadap <i>username, password</i> yang telah dimasukkan dengan menekan tombol <i>Login</i> .	4. Aplikasi melakukan validasi terhadap <i>username, password</i> yang telah dimasukkan oleh pengguna dengan melakukan pengecekan pada basis data.
Kondisi Akhir	Jika pada akhir interaksi <i>username, password</i> yang dimasukkan pengguna <i>valid</i> maka pengguna akan langsung masuk ke halaman utama dan dapat menggunakan aplikasi sesuai hak aksesnya.
Alur Alternatif kondisi akhir	
a. Jika tidak <i>valid</i> maka Aplikasi akan kembali ke tampilan awal <i>login</i> .	

2. Skenario Use Case Login Supervisor

Skenario dalam diagram *use case Login* ada pada tabel seperti berikut ini:

Tabel 5. 6 Skenario Use Case Login Supervisor

Identifikasi

Nomor	UC2
Nama	<i>Login Supervisor</i>
Tujuan	Melakukan <i>Login</i> untuk masuk halaman utama.
Deskripsi	<i>Login</i> dengan memasukkan <i>Username</i> dan <i>Password</i>
Aktor	Supervisor
Skenario Utama	
Kondisi Awal	Menampilkan <i>Form Login</i> .
Aksi Aktor	Reaksi Sistem
1. Memasukkan <i>username</i> dan <i>password</i> .	2. <i>Form Login</i> akan menampilkan <i>textbox username, password</i> dan untuk <i>password</i> ditampilkan dalam bentuk kode ● pada layar untuk jaminan keamanan.
3. Aktor melakukan konfirmasi persetujuan terhadap <i>username, password</i> yang telah dimasukkan dengan menekan tombol <i>Login</i> .	4. Aplikasi melakukan validasi terhadap <i>username, password</i> yang telah dimasukkan oleh pengguna dengan melakukan pengecekan pada basis data.
Kondisi Akhir	Jika pada akhir interaksi <i>username, password</i> yang dimasukan pengguna <i>valid</i> maka pengguna akan langsung masuk ke halaman utama dan dapat menggunakan aplikasi sesuai hak aksesnya.
Alur Alternatif kondisi akhir	
a. Jika tidak <i>valid</i> maka Aplikasi akan kembali ke tampilan awal <i>login</i> .	

3. Skenario Use Case Login Project Manager

Skenario dalam diagram *use case Login* ada pada tabel seperti berikut ini:

Tabel 5. 7 Skenario Use Case Login Project Manager

Identifikasi	
Nomor	UC3

Nama	<i>Login Project Manager</i>
Tujuan	Melakukan <i>Login</i> untuk masuk halaman utama.
Deskripsi	<i>Login</i> dengan memasukan <i>Username</i> dan <i>Password</i>
Aktor	<i>Project Manager</i>
Skenario Utama	
Kondisi Awal	Menampilkan <i>Form Login</i> .
Aksi Aktor	Reaksi Sistem
1. Memasukkan <i>username</i> dan <i>password</i> .	2. <i>Form Login</i> akan menampilkan <i>textbox username, password</i> dan untuk <i>password</i> ditampilkan dalam bentuk kode '●' pada layar untuk jaminan keamanan.
3. Aktor melakukan konfirmasi persetujuan terhadap <i>username, password</i> yang telah dimasukkan dengan menekan tombol <i>Login</i> .	4. Aplikasi melakukan validasi terhadap <i>username, password</i> yang telah dimasukkan oleh pengguna dengan melakukan pengecekan pada basis data.
Kondisi Akhir	Jika pada akhir interaksi <i>username, password</i> yang dimasukan pengguna <i>valid</i> maka pengguna akan langsung masuk ke halaman utama dan dapat menggunakan aplikasi sesuai hak aksesnya.
Alur Alternatif kondisi akhir	
a. Jika tidak <i>valid</i> maka Aplikasi akan kembali ke tampilan awal <i>login</i> .	

4. Skenario *Use Case* Kelola Data User

Skenario dalam diagram *use case* Kelola Data User ada pada tabel seperti berikut ini:

Tabel 5. 8 Skenario Use Case Kelola Data User

Identifikasi	
Nomor	UC4
Nama	Kelola Data User

Tujuan	Melakukan manipulasi isi <i>record</i> data
Deskripsi	Melakukan CRUD terhadap data <i>user</i>
Aktor	Operasional
Skenario Utama	
Kondisi Awal	<i>Form login</i> terlebih dahulu. Jika <i>valid</i> maka masuk ke <i>form</i> halaman utama Operasional jika tidak <i>valid</i> akan kembali ke tampilan awal <i>login</i> .
Aksi Aktor	Reaksi Sistem
1. Operasional memilih <i>link</i> ke halaman yang akan dimanipulasi.	2. Aplikasi menampilkan <i>form</i> sesuai pilihan pengguna.
3. Operasional melakukan manipulasi isi <i>record</i> dengan memilih tombol tambah, edit atau hapus.	4. Aplikasi menampilkan pesan konfirmasi perintah untuk menyimpan data hasil penambahan, perubahan atau penghapusan data <i>user</i> .
5. Admin melakukan konfirmasi persetujuan manipulasi isi <i>record</i> .	6. Aplikasi menyimpan hasil manipulasi pada basis data tabel <i>tbl_user</i> atau yang terkait.
Kondisi Akhir	Isi informasi <i>record</i> pada basis data tabel <i>tbl_user</i> yang dilakukan manipulasi mengalami perubahan.
Alur alternatif no.4 a. Jika data yang dimasukkan tidak sesuai dengan tipe data yang bisa diterima basis data maka akan dimunculkan pesan <i>error</i> . b. Jika ada <i>textbox</i> yang wajib diisi masih kosong maka akan muncul pesan peringatan untuk melakukan pengecekan ulang	

5. Skenario Use Case Kelola Proyek Masuk

Skenario dalam diagram *use case* Kelola Status Masakan ada pada tabel seperti berikut ini:

Tabel 5. 9 Skenario Use Case Kelola Proyek Masuk

Identifikasi

Nomor	UC5
Nama	Kelola Data <i>User</i>
Tujuan	Melakukan manipulasi isi <i>record</i> data
Deskripsi	Melakukan CRUD terhadap data proyek masuk
Aktor	Operasional
Skenario Utama	
Kondisi Awal	<i>Form login</i> terlebih dahulu. Jika <i>valid</i> maka masuk ke <i>form</i> halaman utama Operasional jika tidak <i>valid</i> akan kembali ke tampilan awal <i>login</i> .
Aksi Aktor	Reaksi Sistem
1. Operasional memilih <i>link</i> ke halaman yang akan dimanipulasi.	2. Aplikasi menampilkan <i>form</i> sesuai pilihan pengguna.
3. Operasional melakukan manipulasi isi <i>record</i> dengan memilih tombol tambah, edit atau hapus.	4. Aplikasi menampilkan pesan konfirmasi perintah untuk menyimpan data hasil penambahan, perubahan atau penghapusan data <i>proyek masuk</i> .
5. Admin melakukan konfirmasi persetujuan manipulasi isi <i>record</i> .	6. Aplikasi menyimpan hasil manipulasi pada basis data tabel <i>tbl_proyek</i> atau yang terkait.
Kondisi Akhir	Isi informasi <i>record</i> pada basis data tabel <i>tbl_proyek</i> yang dilakukan manipulasi mengalami perubahan.
Alur alternatif no.4 a. Jika data yang dimasukkan tidak sesuai dengan tipe data yang bisa diterima basis data maka akan dimunculkan pesan <i>error</i> . b. Jika ada <i>textbox</i> yang wajib diisi masih kosong maka akan muncul pesan peringatan untuk melakukan pengecekan ulang	

6. Skenario Use Case View Kegiatan Proyek

Skenario dalam diagram *use case* View Kegiatan Proyek ada pada tabel seperti berikut ini:

Tabel 5. 10 Skenario Use Case View Kegiatan Proyek

Identifikasi	
Nomor	UC6
Nama	View Kegiatan Proyek
Tujuan	Melihat isi <i>record</i> data
Deskripsi	<i>View record</i> data yang telah diinputkan
Aktor	Operasional, Supervisor
Skenario Utama	
Kondisi Awal	<i>Form login</i> terlebih dahulu. Jika <i>valid</i> maka masuk ke <i>form</i> halaman utama untuk kasir jika tidak <i>valid</i> akan kembali ke tampilan awal <i>login</i> .
Aksi Aktor	Reaksi Sistem
1. Staf memilih <i>link</i> ke halaman yang akan dilihat.	2. Aplikasi menampilkan data sesuai pilihan pengguna.
Kondisi Akhir	Isi informasi <i>record</i> pada basis data tabel <i>tbl_kegiatan</i> ditampilkan.

7. Skenario Use Case View Proyek Masuk

Skenario dalam diagram *use case* View Proyek Masuk ada pada tabel seperti berikut ini:

Tabel 5. 11 Skenario Use Case View Proyek Masuk

Identifikasi	
Nomor	UC7
Nama	View Proyek Masuk
Tujuan	Melihat isi <i>record</i> data
Deskripsi	<i>View record</i> data yang telah diinputkan
Aktor	Operasional, Supervisor, <i>Project Manager</i>
Skenario Utama	

Kondisi Awal	<i>Form login</i> terlebih dahulu. Jika <i>valid</i> maka masuk ke <i>form</i> halaman utama untuk kasir jika tidak <i>valid</i> akan kembali ke tampilan awal <i>login</i> .
Aksi Aktor	Reaksi Sistem
1. Staf memilih <i>link</i> ke halaman yang akan dilihat.	2. Aplikasi menampilkan data sesuai pilihan pengguna.
Kondisi Akhir	Isi informasi <i>record</i> pada basis data tabel <i>tbl_proyek</i> ditampilkan.

8. Skenario Use Case Kelola Kegiatan Proyek

Skenario dalam diagram *use case* Kelola Kegiatan Proyek ada pada tabel seperti berikut ini:

Tabel 5. 12 Skenario Use Case Kelola Kegiatan Proyek

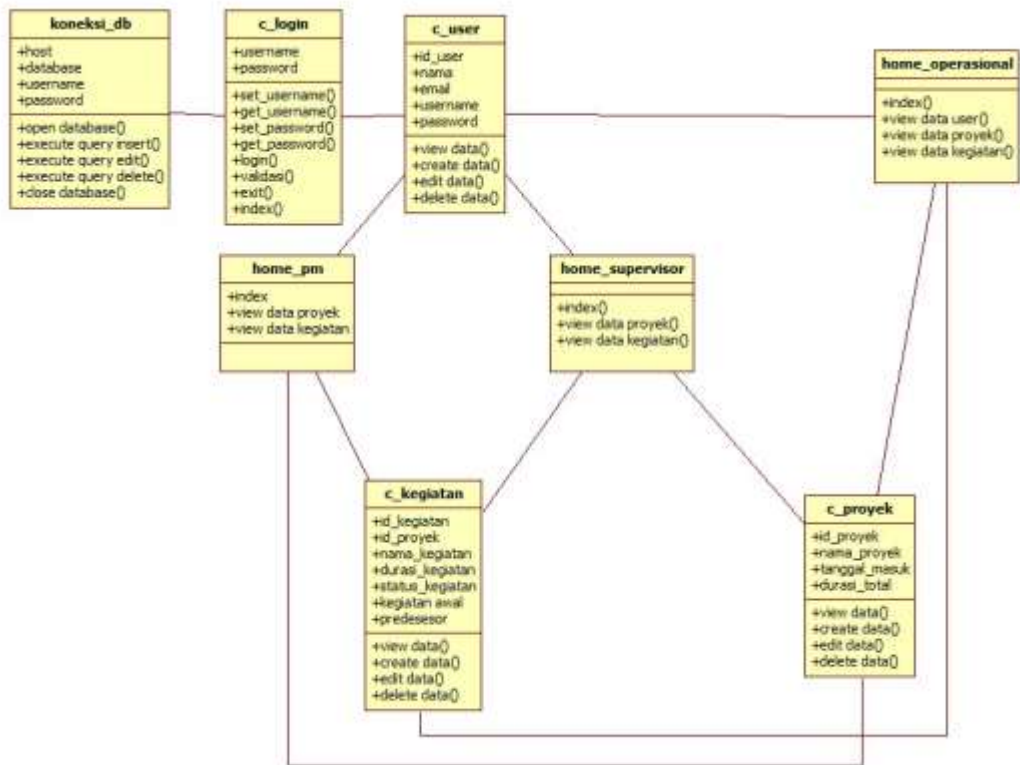
Identifikasi	
Nomor	UC8
Nama	Kelola Kegiatan Proyek
Tujuan	Melakukan manipulasi isi <i>record</i> data
Deskripsi	Melakukan CRUD terhadap data kegiatan proyek
Aktor	Operasional, <i>Project Manager</i>
Skenario Utama	
Kondisi Awal	<i>Form login</i> terlebih dahulu. Jika <i>valid</i> maka masuk ke <i>form</i> halaman utama Operasional jika tidak <i>valid</i> akan kembali ke tampilan awal <i>login</i> .
Aksi Aktor	Reaksi Sistem
1. Operasional memilih <i>link</i> ke halaman yang akan dimanipulasi.	2. Aplikasi menampilkan <i>form</i> sesuai pilihan pengguna.
3. Operasional melakukan manipulasi isi <i>record</i> dengan memilih tombol tambah, edit atau hapus.	4. Aplikasi menampilkan pesan konfirmasi perintah untuk menyimpan data hasil penambahan, perubahan atau penghapusan data <i>kegiatan proyek</i> .

5. Admin melakukan konfirmasi persetujuan manipulasi isi <i>record</i> .	6. Aplikasi menyimpan hasil manipulasi pada basis data tabel <i>tbl_kegiatan</i> atau yang terkait.
Kondisi Akhir	Isi informasi <i>record</i> pada basis data tabel <i>tbl_kegiatan</i> yang dilakukan manipulasi mengalami perubahan.
Alur alternatif no.4 a. Jika data yang dimasukkan tidak sesuai dengan tipe data yang bisa diterima basis data maka akan dimunculkan pesan <i>error</i> . b. Jika ada <i>textbox</i> yang wajib diisi masih kosong maka akan muncul pesan peringatan untuk melakukan pengecekan ulang	

5.2.2 Class Diagram

Class diagram adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek.

Dalam proses pembuatan aplikasi ini *class diagram* yang akan digunakan adalah sebagai berikut:

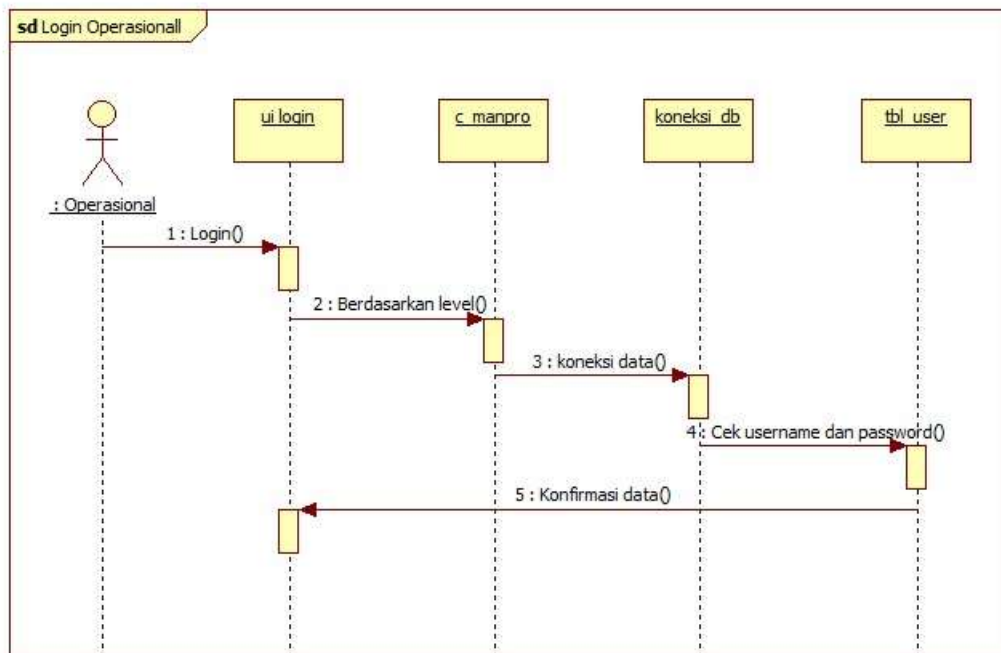


Gambar 5. 7 Class Diagram

5.2.3 Sequence Diagram

Sequence diagram adalah suatu diagram yang menggambarkan interaksi antar objek dan mengindikasikan komunikasi diantara objek-objek di dalam dan disekitar sistem.

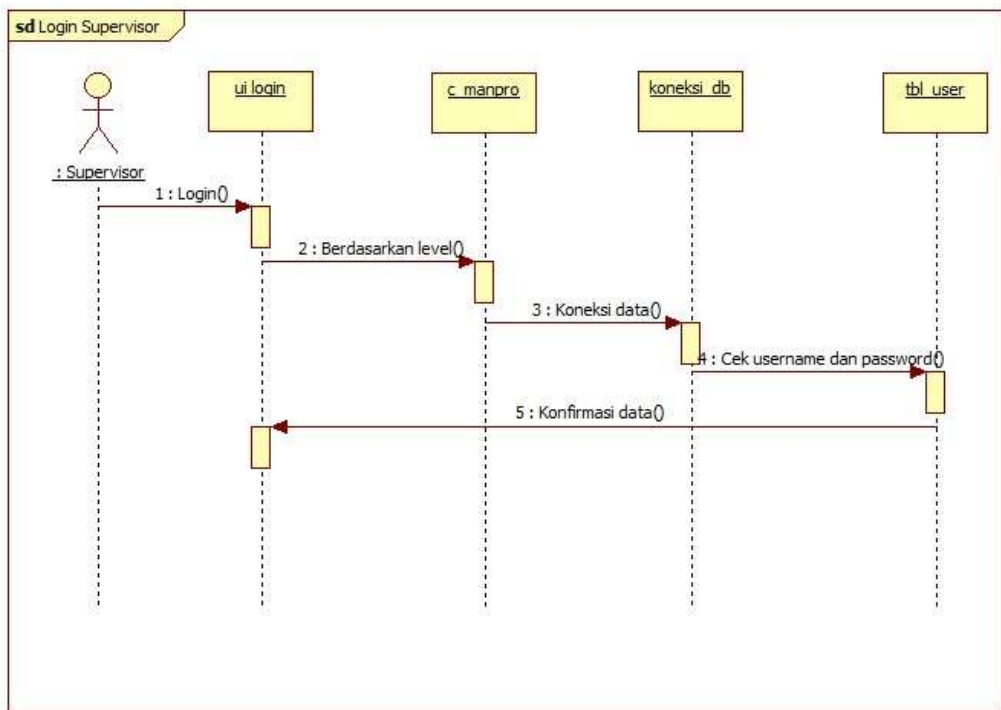
5.2.3.1 Sequence Diagram Login Operasional (UC1)



Gambar 5. 8 Sequence Diagram Login Operasional (UC1)

Pada *sequence diagram* ini menjelaskan proses *login*. Aktor mulai menjalankan aplikasi dan akan tampil *form login*. Kemudian aktor memasukkan *username dan password*, pada *interface login* data yang di *input* akan di cek di tabel `tbl_user` di *database*. Jika sesuai maka akan mendapatkan tampilan form utama ke pengguna.

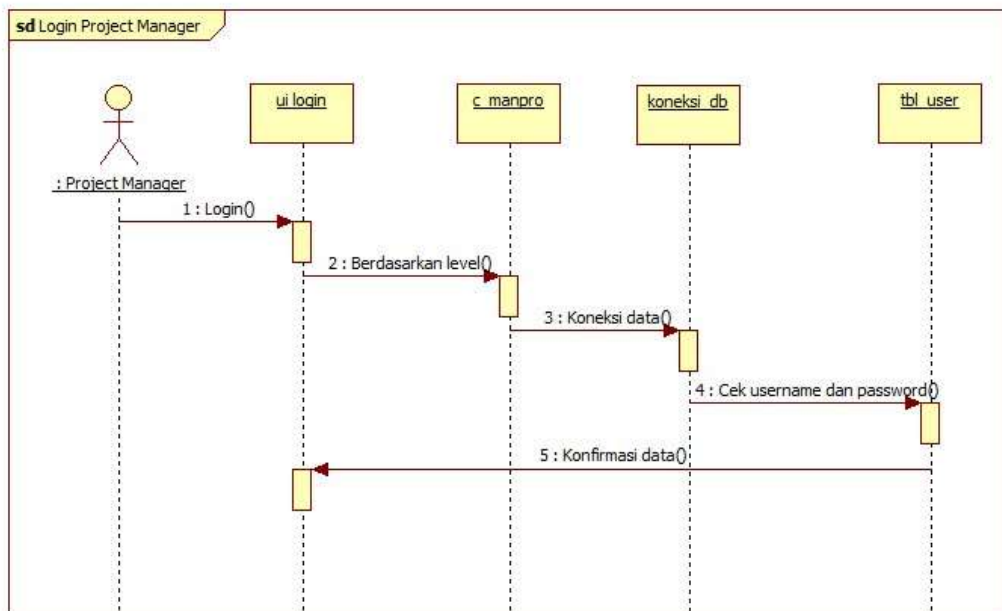
5.2.3.2 Sequence Diagram Login Supervisor (UC2)



Gambar 5. 9 Sequence Diagram Login Supervisor(UC2)

Pada *sequence diagram* ini menjelaskan proses *login*. Aktor mulai menjalankan aplikasi dan akan tampil *form login*. Kemudian aktor memasukkan *username dan password*, pada *interface login* data yang di *input* akan di cek di tabel *tbl_user* di *database*. Jika sesuai maka akan mendapatkan tampilan form utama ke pengguna.

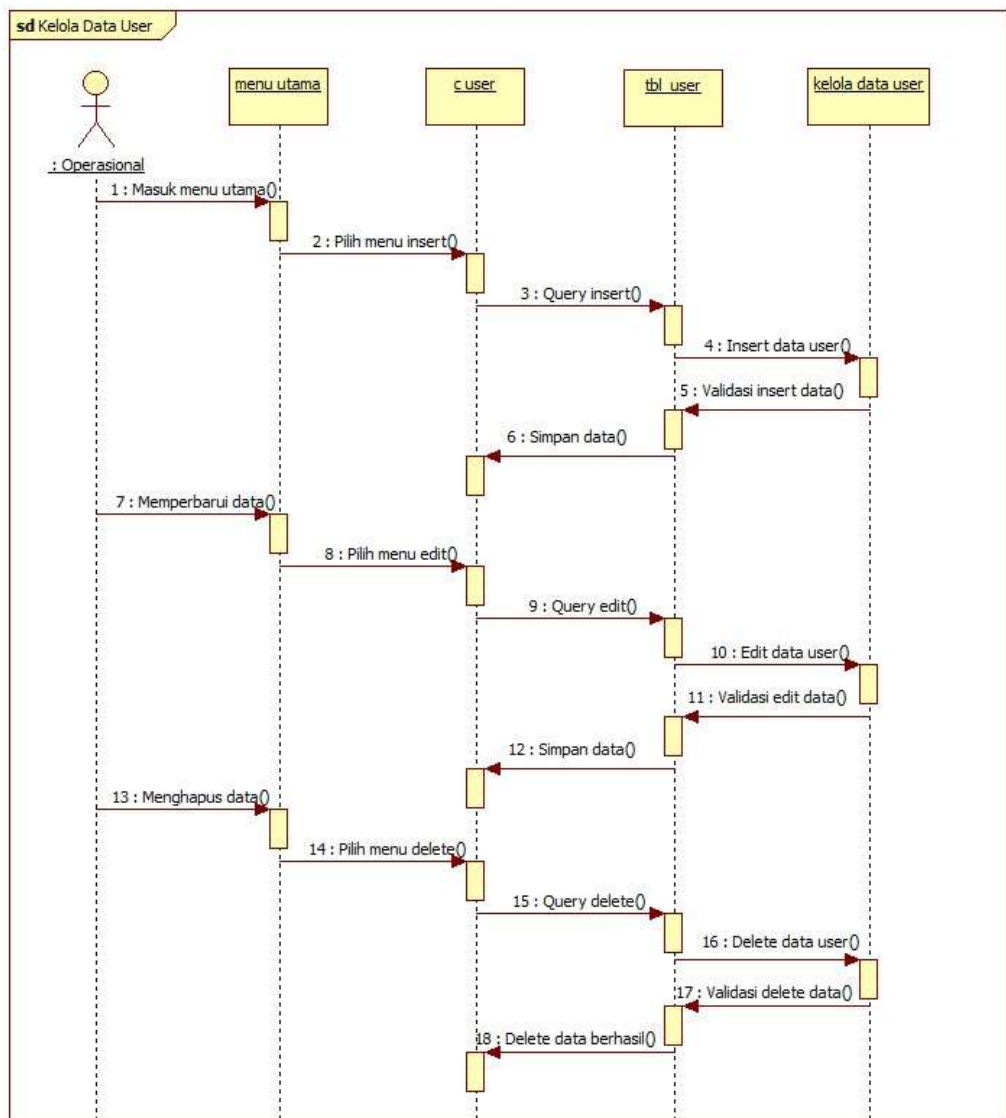
5.2.3.3 Sequence Diagram Login Project Manager (UC3)



Gambar 5. 10 Sequence Diagram Login Project Manager (UC3)

Pada *sequence diagram* ini menjelaskan proses *login*. Aktor mulai menjalankan aplikasi dan akan tampil *form login*. Kemudian aktor memasukkan *username dan password*, pada *interface login* data yang di input akan di cek di tabel *tbl_user* di *database*. Jika sesuai maka akan mendapatkan tampilan form utama ke pengguna.

5.2.3.4 Sequence Diagram Kelola Data User (UC4)

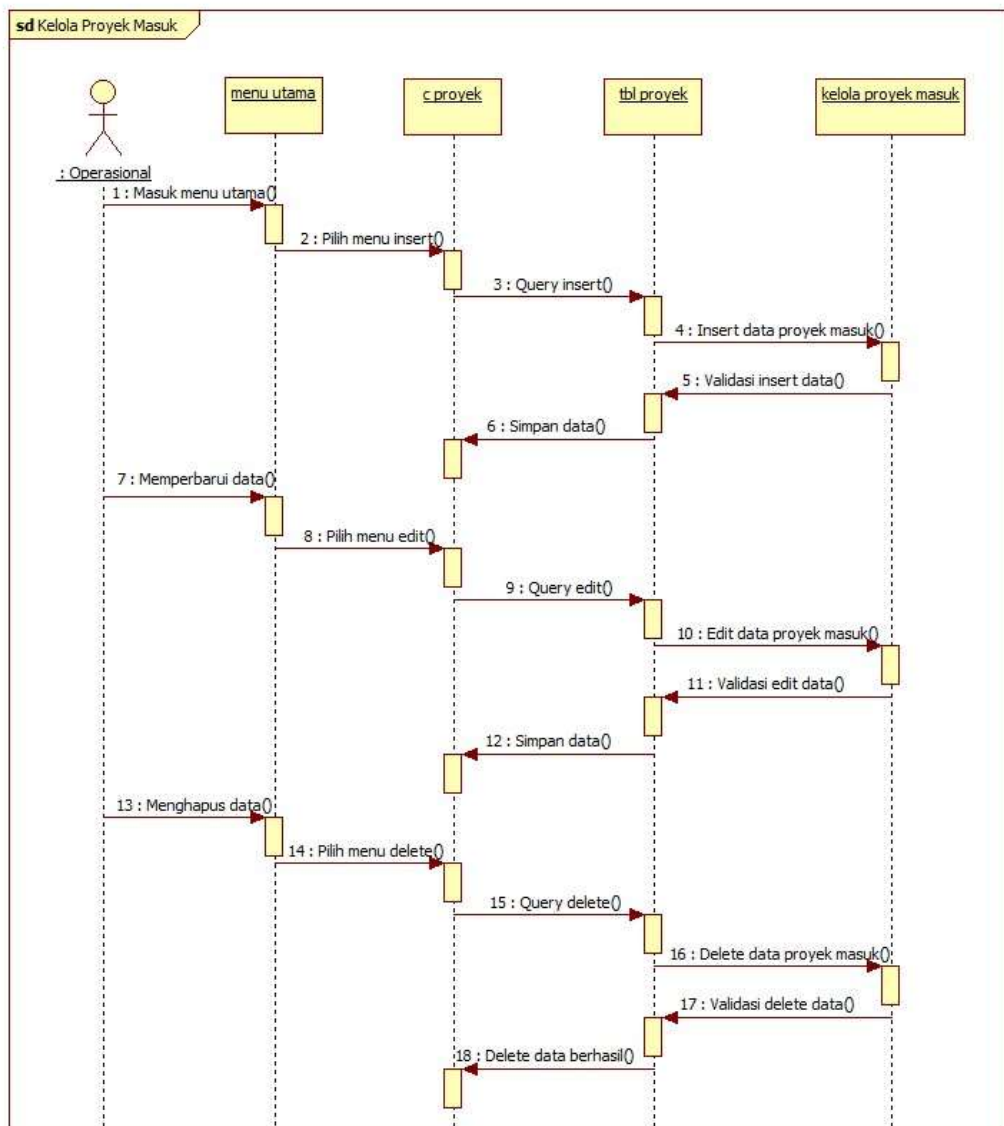


Gambar 5. 11 Sequence Diagram Kelola Data User (UC4)

Pada *sequence diagram* ini menjelaskan proses kelola data *user*. Aktor melakukan *login* terlebih dahulu. Selanjutnya sistem akan memuat *form* menu utama. Untuk memulai kelola ini, masuk ke menu kelola data *user*,

kemudian sistem akan melakukan proses koneksi database dan dilanjutkan dengan proses *OpenTable* ke tabel yang dituju yaitu *tbl_user* dengan melakukan operasi *insert()*, *update()*, *delete()*. Selanjutnya sistem akan memvalidasi hasil dari operasi, apakah berhasil atau gagal. Dan yang terakhir sistem akan melakukan proses *CloseDB* untuk menutup koneksi.

5.2.3.5 Sequence Diagram Kelola Proyek Masuk (UC5)

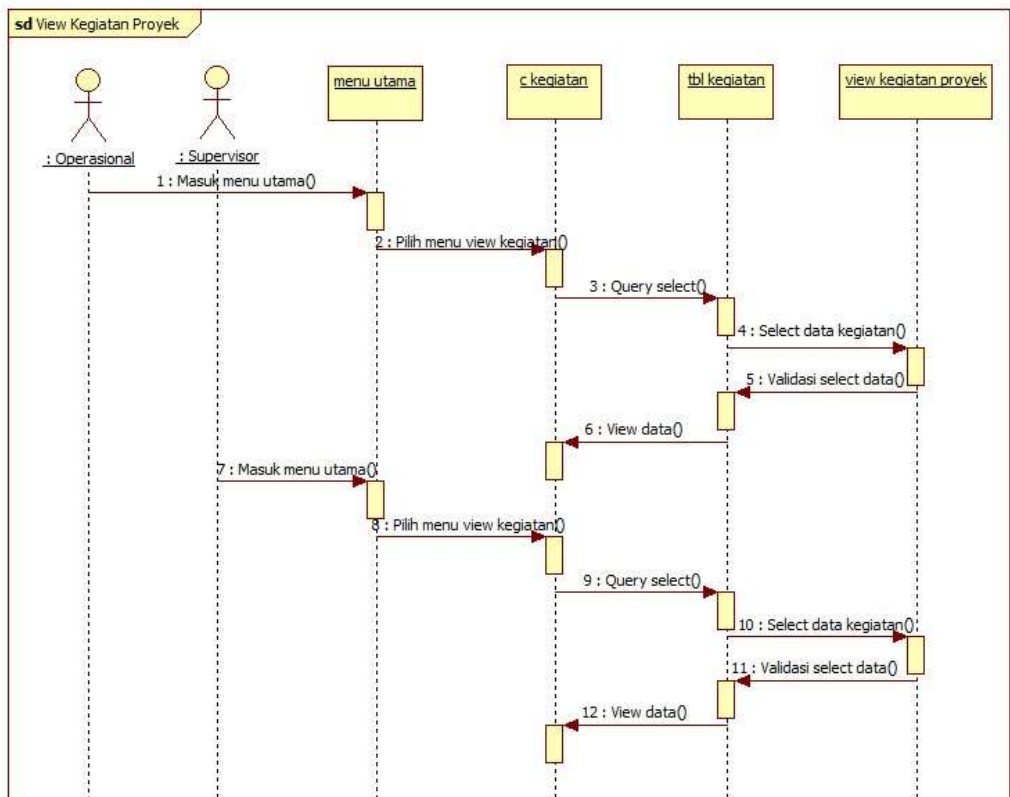


Gambar 5. 12 Sequence Diagram Kelola Proyek Masuk (UC5)

Pada *sequence diagram* ini menjelaskan proses kelola proyek masuk. Aktor melakukan *login* terlebih dahulu. Selanjutnya sistem akan memuat *form* menu utama. Untuk memulai kelola ini, masuk ke menu proyek masuk

kemudian sistem akan melakukan proses koneksi *database* dan dilanjutkan dengan proses *OpenTable* ke tabel yang dituju yaitu *tbl_proyek* dengan melakukan operasi *insert()*, *update()*, *delete()*. Selanjutnya sistem akan memvalidasi hasil dari operasi, apakah berhasil atau gagal. Dan yang terakhir sistem akan melakukan proses *CloseDB* untuk menutup koneksi.

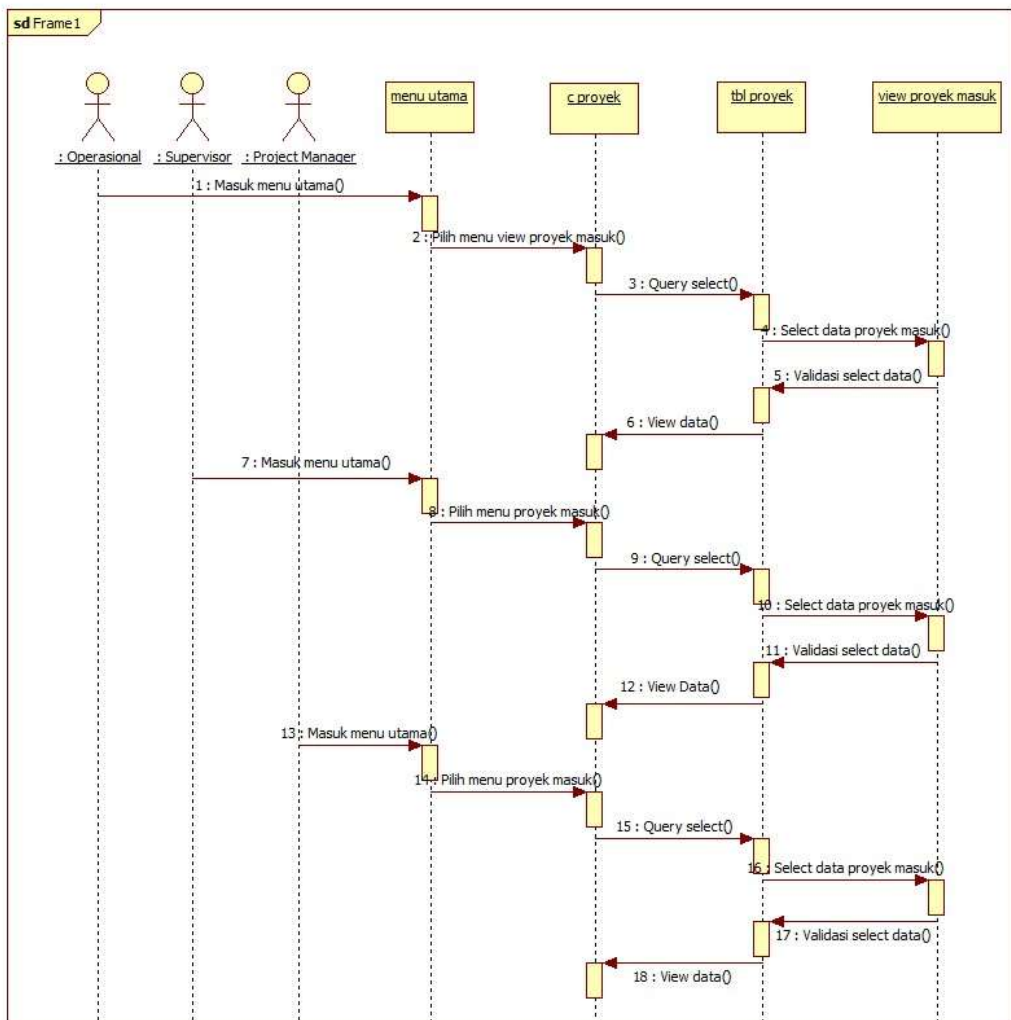
5.2.3.6 Sequence Diagram View Kegiatan Proyek (UC6)



Gambar 5. 13 Sequence Diagram View Kegiatan Proyek (UC6)

Pada *sequence diagram* ini menjelaskan proses *view data* kegiatan proyek. Aktor melakukan *login* terlebih dahulu. Selanjutnya sistem akan memuat *form* menu utama. Untuk memulai proses ini, masuk ke menu data kegiatan, kemudian sistem akan melakukan proses koneksi *database* dan dilanjutkan dengan proses *OpenTable* ke tabel yang dituju yaitu *tbl_kegiatan* dengan melakukan operasi *select()*. Selanjutnya sistem akan memvalidasi hasil dari operasi, apakah berhasil atau gagal. Dan yang terakhir sistem akan melakukan proses *CloseDB* untuk menutup koneksi.

5.2.3.7 Sequence Diagram View Proyek Masuk (UC7)

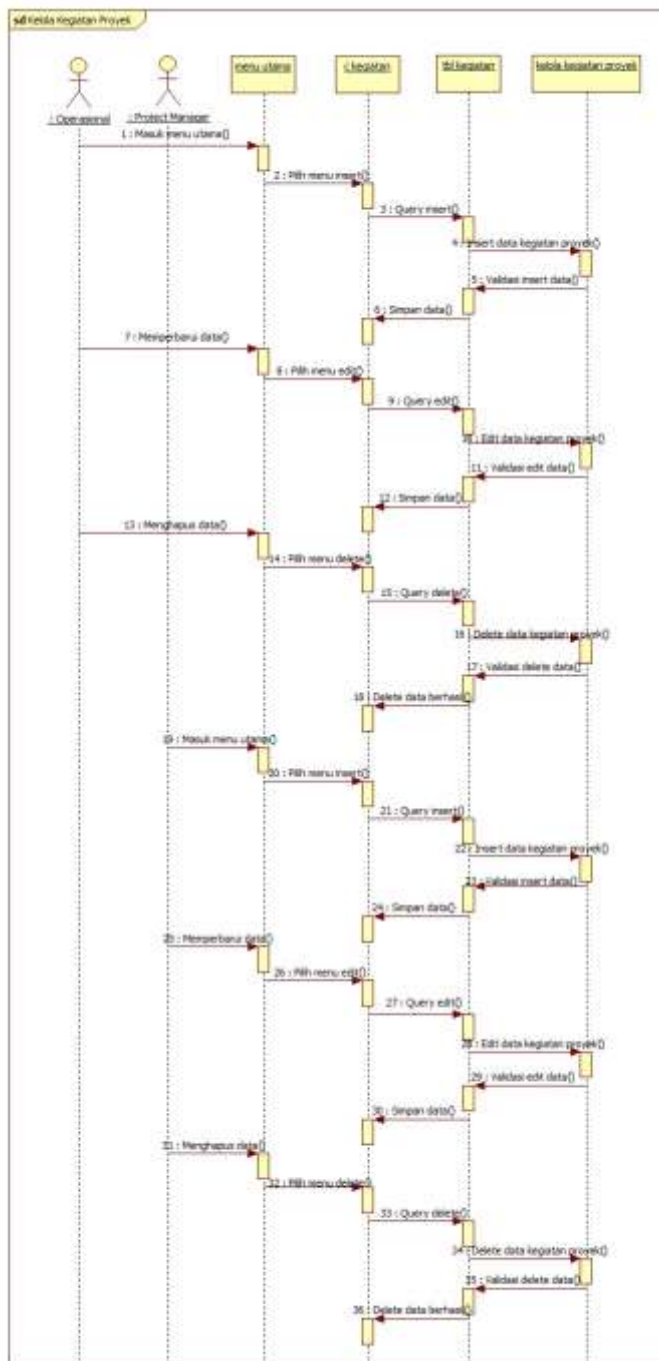


Gambar 5. 14 Sequence Diagram View Proyek Masuk (UC7)

Pada *sequence diagram* ini menjelaskan proses *view data* proyek masuk. Aktor melakukan *login* terlebih dahulu. Selanjutnya sistem akan memuat *form* menu utama. Untuk memulai proses ini, masuk ke menu data proyek, kemudian sistem akan melakukan proses koneksi *database* dan dilanjutkan dengan proses *OpenTable* ke tabel yang dituju yaitu *tbl_proyek*

dengan melakukan operasi *select()*. Selanjutnya sistem akan memvalidasi hasil dari operasi, apakah berhasil atau gagal. Dan yang terakhir sistem akan melakukan proses *CloseDB* untuk menutup koneksi.

5.2.3.8 Sequence Diagram Kelola Kegiatan Proyek (UC8)



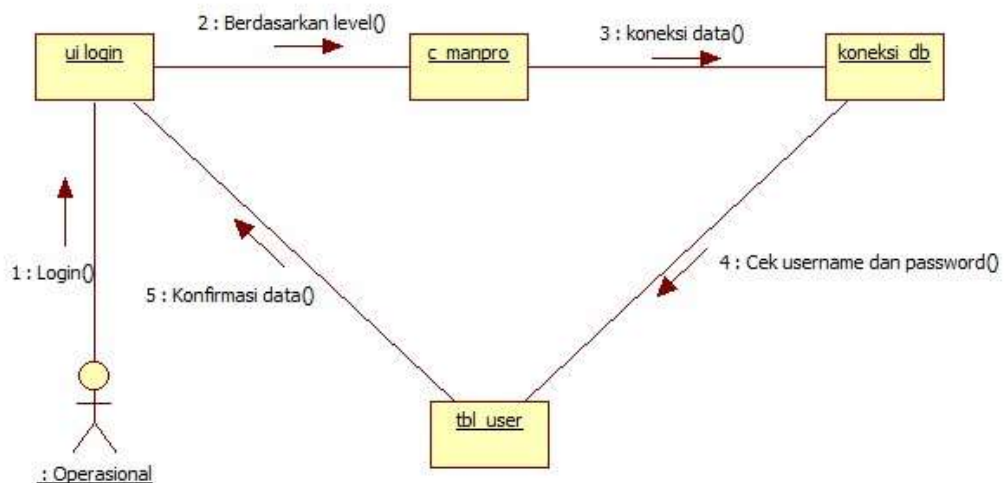
Gambar 5. 15 Sequence Diagram Kelola Kegiatan Proyek (UC8)

Pada *sequence diagram* ini menjelaskan proses kelola kegiatan proyek. Aktor melakukan *login* terlebih dahulu. Selanjutnya sistem akan memuat *form* menu utama. Untuk memulai kelola ini, masuk ke menu kelola kegiatan proyek, kemudian sistem akan melakukan proses koneksi *database* dan dilanjutkan dengan proses *OpenTable* ke tabel yang dituju yaitu *tbl_kegiatan* dengan melakukan operasi *update()*. Selanjutnya sistem akan memvalidasi hasil dari operasi, apakah berhasil atau gagal. Dan yang terakhir sistem akan melakukan proses *CloseDB* untuk menutup koneksi.

5.2.4 Communication Diagram

Communication diagram adalah sebuah interaksi yang memberi tekanan pada hubungan-hubungan data antar partisipan yang berbeda dalam sebuah interaksi.

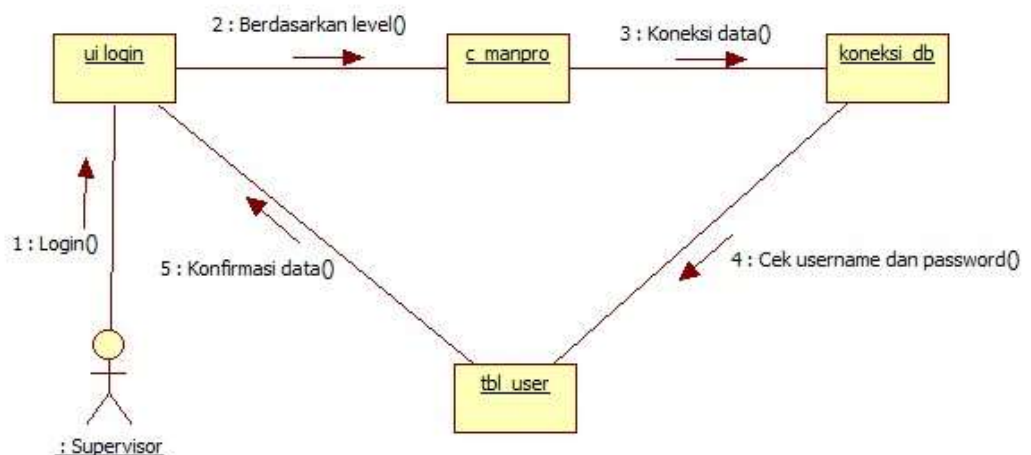
5.2.3.1 Communication Diagram Login Operasional (UC1)



Gambar 5. 16 Communication Diagram Login Operasional (UC1)

Pada *Communication diagram* ini menjelaskan proses *login*. Aktor mulai menjalankan aplikasi dan akan tampil *form login*. Kemudian aktor memasukkan *username dan password*, pada *interface login* data yang di *input* akan di cek di tabel *tbl_user* di *database*. Jika sesuai maka akan mendapatkan tampilan form utama ke pengguna.

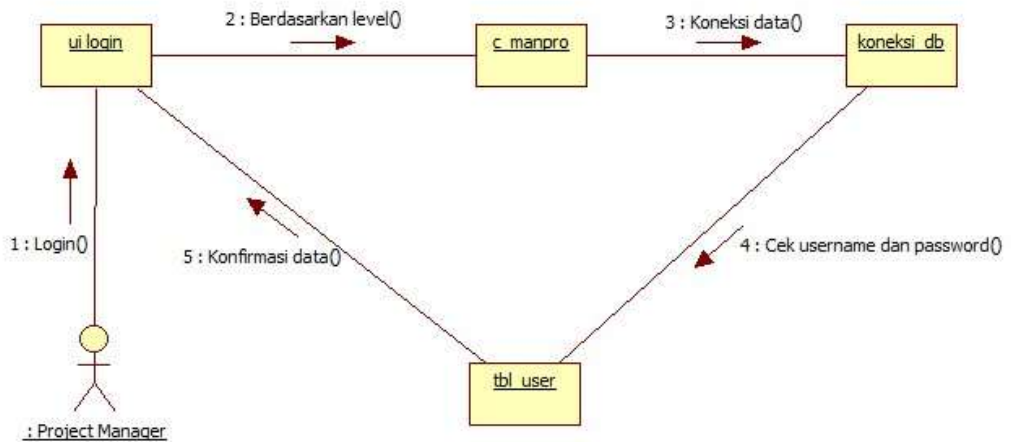
5.2.3.2 Communication Diagram Login Supervisor (UC2)



Gambar 5. 17 Communication Diagram Login Supervisor (UC2)

Pada *Communication diagram* ini menjelaskan proses *login*. Aktor mulai menjalankan aplikasi dan akan tampil *form login*. Kemudian aktor memasukkan *username dan password*, pada *interface login* data yang di *input* akan di cek di tabel *tbl_user* di *database*. Jika sesuai maka akan mendapatkan tampilan form utama ke pengguna.

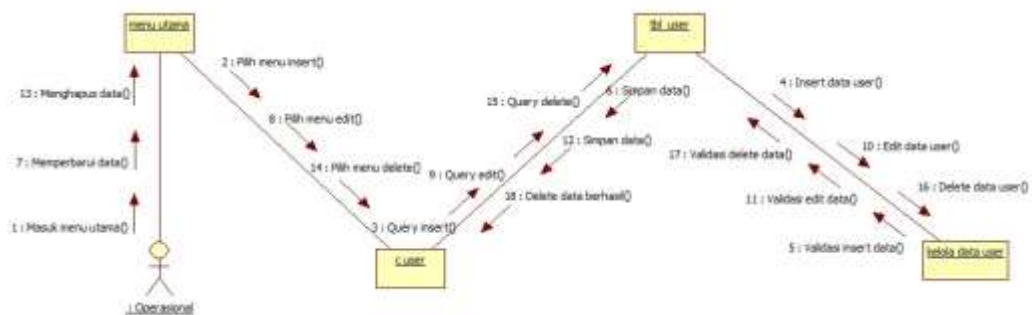
5.2.3.3 Communication Diagram Login Project Manager (UC3)



Gambar 5. 18 Communication Diagram Login Project Manager (UC3)

Pada *Communication diagram* ini menjelaskan proses *login*. Aktor mulai menjalankan aplikasi dan akan tampil *form login*. Kemudian aktor memasukkan *username dan password*, pada *interface login* data yang di input akan di cek di tabel `tbl_user` di *database*. Jika sesuai maka akan mendapatkan tampilan form utama ke pengguna.

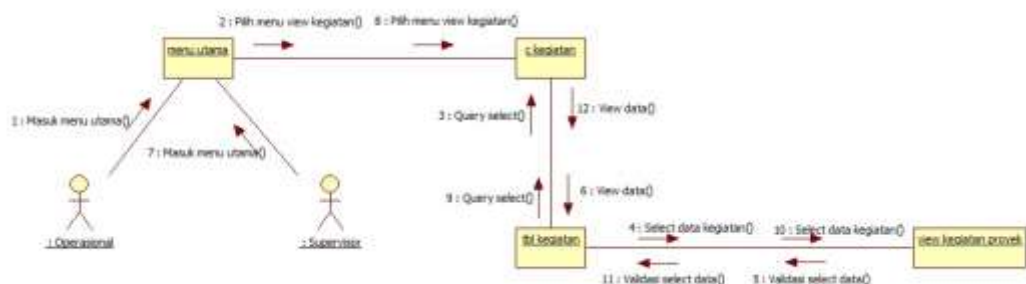
5.2.3.4 Communication Diagram Kelola Data User (UC4)



Gambar 5. 19 Communication Diagram Kelola Data User (UC4)

Pada *Communication diagram* ini menjelaskan proses kelola data *user*. Aktor melakukan *login* terlebih dahulu. Selanjutnya sistem akan memuat *form* menu utama. Untuk memulai kelola ini, masuk ke menu kelola data *user*, kemudian sistem akan melakukan proses koneksi database dan dilanjutkan dengan proses *OpenTable* ke tabel yang dituju yaitu *tbl_user* dengan melakukan operasi *insert()*, *update()*, *delete()*. Selanjutnya sistem akan memvalidasi hasil dari operasi, apakah berhasil atau gagal. Dan yang terakhir sistem akan melakukan proses *CloseDB* untuk menutup koneksi.

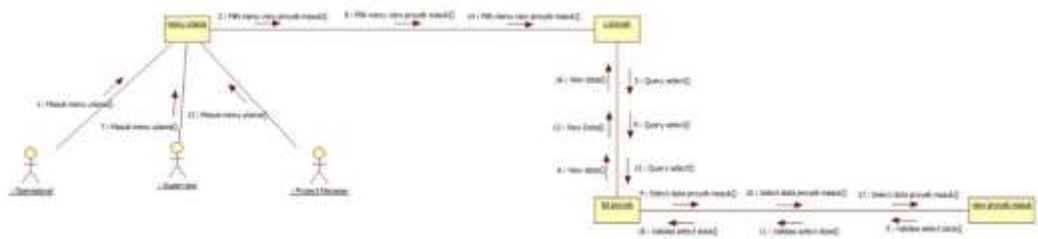
5.2.3.5 Communication Diagram Kelola Proyek Masuk (UC5)



Gambar 5. 21 Communication Diagram View Kegiatan Proyek (UC6)

Pada *Communication diagram* ini menjelaskan proses *view data* kegiatan proyek. Aktor melakukan *login* terlebih dahulu. Selanjutnya sistem akan memuat *form* menu utama. Untuk memulai proses ini, masuk ke menu data kegiatan, kemudian sistem akan melakukan proses koneksi *database* dan dilanjutkan dengan proses *OpenTable* ke tabel yang dituju yaitu *tbl_kegiatan* dengan melakukan operasi *select()*. Selanjutnya sistem akan memvalidasi hasil dari operasi, apakah berhasil atau gagal. Dan yang terakhir sistem akan melakukan proses *CloseDB* untuk menutup koneksi.

5.2.3.7 Communication Diagram View Proyek Masuk (UC7)

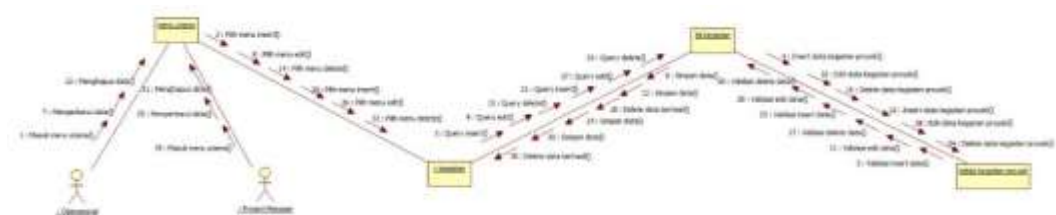


Gambar 5. 22 Communication Diagram View Proyek Masuk (UC7)

Pada *Communication diagram* ini menjelaskan proses *view data* proyek masuk. Aktor melakukan *login* terlebih dahulu. Selanjutnya sistem akan memuat *form* menu utama. Untuk memulai proses ini, masuk ke menu data proyek, kemudian sistem akan melakukan proses koneksi *database* dan dilanjutkan dengan proses *OpenTable* ke tabel yang dituju yaitu tbl proyek

dengan melakukan operasi *select()*. Selanjutnya sistem akan memvalidasi hasil dari operasi, apakah berhasil atau gagal. Dan yang terakhir sistem akan melakukan proses *CloseDB* untuk menutup koneksi.

5.2.3.8 Communication Diagram Kelola Kegiatan Proyek (UC8)



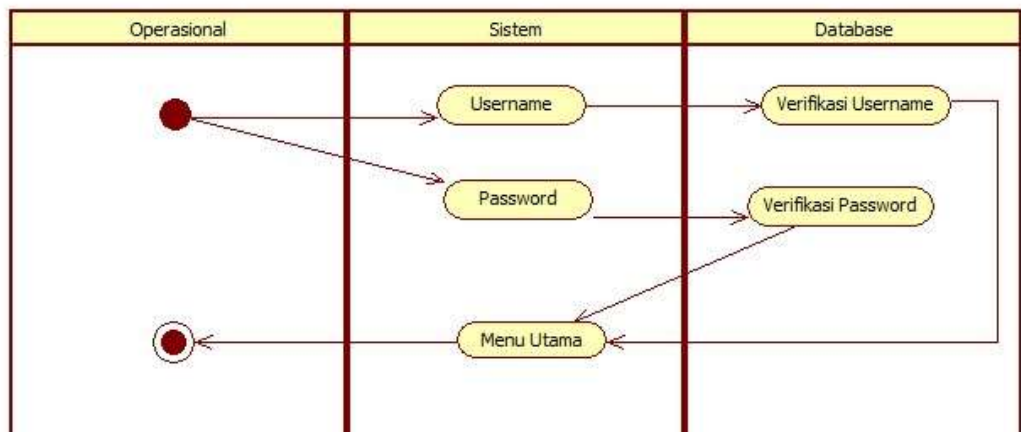
Gambar 5. 23 Communication Diagram Kelola Kegiatan Proyek (UC8)

Pada *Communication diagram* ini menjelaskan proses kelola kegiatan proyek. Aktor melakukan *login* terlebih dahulu. Selanjutnya sistem akan memuat *form* menu utama. Untuk memulai kelola ini, masuk ke menu kelola kegiatan proyek, kemudian sistem akan melakukan proses koneksi *database* dan dilanjutkan dengan proses *OpenTable* ke tabel yang dituju yaitu *tbl_kegiatan* dengan melakukan operasi *update()*. Selanjutnya sistem akan memvalidasi hasil dari operasi, apakah berhasil atau gagal. Dan yang terakhir sistem akan melakukan proses *CloseDB* untuk menutup koneksi.

5.2.5 Activity Diagram

Activity Diagram menggambarkan berbagai alir aktivitas alam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir [33].

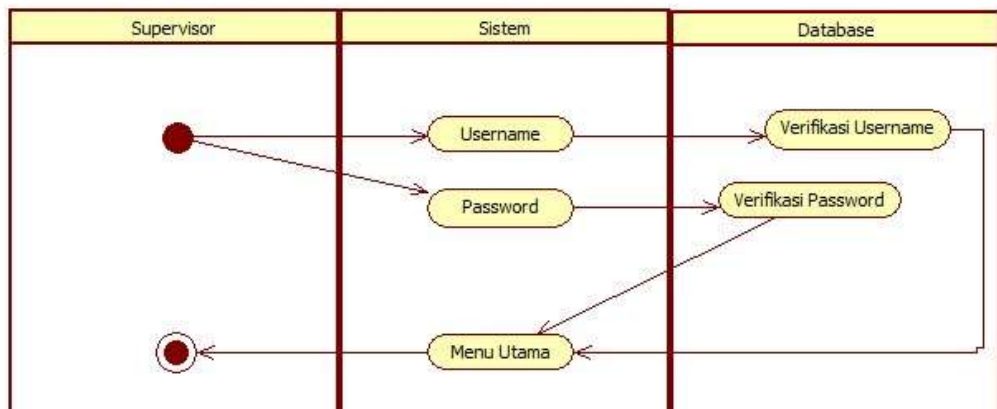
5.2.5.1 Activity Diagram Login Operasional (UC1)



Gambar 5. 26 Activity Diagram Login Operasional (UC1)

Pada *activity diagram* ini menjelaskan proses *login* Operasional. Aktor mulai dengan memasukkan *username* dan *password*, kemudian sistem akan memvalidasi apakah *username* dan *password* yang dimasukkan *valid* atau tidak *valid*. Jika *valid* maka akan tampil menu utama, tetapi jika tidak *valid* akan kembali ke *form login*.

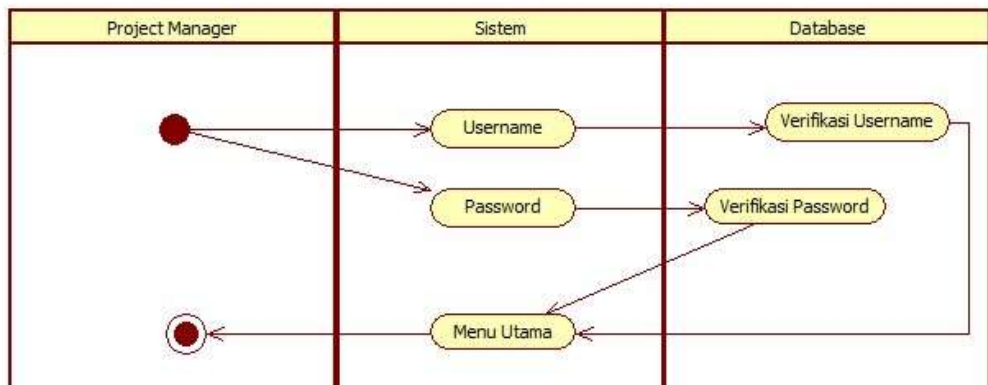
5.2.5.2 Activity Diagram Login Supervisor (UC2)



Gambar 5. 27 Activity Diagram Login Supervisor (UC2)

Pada *activity diagram* ini menjelaskan proses *login* Supervisor. Aktor mulai dengan memasukkan *username* dan *password*, kemudian sistem akan memvalidasi apakah *username* dan *password* yang dimasukkan *valid* atau tidak *valid*. Jika *valid* maka akan tampil menu utama, tetapi jika tidak *valid* akan kembali ke *form login*.

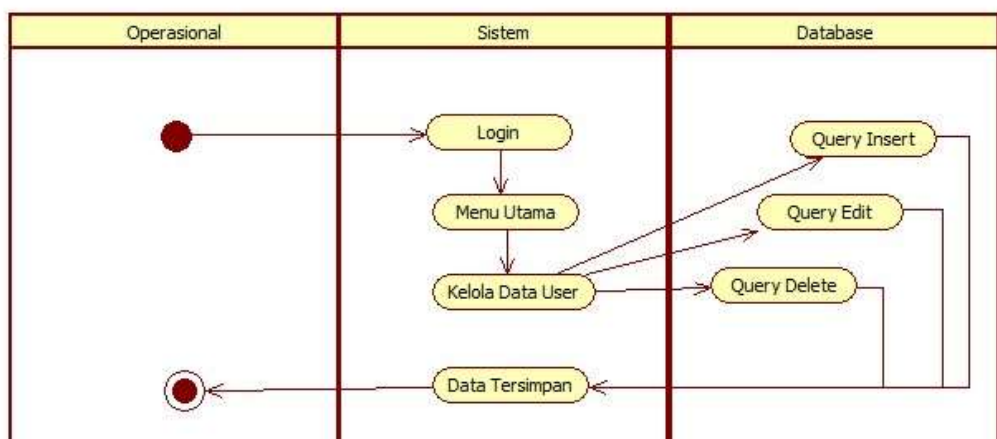
5.2.5.3 Activity Diagram Login Project Manager (UC3)



Gambar 5. 28 Activity Diagram Project Manager (UC3)

Pada *activity diagram* ini menjelaskan proses *login Project Manager*. Aktor mulai dengan memasukkan *username* dan *password*, kemudian sistem akan memvalidasi apakah *username* dan *password* yang dimasukkan *valid* atau tidak *valid*. Jika *valid* maka akan tampil menu utama, tetapi jika tidak *valid* akan kembali ke *form login*.

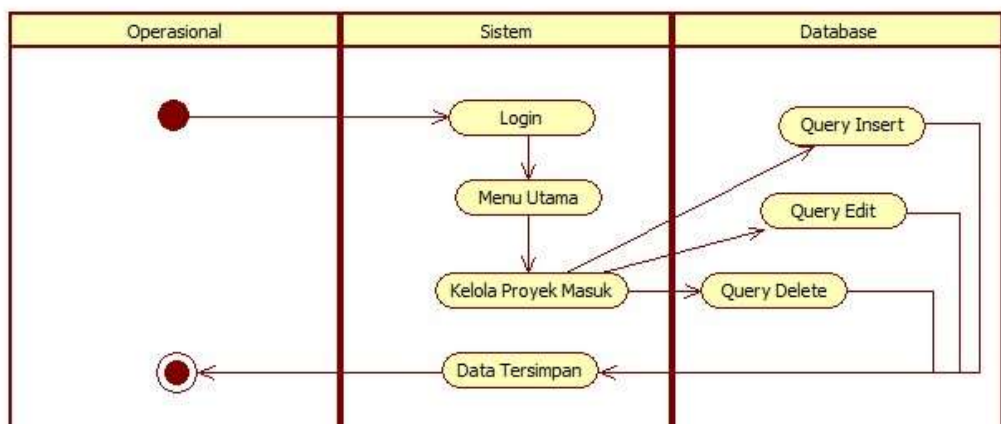
5.2.5.4 Activity Diagram Kelola Data User (UC4)



Gambar 5. 29 1 Activity Diagram Kelola Data User (UC4)

Setelah *login*, Admin memilih menu kelola data *user*. Kemudian sistem akan menampilkan kelola data *user*. Aktor akan melakukan proses *insert*, *update*, dan *delete* data meja, selanjutnya di sistem akan disimpan data *user*. Jika *valid* maka akan tampil data *user*, tetapi jika tidak *valid* akan kembali ke kelola data *user*. Admin dapat *logout* dan sistem menutup *database*.

5.2.5.5 Activity Diagram Kelola Proyek Masuk (UC5)

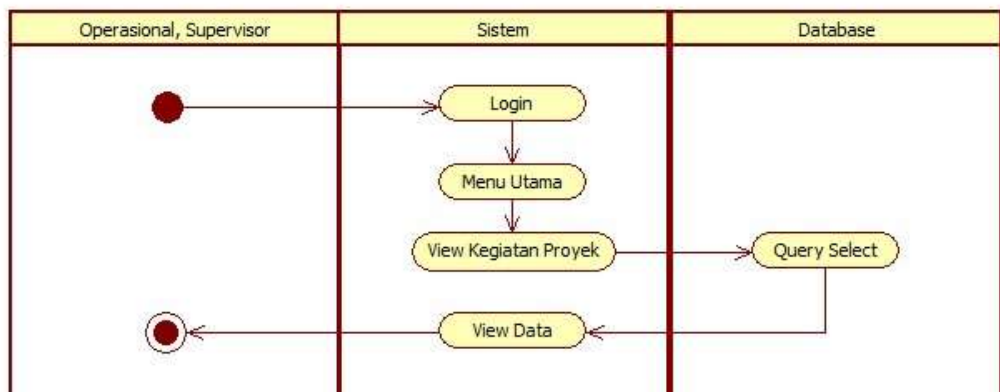


Gambar 5. 30 Activity Diagram Kelola Data Proyek Masuk (UC5)

Setelah *login*, Admin memilih menu kelola Proyek Masuk. Kemudian sistem akan menampilkan kelola data Proyek Masuk. Aktor akan melakukan proses *insert*, *update*, dan *delete* data Proyek Masuk, selanjutnya di sistem

akan disimpan data Proyek Masuk. Jika *valid* maka akan tampil data Proyek Masuk, tetapi jika tidak *valid* akan kembali ke kelola data Proyek Masuk. Admin dapat *logout* dan sistem menutup *database*.

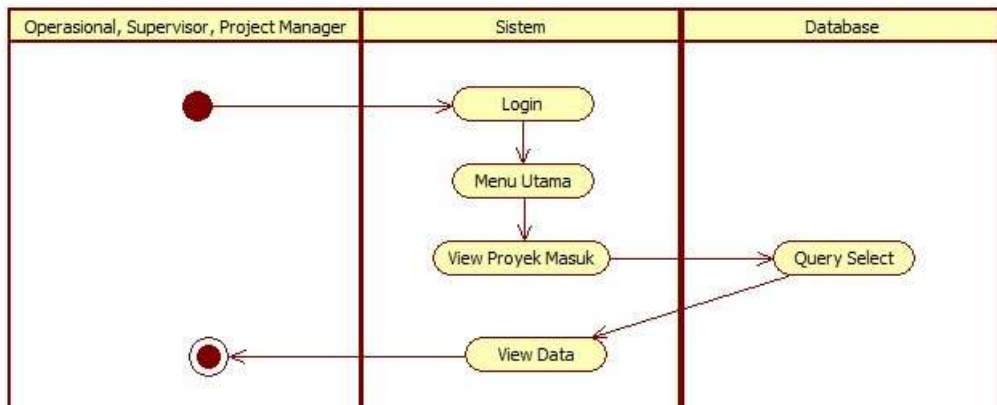
5.2.5.6 Activity Diagram View Kegiatan Proyek (UC6)



Gambar 5. 31 Activity Diagram View Kegiatan Proyek (UC6)

Setelah *login*, Operasional dan Supervisor memilih menu kegiatan proyek. Kemudian sistem akan menampilkan data kegiatan proyek. Aktor akan melakukan proses *view data* kegiatan proyek, selanjutnya di sistem akan ditampilkan.

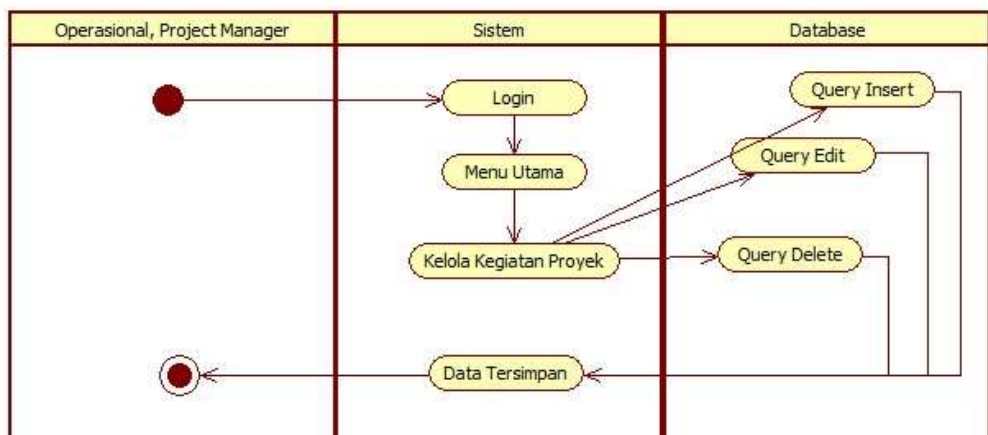
5.2.5.7 Activity Diagram View Proyek Masuk (UC7)



Gambar 5. 33 Activity Diagram View Proyek Masuk (UC7)

Setelah *login*, Operasional dan *Project Manager* memilih menu proyek masuk di. Kemudian sistem akan menampilkan data proyek masuk. Aktor akan melakukan proses *view data* proyek masuk.

5.2.5.8 Activity Diagram Kelola Kegiatan Proyek (UC8)

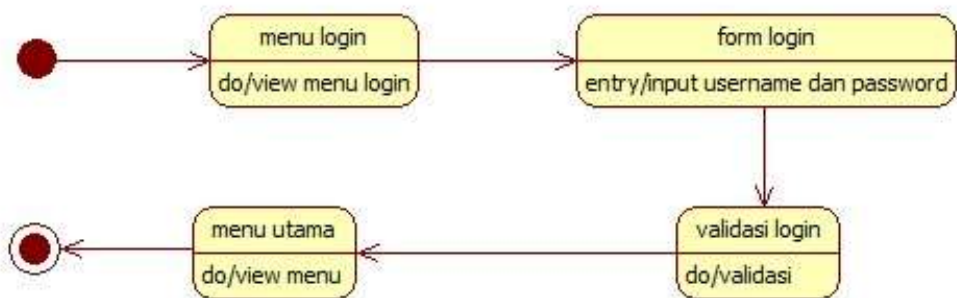


Gambar 5. 33 Activity Diagram Kelola Kegiatan Proyek (UC8)

Setelah *login*, Operasional dan *Project Manager* memilih menu kelola Kegiatan Proyek. Kemudian sistem akan menampilkan kelola Kegiatan Proyek. Aktor akan melakukan proses *insert*, *update*, dan *delete* data Kegiatan Proyek, selanjutnya di sistem akan disimpan data Kegiatan Proyek. Jika *valid* maka akan tampil data Kegiatan Proyek, tetapi jika tidak *valid* akan kembali ke kelola data Kegiatan Proyek. Aktor dapat *logout* dan sistem menutup *database*.

5.2.6 Statechart Diagram

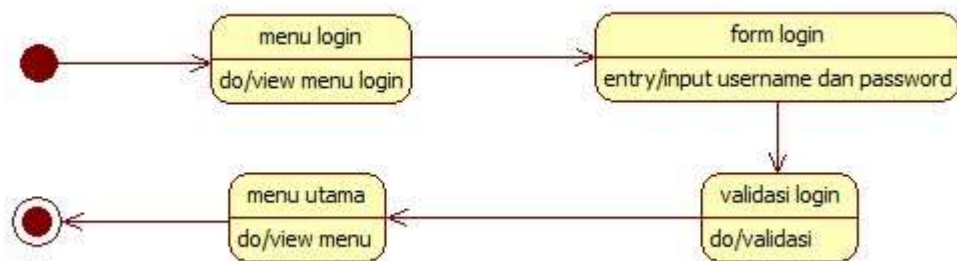
5.2.6.1 Statechart Diagram Login Operasional (UC1)



Gambar 5. 24 Statechart Diagram Login Operasional (UC1)

Operasional masuk ke menu *login* menginputkan *username* dan *password*. Jika *valid* maka akan masuk ke menu Operasional dan jika tidak *valid* akan masuk kembali ke *form login*.

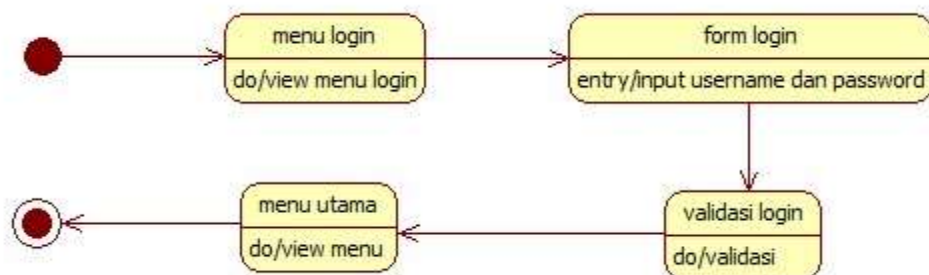
5.2.6.2 Statechart Diagram Login Supervisor (UC2)



Gambar 5. 25 Statechart Diagram Login Supervisor (UC2)

Supervisor masuk ke menu *login* menginputkan *username* dan *password*. Jika *valid* maka akan masuk ke menu Supervisor dan jika tidak *valid* akan masuk kembali ke *form login*.

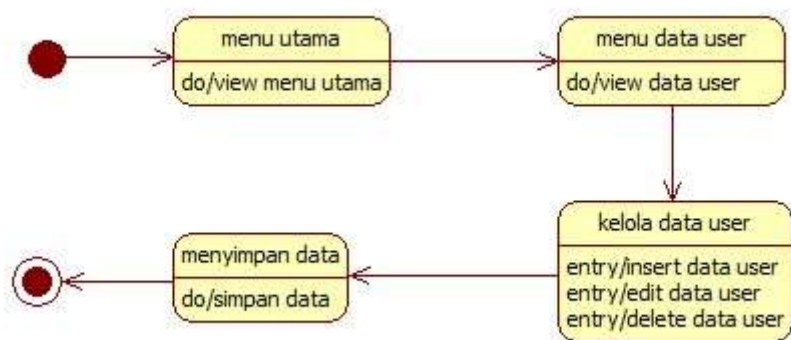
5.2.6.3 Statechart Diagram Login Project Manager (UC3)



Gambar 5. 26 Statechart Diagram Project Manager (UC3)

Project Manager masuk ke menu *login* menginputkan *username* dan *password*. Jika *valid* maka akan masuk ke menu *Project Manager* dan jika tidak *valid* akan masuk kembali ke *form login*.

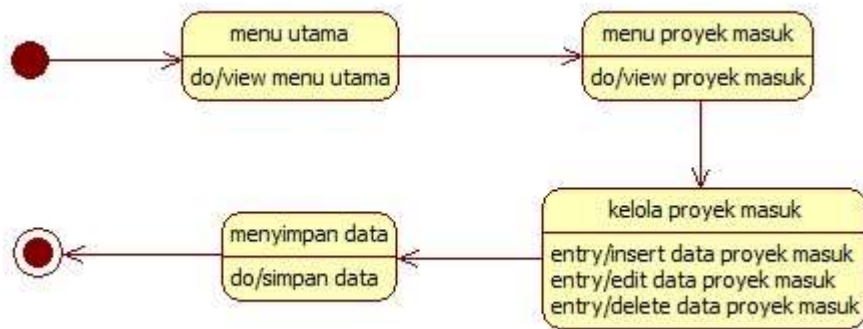
5.2.6.4 Statechart Diagram Kelola Data User (UC4)



Gambar 5. 27 Statechart Diagram Kelola Data User (UC4)

Operasional mengelola menu data user yang nanti di validasi ke *database* jika benar maka *query* berhasil.

5.2.6.5 Statechart Diagram Kelola Proyek Masuk (UC5)



Gambar 5. 28 Statechart Diagram Kelola Data Proyek Masuk (UC5)

Operasional mengelola menu data proyek masuk yang nanti di validasi ke *database* jika benar maka *query* berhasil.

5.2.6.6 Statechart Diagram View Kegiatan Proyek (UC6)



Gambar 5. 29 Statechart Diagram View Kegiatan Proyek (UC6)

Operasional dan Supervisor melihat kegiatan proyek yang sebelumnya sudah tersimpan di *database*.

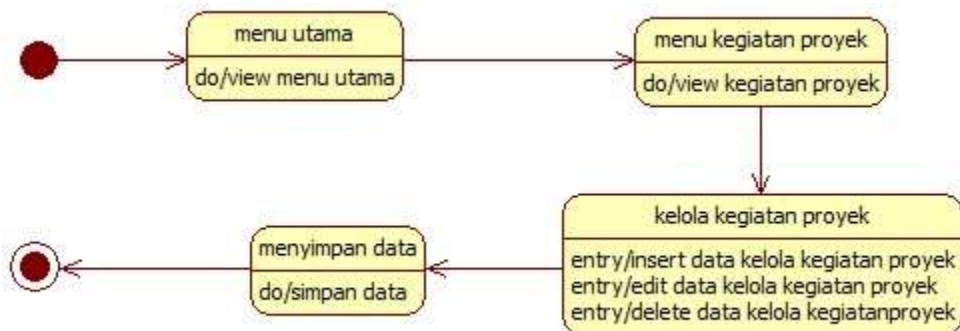
5.2.6.7 Statechart Diagram View Proyek Masuk (UC7)



Gambar 5. 30 Statechart Diagram View Proyek Masuk (UC7)

Operasional, Supervisor, dan *Project Manager* melihat proyek masuk yang sebelumnya sudah tersimpan di *database*.

5.2.6.8 Statechart Diagram Kelola Kegiatan Proyek (UC8)

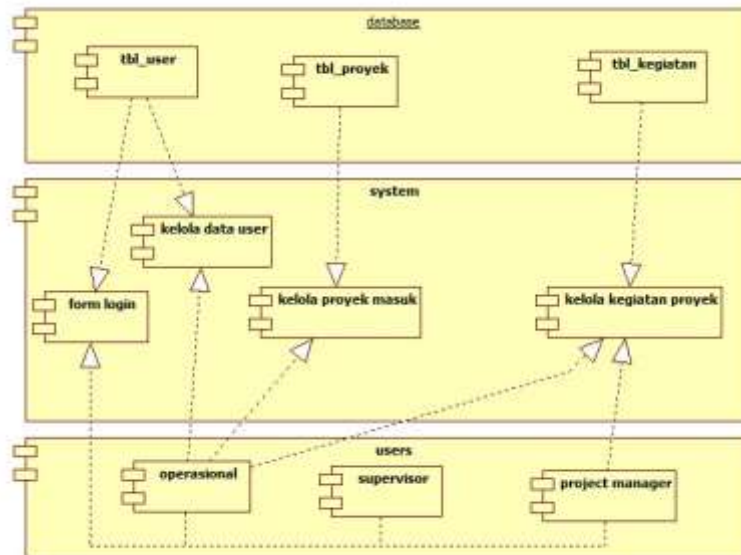


Gambar 5. 31 Statechart Diagram Kelola Kegiatan Proyek (UC8)

Operasional dan *Project Manager* mengelola menu kegiatan proyek yang nanti di validasi ke *database* jika benar maka *query* berhasil.

5.2.7 Component Diagram

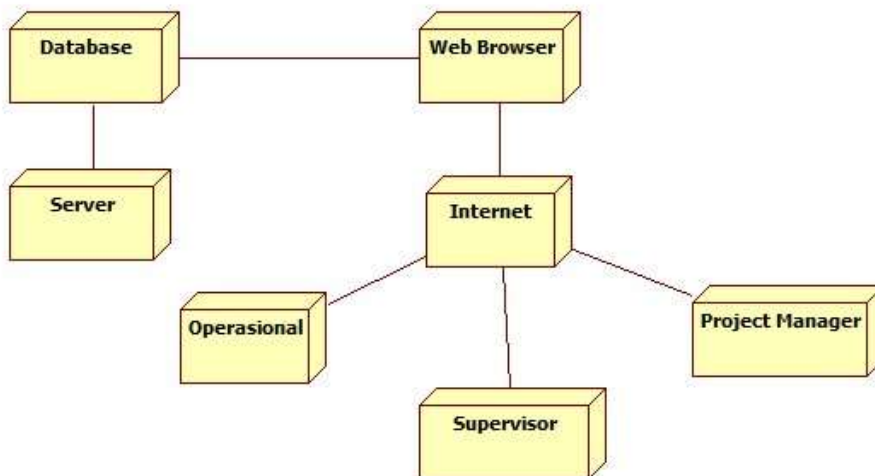
Hal penting pada *component* adalah *component* mewakili potongan-potongan yang independen yang bisa dipesan dan diperbaharui sewaktu-waktu. *Component Diagram* adalah sebuah bentuk pemodelan sebuah sistem kedalam komponen-komponen yang lebih rendah.



Gambar 5. 32 Component Diagram

5.2.8 Deployment Diagram

Deployment diagram menunjukkan tata letak sebuah sistem secara fisik, menampakkan bagian-bagian *software* yang berjalan pada bagian *hardware*. Bagian utama *hardware* atau perangkat keras adalah node yaitu nama umum untuk semua jenis sumber komputasi.



Gambar 5. 33 Deployment Diagram

6.1 Pengkajian

Dalam pengkajian ini, penulis akan mengkaji kembali data yang diperoleh dari hasil pengumpulan data yang telah dilakukan sebelumnya untuk mendukung analisis dari penelitian ini, untuk memberikan solusi. Pada Bab IV bagian 4.2 telah dijelaskan bahwa penulis melakukan analisis dengan beberapa tahapan dan metode yang digunakan yaitu *Critical Path Method (CPM)*.

6.2 Identifikasi Masalah

Pada identifikasi masalah yang telah dipaparkan sebelumnya yaitu untuk menentukan total durasi pengerjaan proyek dengan menggunakan *critical path method* untuk membantu mempermudah *Project Manager* dalam melakukan perencanaan dengan mendapatkan data *Project Completion*. Untuk data dan hasil dari pengolahan data tersebut dapat dilihat pada table dibawah ini :

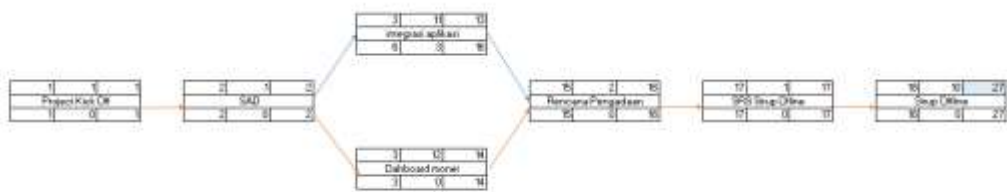
Tabel 6. 1 Data Project Completion LKPP-Sirup

Modul	Estimated Man Days	Predecessor
GENERAL	1	-
PROJECT KICK OFF		
SAD	1	Project Kick Off
INTEGRASI APLIKASI	11	SAD
Web Service SIMDA		
Create API untuk insert ke Database		
Update data RKA dan buatkan menjadi history		
display data RKA yang diupload dalam bentuk Tab		
Web Service SIMRAL		
Create API untuk insert ke Database		
Update Data RKA dan insert ke table History		
Fixing WS Om SPAN		
Integrasi KRISNA		
Update Data SIMDA menggunakan API		
Update Data SIMRAL menggunakan API		
Create paket menggunakan API		
DASHBOARD MONER	12	SAD
Rekap paket sesuai tagging krisna		
Optimasi query fitur statistic		
Bug Fixing OLAP		
Dashboard notifikasi tahun aktif satker /OPD		
Dashboard notifikasi perubahan pagu PKOR/PKOK		
Status upload file RKAD		
Rekap pengumuman RUP		
Notif perubahan paket		
Notif verifikasi akun PPK		

Rekap pengumuman RUP di PPK yang sama		
Rekap pengumuman RUP seluruh KLPD		
Model Rekapitulasi		
RENCANA PENGADAAN	2	SAD
Notifikasi Perubahan struktur anggaran		
Create paket dengan lintas anggaran kegiatan		
SRS SIRUP OFFLINE	1	Integrasi, Dashboard Moner, Rencana Pengadaan
SIRUP OFFLINE	10	SRS Sirup Offline
Sirup Installer		
Create Paket		
Download daftar paket		
Upload daftar paket		
Auto Upload daftar paket		
TOTAL MAIN DAYS	41	

6.3 Pengujian Metode

Pada tahap pengujian metode ini, dilakukan pengujian metode untuk mengetahui apakah *Critical Path Method* dapat menghasilkan total durasi pengerjaan proyek yang lebih optimal untuk membantu *Project Manajer* dalam melakukan perencanaan.



Gambar 6. 1 Perhitungan Critical Path Method

6.4 Evaluasi Metode

Pada data yang diperoleh hasil total waktu pengerjaan pada proyek LKPP-Sirup adalah 41 hari, sedangkan ketika diolah menggunakan Critical Path Method menghasilkan durasi waktu yang lebih optimal yaitu 27 hari.

9. Skenario Use Case Login Supervisor

Skenario dalam diagram *use case Login* ada pada tabel seperti berikut ini:

Tabel 5. 13 Skenario Use Case Login Supervisor

Identifikasi	
Nomor	UC2
Nama	Login Supervisor
Tujuan	Melakukan <i>Login</i> untuk masuk halaman utama.
Deskripsi	<i>Login</i> dengan memasukkan <i>Username</i> dan <i>Password</i>
Aktor	Supervisor
Skenario Utama	
Kondisi Awal	Menampilkan <i>Form Login</i> .

Aksi Aktor	Reaksi Sistem
1. Memasukkan <i>username</i> dan <i>password</i> .	2. <i>Form Login</i> akan menampilkan <i>textbox username, password</i> dan untuk <i>password</i> ditampilkan dalam bentuk kode ● pada layar untuk jaminan keamanan.
3. Aktor melakukan konfirmasi persetujuan terhadap <i>username, password</i> yang telah dimasukkan dengan menekan tombol <i>Login</i> .	4. Aplikasi melakukan validasi terhadap <i>username, password</i> yang telah dimasukkan oleh pengguna dengan melakukan pengecekan pada basis data.
Kondisi Akhir	Jika pada akhir interaksi <i>username, password</i> yang dimasukan pengguna <i>valid</i> maka pengguna akan langsung masuk ke halaman utama dan dapat menggunakan aplikasi sesuai hak aksesnya.
Alur Alternatif kondisi akhir a. Jika tidak <i>valid</i> maka Aplikasi akan kembali ke tampilan awal <i>login</i> .	

10. Skenario Use Case Login Project Manager

Skenario dalam diagram *use case Login* ada pada tabel seperti berikut ini:

Tabel 5. 14 Skenario Use Case Login Project Manager

Identifikasi	
Nomor	UC3
Nama	<i>Login Project Manager</i>
Tujuan	Melakukan <i>Login</i> untuk masuk halaman utama.
Deskripsi	<i>Login</i> dengan memasukan <i>Username</i> dan <i>Password</i>
Aktor	<i>Project Manager</i>
Skenario Utama	
Kondisi Awal	Menampilkan <i>Form Login</i> .
Aksi Aktor	Reaksi Sistem

1. Memasukkan <i>username</i> dan <i>password</i> .	2. <i>Form Login</i> akan menampilkan <i>textbox username, password</i> dan untuk <i>password</i> ditampilkan dalam bentuk kode '●' pada layar untuk jaminan keamanan.
3. Aktor melakukan konfirmasi persetujuan terhadap <i>username, password</i> yang telah dimasukkan dengan menekan tombol <i>Login</i> .	4. Aplikasi melakukan validasi terhadap <i>username, password</i> yang telah dimasukkan oleh pengguna dengan melakukan pengecekan pada basis data.
Kondisi Akhir	Jika pada akhir interaksi <i>username, password</i> yang dimasukan pengguna <i>valid</i> maka pengguna akan langsung masuk ke halaman utama dan dapat menggunakan aplikasi sesuai hak aksesnya.
Alur Alternatif kondisi akhir a. Jika tidak <i>valid</i> maka Aplikasi akan kembali ke tampilan awal <i>login</i> .	

11. Skenario Use Case Kelola Data User

Skenario dalam diagram *use case* Kelola Data User ada pada tabel seperti berikut ini:

Tabel 5. 15 Skenario Use Case Kelola Data User

Identifikasi	
Nomor	UC4
Nama	Kelola Data User
Tujuan	Melakukan manipulasi isi <i>record</i> data
Deskripsi	Melakukan CRUD terhadap data <i>user</i>
Aktor	Operasional
Skenario Utama	
Kondisi Awal	<i>Form login</i> terlebih dahulu. Jika <i>valid</i> maka masuk ke <i>form</i> halaman utama Operasional jika tidak <i>valid</i> akan kembali ke tampilan awal <i>login</i> .
Aksi Aktor	Reaksi Sistem

1. Operasional memilih <i>link</i> ke halaman yang akan dimanipulasi.	2. Aplikasi menampilkan <i>form</i> sesuai pilihan pengguna.
3. Operasional melakukan manipulasi isi <i>record</i> dengan memilih tombol tambah, edit atau hapus.	4. Aplikasi menampilkan pesan konfirmasi perintah untuk menyimpan data hasil penambahan, perubahan atau penghapusan data <i>user</i> .
5. Admin melakukan konfirmasi persetujuan manipulasi isi <i>record</i> .	6. Aplikasi menyimpan hasil manipulasi pada basis data tabel <i>tbl_user</i> atau yang terkait.
Kondisi Akhir	Isi informasi <i>record</i> pada basis data tabel <i>tbl_user</i> yang dilakukan manipulasi mengalami perubahan.
Alur alternatif no.4 a. Jika data yang dimasukkan tidak sesuai dengan tipe data yang bisa diterima basis data maka akan dimunculkan pesan <i>error</i> . b. Jika ada <i>textbox</i> yang wajib diisi masih kosong maka akan muncul pesan peringatan untuk melakukan pengecekan ulang	

12. Skenario Use Case Kelola Proyek Masuk

Skenario dalam diagram *use case* Kelola Status Masakan ada pada tabel seperti berikut ini:

Tabel 5. 16 Skenario Use Case Kelola Proyek Masuk

Identifikasi	
Nomor	UC5
Nama	Kelola Data <i>User</i>
Tujuan	Melakukan manipulasi isi <i>record</i> data
Deskripsi	Melakukan CRUD terhadap data proyek masuk
Aktor	Operasional
Skenario Utama	

Kondisi Awal	<i>Form login</i> terlebih dahulu. Jika <i>valid</i> maka masuk ke <i>form</i> halaman utama Operasional jika tidak <i>valid</i> akan kembali ke tampilan awal <i>login</i> .
Aksi Aktor	Reaksi Sistem
1. Operasional memilih <i>link</i> ke halaman yang akan dimanipulasi.	2. Aplikasi menampilkan <i>form</i> sesuai pilihan pengguna.
3. Operasional melakukan manipulasi isi <i>record</i> dengan memilih tombol tambah, edit atau hapus.	4. Aplikasi menampilkan pesan konfirmasi perintah untuk menyimpan data hasil penambahan, perubahan atau penghapusan data <i>proyek masuk</i> .
5. Admin melakukan konfirmasi persetujuan manipulasi isi <i>record</i> .	6. Aplikasi menyimpan hasil manipulasi pada basis data tabel <i>tbl_proyek</i> atau yang terkait.
Kondisi Akhir	Isi informasi <i>record</i> pada basis data tabel <i>tbl_proyek</i> yang dilakukan manipulasi mengalami perubahan.
Alur alternatif no.4 a. Jika data yang dimasukkan tidak sesuai dengan tipe data yang bisa diterima basis data maka akan dimunculkan pesan <i>error</i> . b. Jika ada <i>textbox</i> yang wajib diisi masih kosong maka akan muncul pesan peringatan untuk melakukan pengecekan ulang	

13. Skenario *Use Case View* Kegiatan Proyek

Skenario dalam diagram *use case View* Kegiatan Proyek ada pada tabel seperti berikut ini:

Tabel 5. 17 Skenario *Use Case View* Kegiatan Proyek

Identifikasi	
Nomor	UC6
Nama	View Kegiatan Proyek
Tujuan	Melihat isi <i>record</i> data
Deskripsi	<i>View record</i> data yang telah diinputkan
Aktor	Operasional, Supervisor

Skenario Utama	
Kondisi Awal	<i>Form login</i> terlebih dahulu. Jika <i>valid</i> maka masuk ke <i>form</i> halaman utama untuk kasir jika tidak <i>valid</i> akan kembali ke tampilan awal <i>login</i> .
Aksi Aktor	Reaksi Sistem
1. Staf memilih <i>link</i> ke halaman yang akan dilihat.	2. Aplikasi menampilkan data sesuai pilihan pengguna.
Kondisi Akhir	Isi informasi <i>record</i> pada basis data tabel <i>tbl_kegiatan</i> ditampilkan.

14. Skenario *Use Case View* Proyek Masuk

Skenario dalam diagram *use case View* Proyek Masuk ada pada tabel seperti berikut ini:

Tabel 5. 18 Skenario *Use Case View* Proyek Masuk

Identifikasi	
Nomor	UC7
Nama	View Proyek Masuk
Tujuan	Melihat isi <i>record</i> data
Deskripsi	<i>View record</i> data yang telah diinputkan
Aktor	Operasional, Supervisor, <i>Project Manager</i>
Skenario Utama	
Kondisi Awal	<i>Form login</i> terlebih dahulu. Jika <i>valid</i> maka masuk ke <i>form</i> halaman utama untuk kasir jika tidak <i>valid</i> akan kembali ke tampilan awal <i>login</i> .
Aksi Aktor	Reaksi Sistem
1. Staf memilih <i>link</i> ke halaman yang akan dilihat.	2. Aplikasi menampilkan data sesuai pilihan pengguna.
Kondisi Akhir	Isi informasi <i>record</i> pada basis data tabel <i>tbl_proyek</i> ditampilkan.

15. Skenario *Use Case* Kelola Kegiatan Proyek

Skenario dalam diagram *use case* Kelola Kegiatan Proyek ada pada tabel seperti berikut ini:

Tabel 5. 19 Skenario Use Case Kelola Kegiatan Proyek

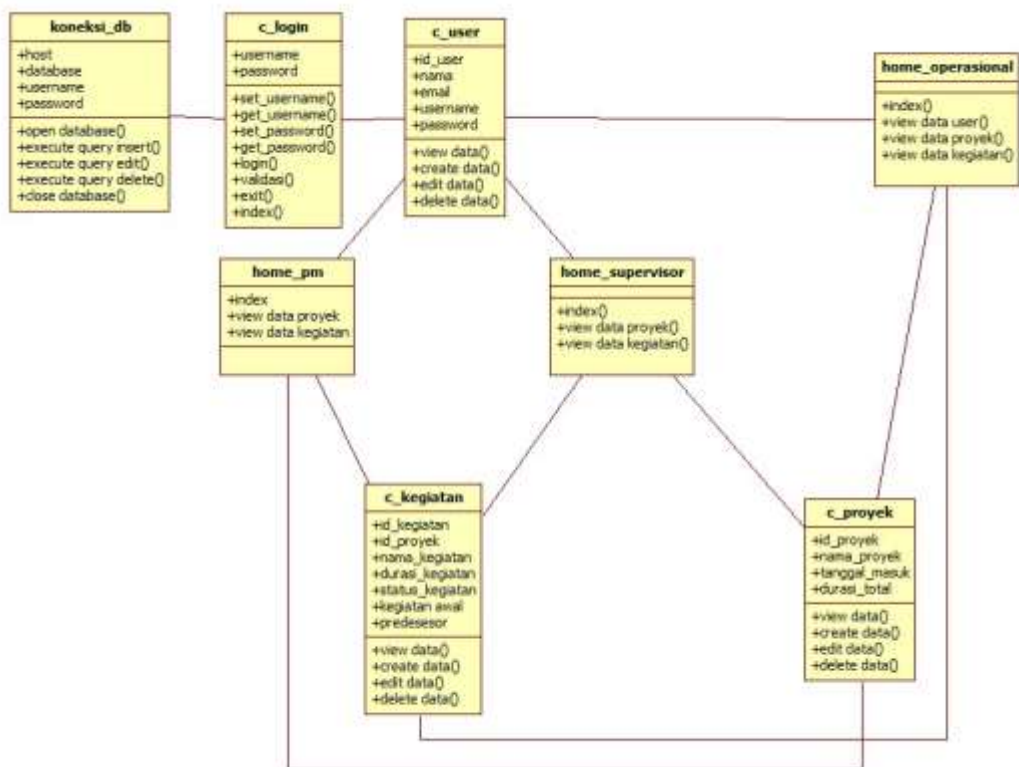
Identifikasi	
Nomor	UC8
Nama	Kelola Kegiatan Proyek
Tujuan	Melakukan manipulasi isi <i>record</i> data
Deskripsi	Melakukan CRUD terhadap data kegiatan proyek
Aktor	Operasional, <i>Project Manager</i>
Skenario Utama	
Kondisi Awal	<i>Form login</i> terlebih dahulu. Jika <i>valid</i> maka masuk ke <i>form</i> halaman utama Operasional jika tidak <i>valid</i> akan kembali ke tampilan awal <i>login</i> .
Aksi Aktor	Reaksi Sistem
1. Operasional memilih <i>link</i> ke halaman yang akan dimanipulasi.	2. Aplikasi menampilkan <i>form</i> sesuai pilihan pengguna.
3. Operasional melakukan manipulasi isi <i>record</i> dengan memilih tombol tambah, edit atau hapus.	4. Aplikasi menampilkan pesan konfirmasi perintah untuk menyimpan data hasil penambahan, perubahan atau penghapusan data <i>kegiatan proyek</i> .
5. Admin melakukan konfirmasi persetujuan manipulasi isi <i>record</i> .	6. Aplikasi menyimpan hasil manipulasi pada basis data tabel <i>tbl_kegiatan</i> atau yang terkait.
Kondisi Akhir	Isi informasi <i>record</i> pada basis data tabel <i>tbl_kegiatan</i> yang dilakukan manipulasi mengalami perubahan.
Alur alternatif no.4	
a. Jika data yang dimasukkan tidak sesuai dengan tipe data yang bisa diterima basis data maka akan dimunculkan pesan <i>error</i> .	

b. Jika ada *textbox* yang wajib diisi masih kosong maka akan muncul pesan peringatan untuk melakukan pengecekan ulang

5.2.2 Class Diagram

Class diagram adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek.

Dalam proses pembuatan aplikasi ini *class diagram* yang akan digunakan adalah sebagai berikut:

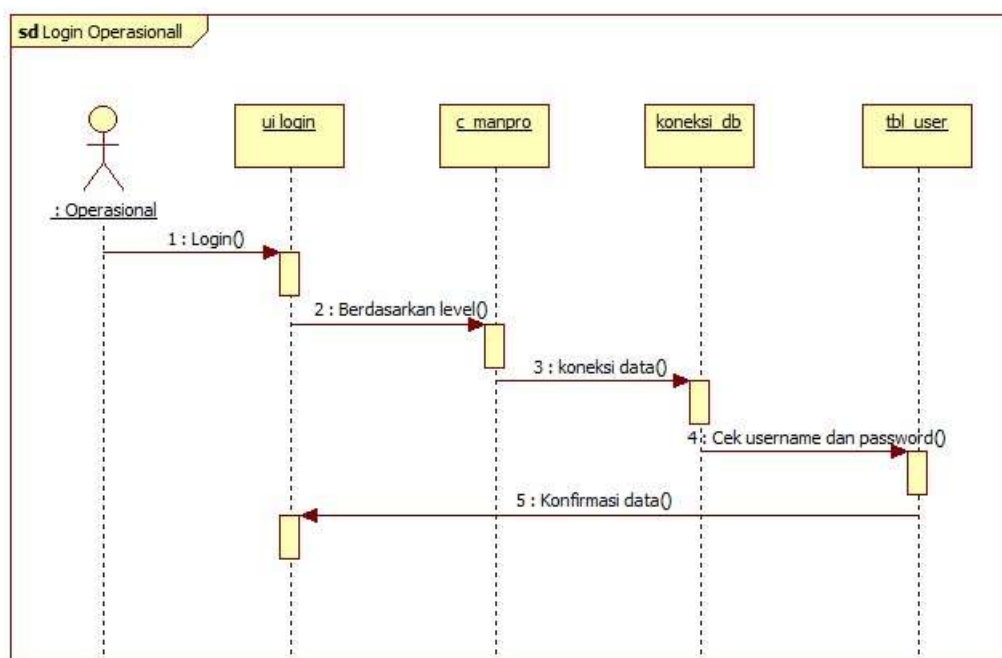


Gambar 5. 34 Class Diagram

5.2.3 Sequence Diagram

Sequence diagram adalah suatu diagram yang menggambarkan interaksi antar objek dan mengindikasikan komunikasi diantara objek-objek di dalam dan disekitar sistem.

5.2.3.1 Sequence Diagram Login Operasional (UC1)

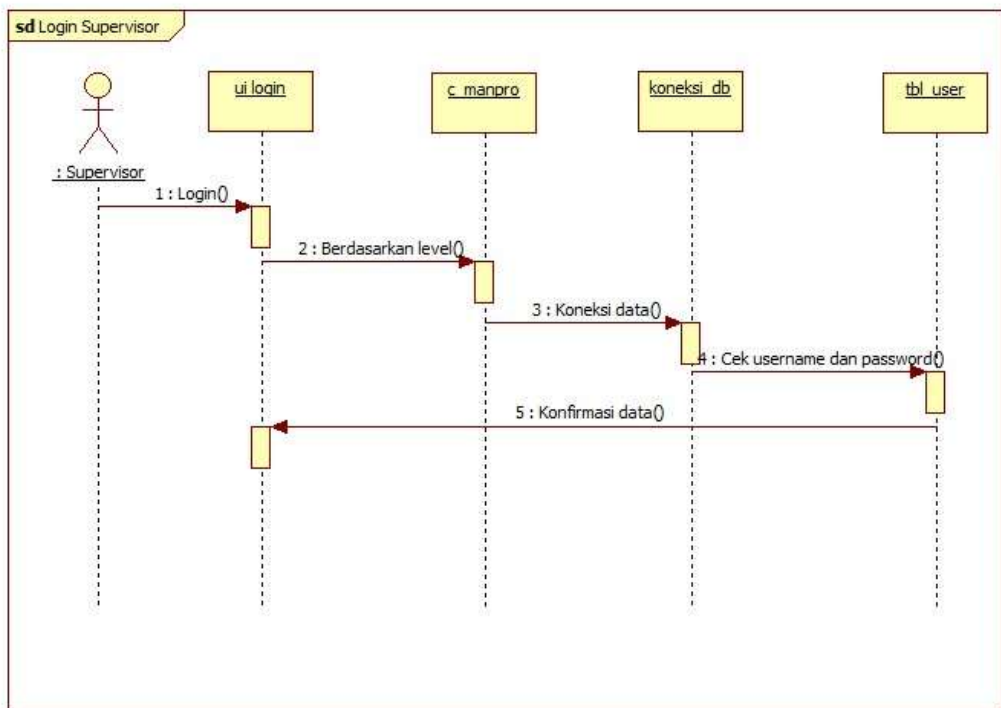


Gambar 5. 35 Sequence Diagram Login Operasional (UC1)

Pada *sequence diagram* ini menjelaskan proses *login*. Aktor mulai menjalankan aplikasi dan akan tampil *form login*. Kemudian aktor memasukkan *username dan password*, pada *interface login* data yang di

input akan di cek di tabel `tbl_user` di *database*. Jika sesuai maka akan mendapatkan tampilan form utama ke pengguna.

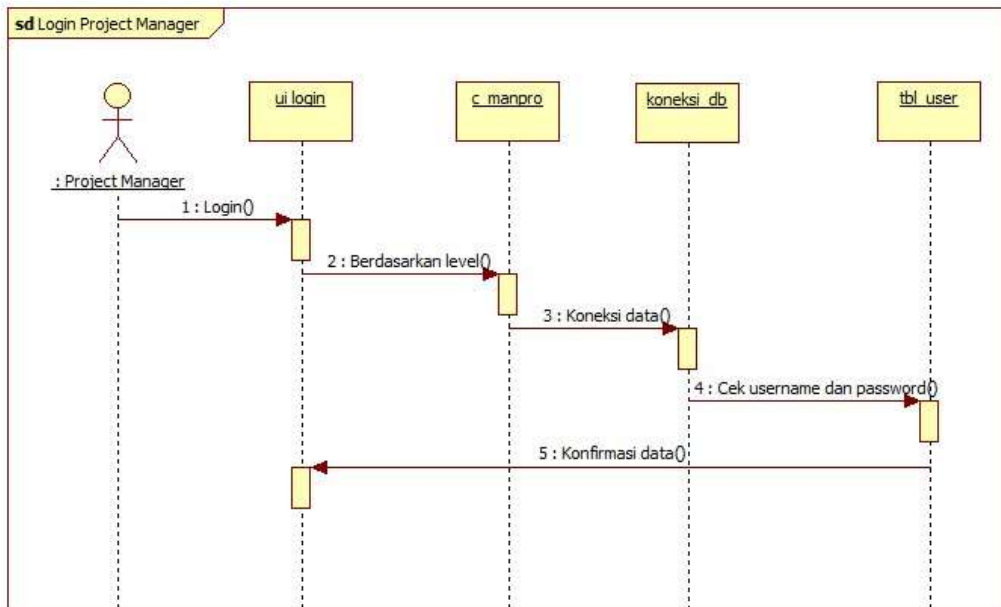
5.2.3.2 Sequence Diagram Login Supervisor (UC2)



Gambar 5. 36 Sequence Diagram Login Supervisor(UC2)

Pada *sequence diagram* ini menjelaskan proses *login*. Aktor mulai menjalankan aplikasi dan akan tampil *form login*. Kemudian aktor memasukkan *username dan password*, pada *interface login* data yang di *input* akan di cek di tabel `tbl_user` di *database*. Jika sesuai maka akan mendapatkan tampilan form utama ke pengguna.

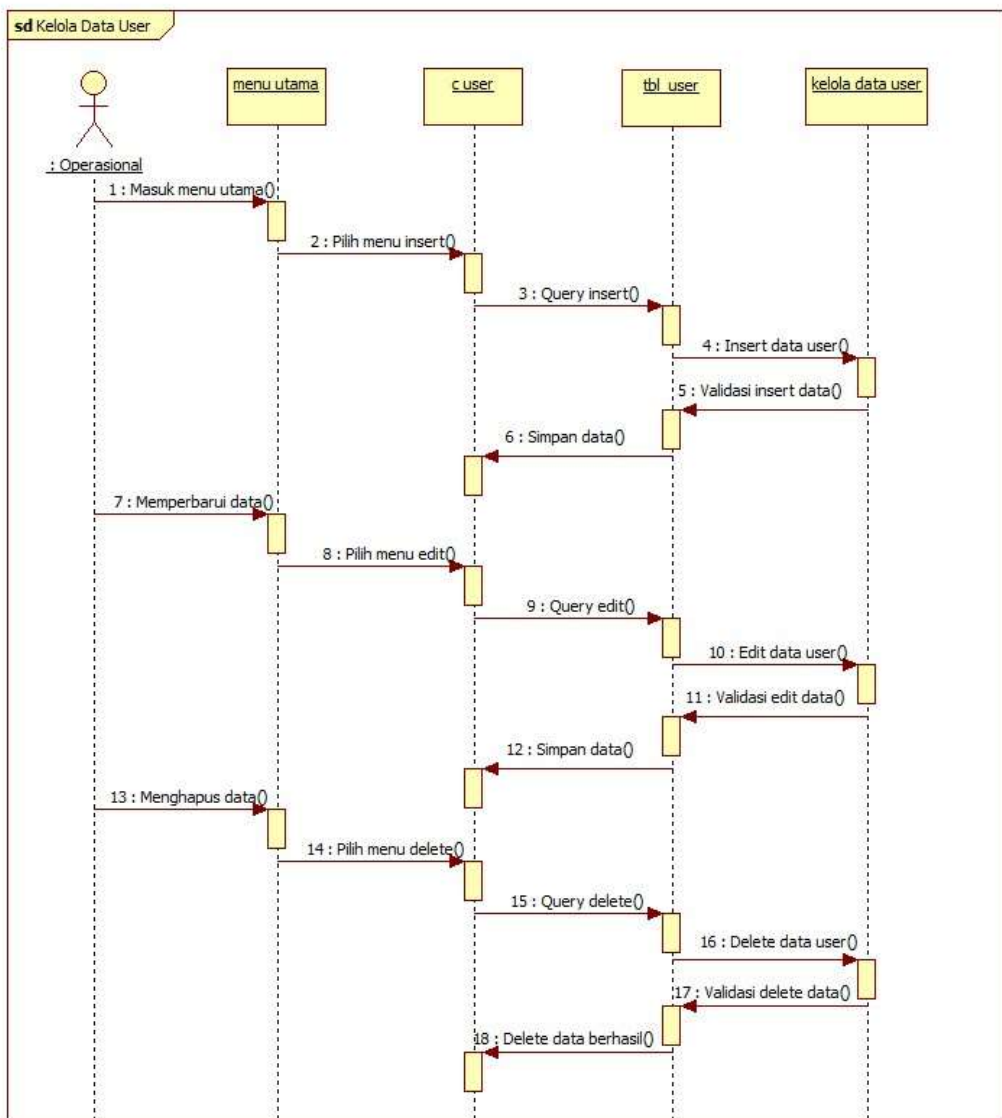
5.2.3.3 Sequence Diagram Login Project Manager (UC3)



Gambar 5. 37 Sequence Diagram Login Project Manager (UC3)

Pada *sequence diagram* ini menjelaskan proses *login*. Aktor mulai menjalankan aplikasi dan akan tampil *form login*. Kemudian aktor memasukkan *username dan password*, pada *interface login* data yang di input akan di cek di tabel **tbl_user** di *database*. Jika sesuai maka akan mendapatkan tampilan form utama ke pengguna.

5.2.3.4 Sequence Diagram Kelola Data User (UC4)

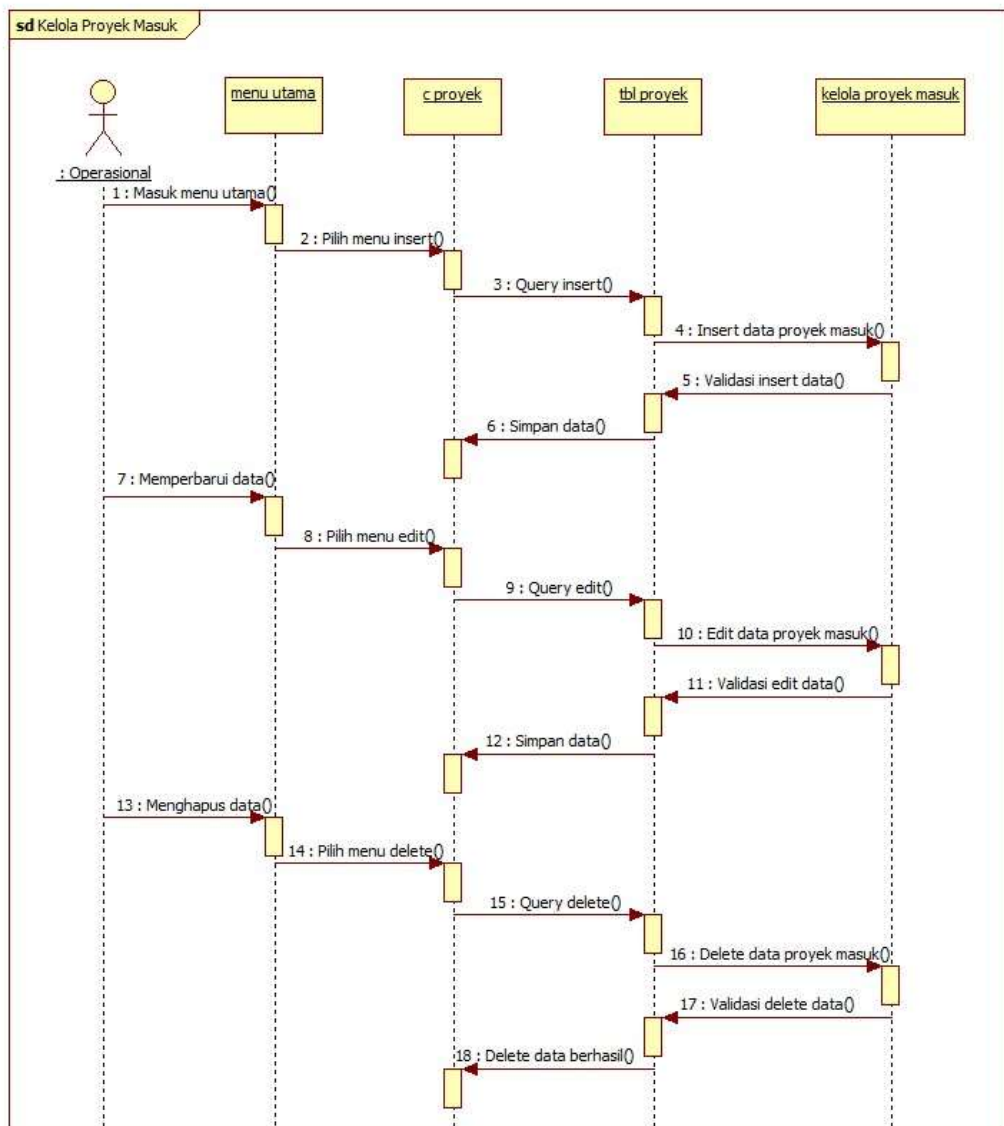


Gambar 5. 38 Sequence Diagram Kelola Data User (UC4)

Pada *sequence diagram* ini menjelaskan proses kelola data *user*. Aktor melakukan *login* terlebih dahulu. Selanjutnya sistem akan memuat *form* menu utama. Untuk memulai kelola ini, masuk ke menu kelola data *user*,

kemudian sistem akan melakukan proses koneksi database dan dilanjutkan dengan proses *OpenTable* ke tabel yang dituju yaitu *tbl_user* dengan melakukan operasi *insert()*, *update()*, *delete()*. Selanjutnya sistem akan memvalidasi hasil dari operasi, apakah berhasil atau gagal. Dan yang terakhir sistem akan melakukan proses *CloseDB* untuk menutup koneksi.

5.2.3.5 Sequence Diagram Kelola Proyek Masuk (UC5)

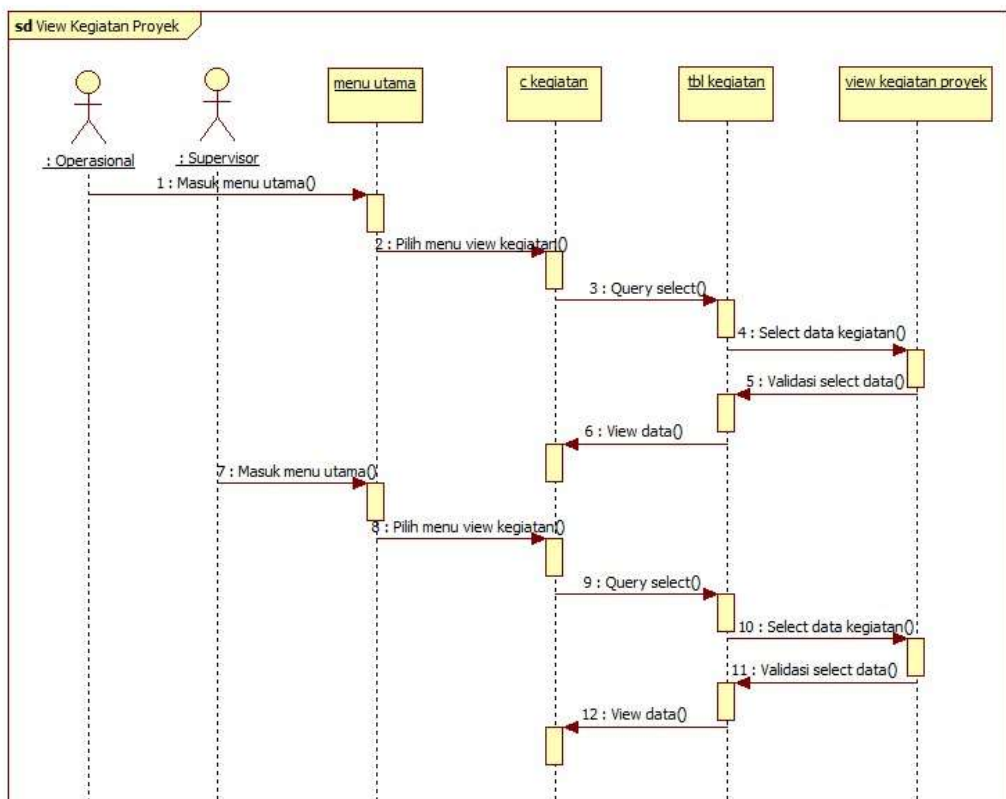


Gambar 5. 39 Sequence Diagram Kelola Proyek Masuk (UC5)

Pada *sequence diagram* ini menjelaskan proses kelola proyek masuk. Aktor melakukan *login* terlebih dahulu. Selanjutnya sistem akan memuat *form* menu utama. Untuk memulai kelola ini, masuk ke menu proyek masuk

kemudian sistem akan melakukan proses koneksi *database* dan dilanjutkan dengan proses *OpenTable* ke tabel yang dituju yaitu *tbl_proyek* dengan melakukan operasi *insert()*, *update()*, *delete()*. Selanjutnya sistem akan memvalidasi hasil dari operasi, apakah berhasil atau gagal. Dan yang terakhir sistem akan melakukan proses *CloseDB* untuk menutup koneksi.

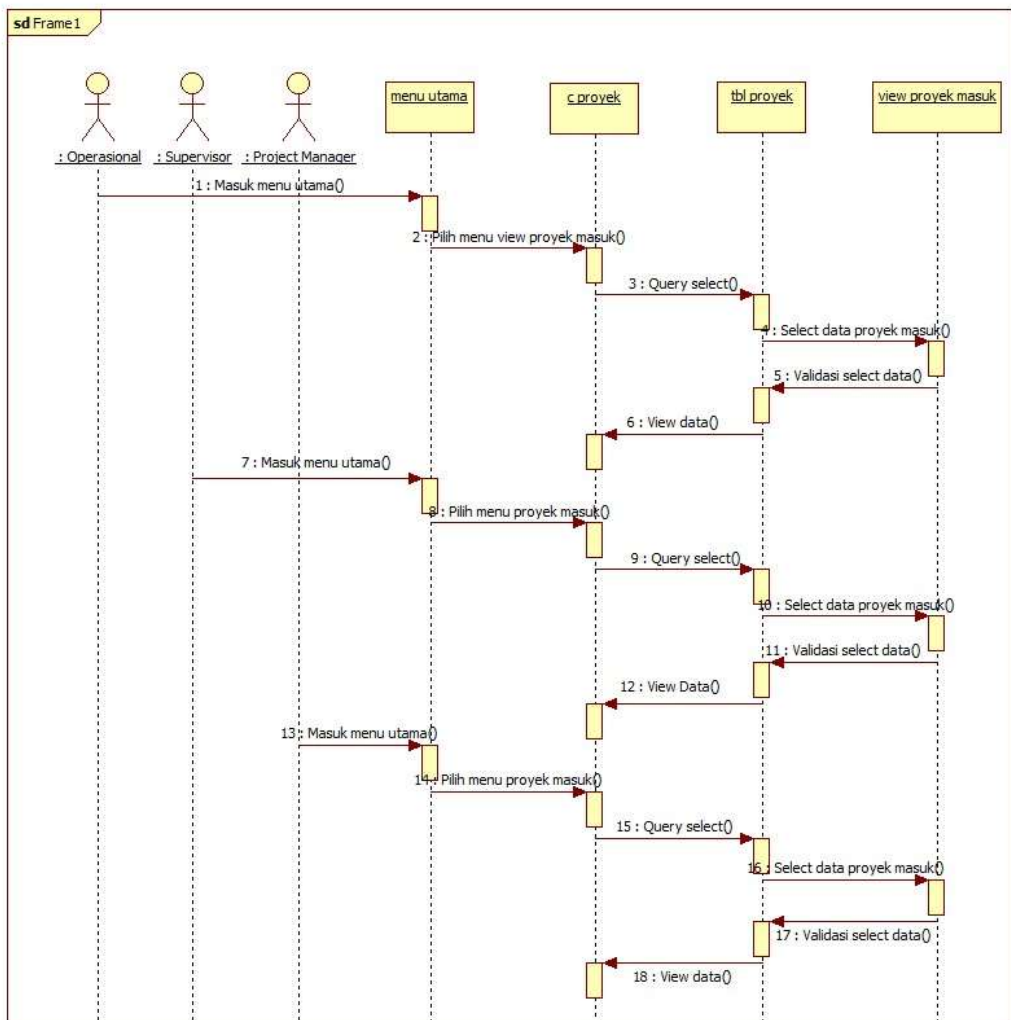
5.2.3.6 Sequence Diagram View Kegiatan Proyek (UC6)



Gambar 5. 40 Sequence Diagram View Kegiatan Proyek (UC6)

Pada *sequence diagram* ini menjelaskan proses *view data* kegiatan proyek. Aktor melakukan *login* terlebih dahulu. Selanjutnya sistem akan memuat *form* menu utama. Untuk memulai proses ini, masuk ke menu data kegiatan, kemudian sistem akan melakukan proses koneksi *database* dan dilanjutkan dengan proses *OpenTable* ke tabel yang dituju yaitu *tbl_kegiatan* dengan melakukan operasi *select()*. Selanjutnya sistem akan memvalidasi hasil dari operasi, apakah berhasil atau gagal. Dan yang terakhir sistem akan melakukan proses *CloseDB* untuk menutup koneksi.

5.2.3.7 Sequence Diagram View Proyek Masuk (UC7)

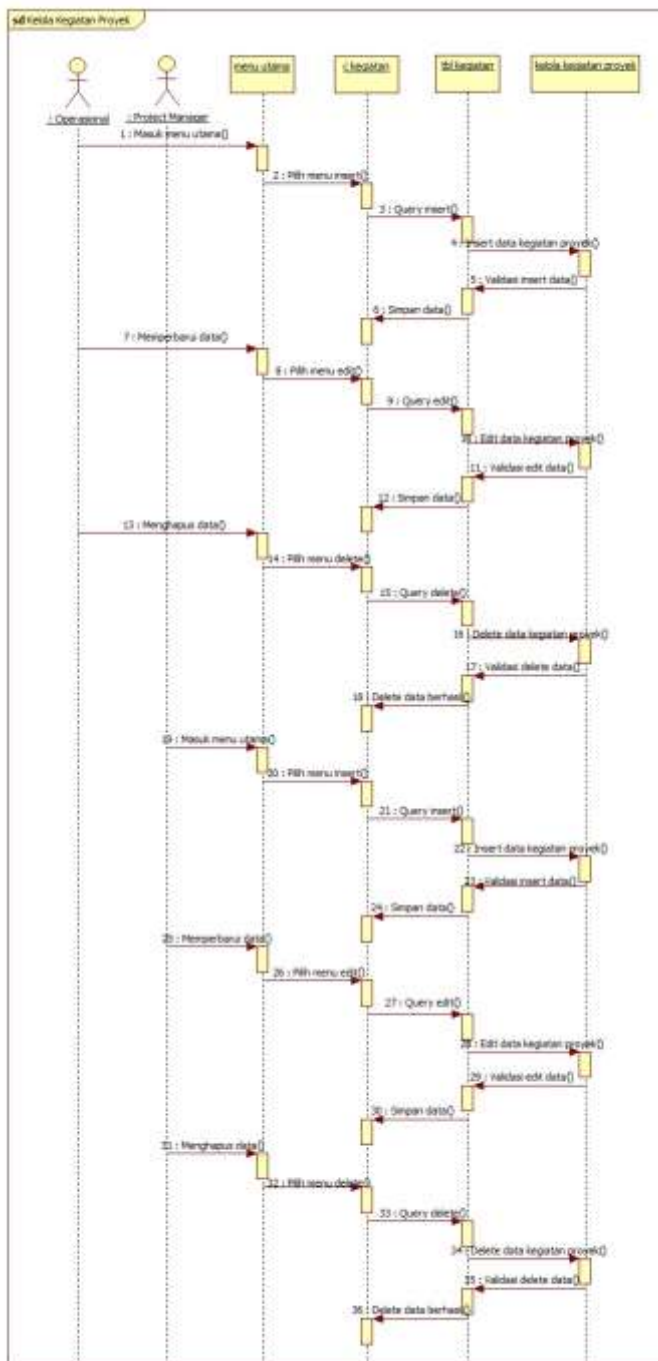


Gambar 5. 41 Sequence Diagram View Proyek Masuk (UC7)

Pada *sequence diagram* ini menjelaskan proses *view data* proyek masuk. Aktor melakukan *login* terlebih dahulu. Selanjutnya sistem akan memuat *form* menu utama. Untuk memulai proses ini, masuk ke menu data proyek, kemudian sistem akan melakukan proses koneksi *database* dan dilanjutkan dengan proses *OpenTable* ke tabel yang dituju yaitu *tbl_proyek*

dengan melakukan operasi *select()*. Selanjutnya sistem akan memvalidasi hasil dari operasi, apakah berhasil atau gagal. Dan yang terakhir sistem akan melakukan proses *CloseDB* untuk menutup koneksi.

5.2.3.8 Sequence Diagram Kelola Kegiatan Proyek (UC8)



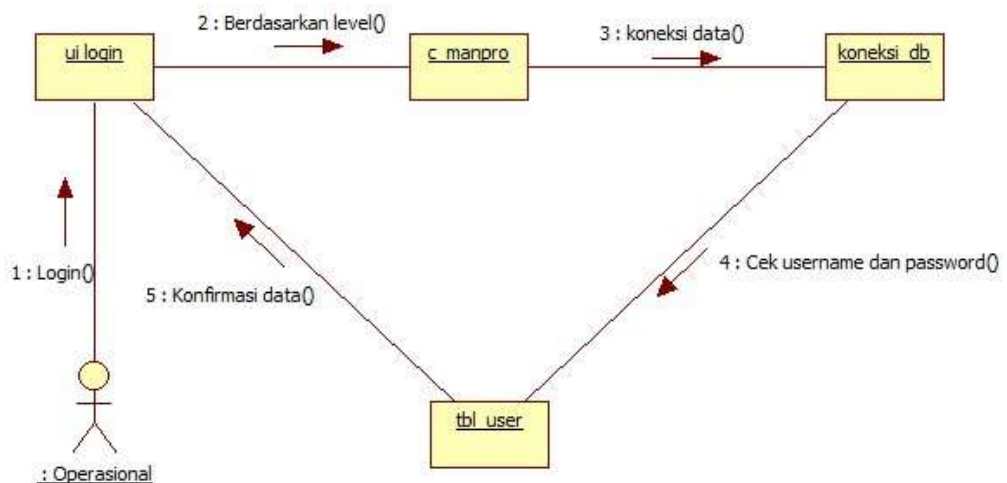
Gambar 5. 42 Sequence Diagram Kelola Kegiatan Proyek (UC8)

Pada *sequence diagram* ini menjelaskan proses kelola kegiatan proyek. Aktor melakukan *login* terlebih dahulu. Selanjutnya sistem akan memuat *form* menu utama. Untuk memulai kelola ini, masuk ke menu kelola kegiatan proyek, kemudian sistem akan melakukan proses koneksi *database* dan dilanjutkan dengan proses *OpenTable* ke tabel yang dituju yaitu *tbl_kegiatan* dengan melakukan operasi *update()*. Selanjutnya sistem akan memvalidasi hasil dari operasi, apakah berhasil atau gagal. Dan yang terakhir sistem akan melakukan proses *CloseDB* untuk menutup koneksi.

5.2.4 Communication Diagram

Communication diagram adalah sebuah interaksi yang memberi tekanan pada hubungan-hubungan data antar partisipan yang berbeda dalam sebuah interaksi.

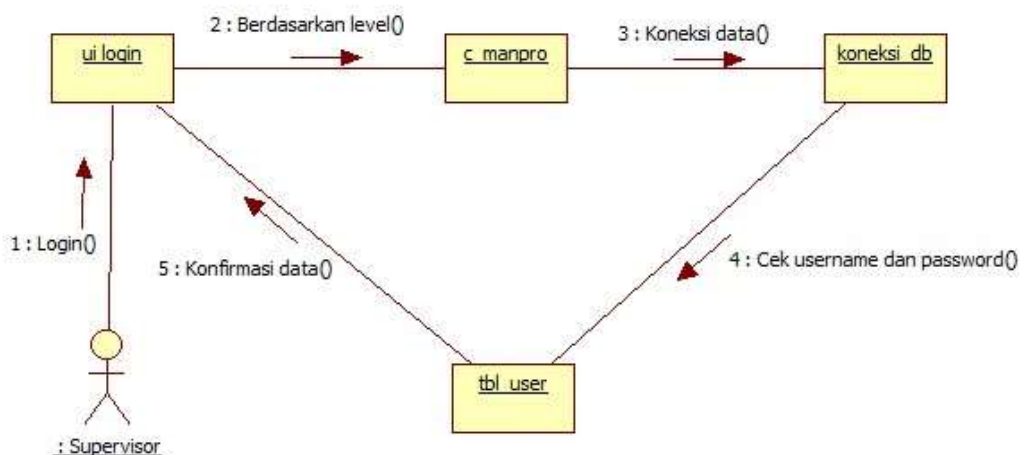
5.2.3.1 Communication Diagram Login Operasional (UC1)



Gambar 5. 43 Communication Diagram Login Operasional (UC1)

Pada *Communication diagram* ini menjelaskan proses *login*. Aktor mulai menjalankan aplikasi dan akan tampil *form login*. Kemudian aktor memasukkan *username dan password*, pada *interface login* data yang di *input* akan di cek di tabel *tbl_user* di *database*. Jika sesuai maka akan mendapatkan tampilan form utama ke pengguna.

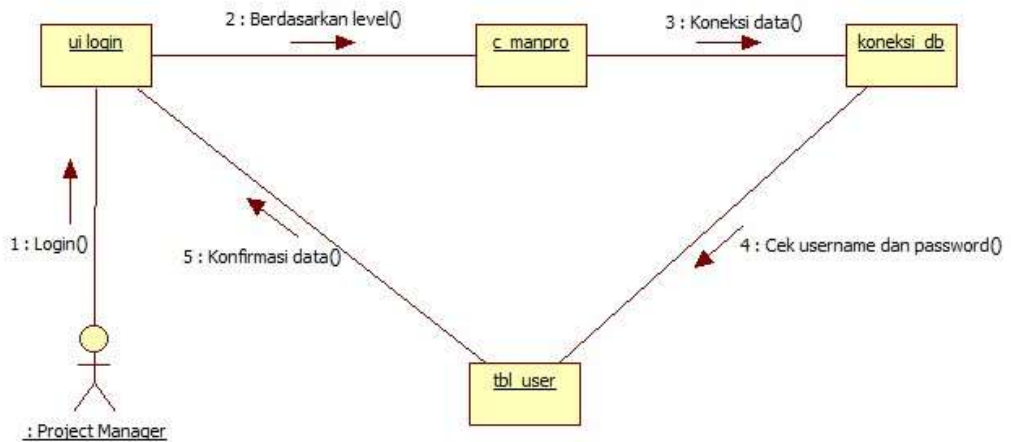
5.2.3.2 Communication Diagram Login Supervisor (UC2)



Gambar 5. 44 Communication Diagram Login Supervisor (UC2)

Pada *Communication diagram* ini menjelaskan proses *login*. Aktor mulai menjalankan aplikasi dan akan tampil *form login*. Kemudian aktor memasukkan *username dan password*, pada *interface login* data yang di *input* akan di cek di tabel *tbl_user* di *database*. Jika sesuai maka akan mendapatkan tampilan form utama ke pengguna.

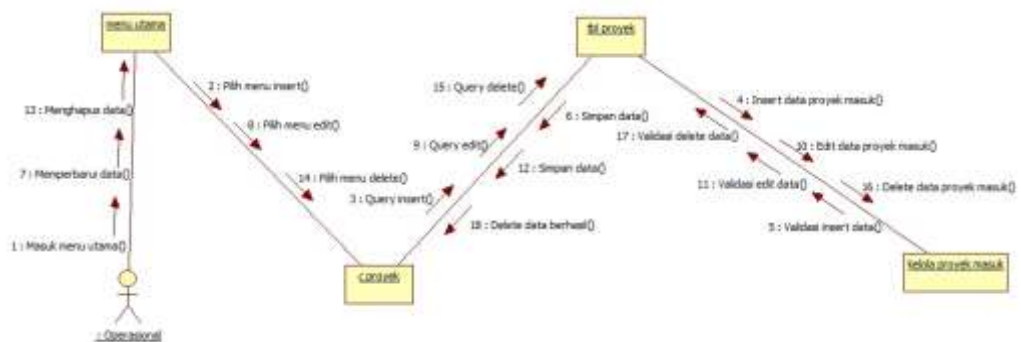
5.2.3.3 Communication Diagram Login Project Manager (UC3)



Gambar 5. 45 Communication Diagram Login Project Manager (UC3)

Pada *Communication diagram* ini menjelaskan proses *login*. Aktor mulai menjalankan aplikasi dan akan tampil *form login*. Kemudian aktor memasukkan *username dan password*, pada *interface login* data yang di input akan di cek di tabel `tbl_user` di *database*. Jika sesuai maka akan mendapatkan tampilan form utama ke pengguna.

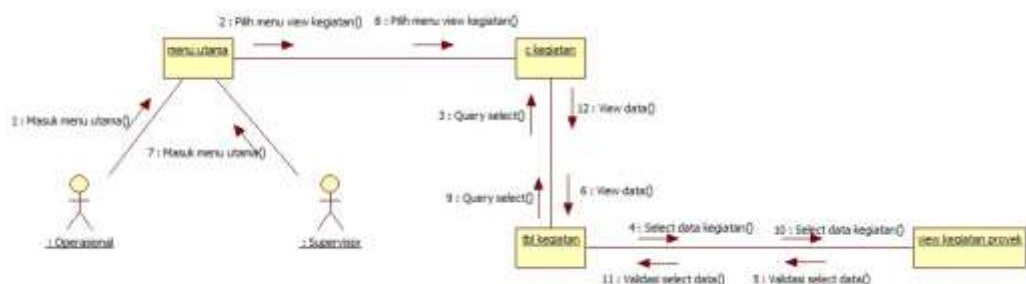
5.2.3.4 Communication Diagram Kelola Data User (UC4)



Gambar 5. 47 Communication Diagram Kelola Proyek Masuk (UC5)

Pada *Communication diagram* ini menjelaskan proses kelola proyek masuk. Aktor melakukan *login* terlebih dahulu. Selanjutnya sistem akan memuat *form* menu utama. Untuk memulai kelola ini, masuk ke menu proyek masuk kemudian sistem akan melakukan proses koneksi *database* dan dilanjutkan dengan proses *OpenTable* ke tabel yang dituju yaitu *tbl_proyek* dengan melakukan operasi *insert()*, *update()*, *delete()*. Selanjutnya sistem akan memvalidasi hasil dari operasi, apakah berhasil atau gagal. Dan yang terakhir sistem akan melakukan proses *CloseDB* untuk menutup koneksi.

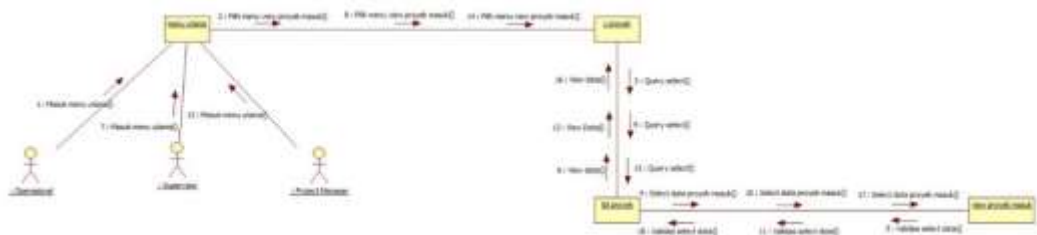
5.2.3.6 Communication Diagram View Kegiatan Proyek (UC6)



Gambar 5. 48 Communication Diagram View Kegiatan Proyek (UC6)

Pada *Communication diagram* ini menjelaskan proses *view data* kegiatan proyek. Aktor melakukan *login* terlebih dahulu. Selanjutnya sistem akan memuat *form* menu utama. Untuk memulai proses ini, masuk ke menu data kegiatan, kemudian sistem akan melakukan proses koneksi *database* dan dilanjutkan dengan proses *OpenTable* ke tabel yang dituju yaitu *tbl_kegiatan* dengan melakukan operasi *select()*. Selanjutnya sistem akan memvalidasi hasil dari operasi, apakah berhasil atau gagal. Dan yang terakhir sistem akan melakukan proses *CloseDB* untuk menutup koneksi.

5.2.3.7 Communication Diagram View Proyek Masuk (UC7)

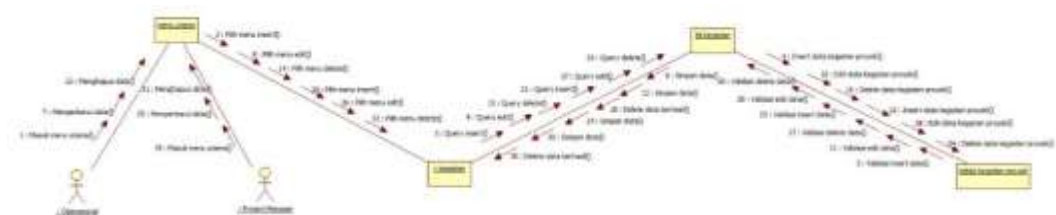


Gambar 5. 49 Communication Diagram View Proyek Masuk (UC7)

Pada *Communication diagram* ini menjelaskan proses *view data* proyek masuk. Aktor melakukan *login* terlebih dahulu. Selanjutnya sistem akan memuat *form* menu utama. Untuk memulai proses ini, masuk ke menu data proyek, kemudian sistem akan melakukan proses koneksi *database* dan dilanjutkan dengan proses *OpenTable* ke tabel yang dituju yaitu *tbl_proyek*

dengan melakukan operasi *select()*. Selanjutnya sistem akan memvalidasi hasil dari operasi, apakah berhasil atau gagal. Dan yang terakhir sistem akan melakukan proses *CloseDB* untuk menutup koneksi.

5.2.3.8 Communication Diagram Kelola Kegiatan Proyek (UC8)



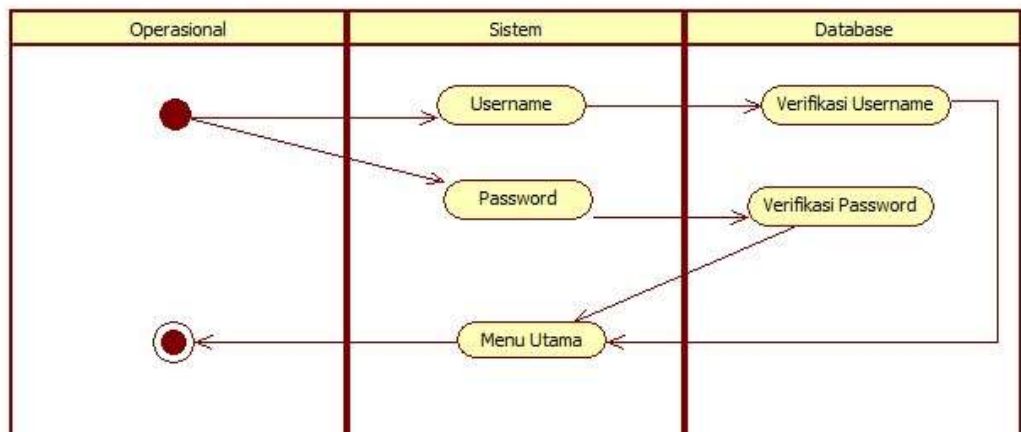
Gambar 5. 50 Communication Diagram Kelola Kegiatan Proyek (UC8)

Pada *Communication diagram* ini menjelaskan proses kelola kegiatan proyek. Aktor melakukan *login* terlebih dahulu. Selanjutnya sistem akan memuat *form* menu utama. Untuk memulai kelola ini, masuk ke menu kelola kegiatan proyek, kemudian sistem akan melakukan proses koneksi *database* dan dilanjutkan dengan proses *OpenTable* ke tabel yang dituju yaitu *tbl_kegiatan* dengan melakukan operasi *update()*. Selanjutnya sistem akan memvalidasi hasil dari operasi, apakah berhasil atau gagal. Dan yang terakhir sistem akan melakukan proses *CloseDB* untuk menutup koneksi.

5.2.5 Activity Diagram

Activity Diagram menggambarkan berbagai alir aktivitas alam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir [33].

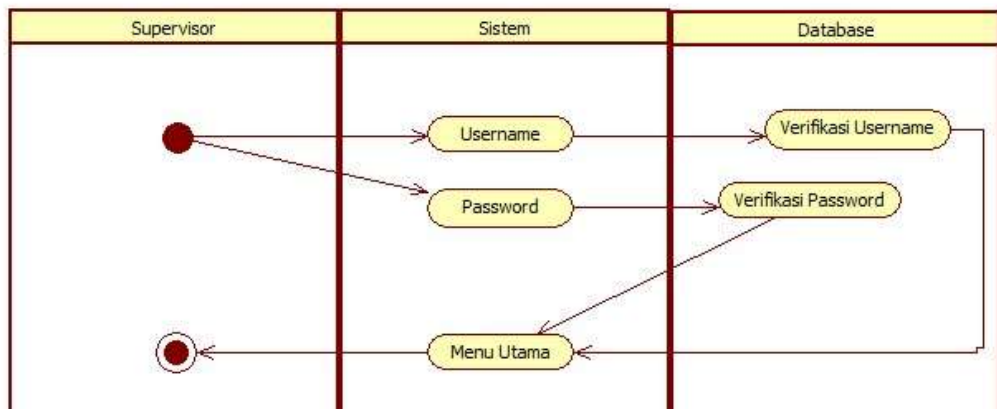
5.2.5.1 Activity Diagram Login Operasional (UC1)



Gambar 5. 26 Activity Diagram Login Operasional (UC1)

Pada *activity diagram* ini menjelaskan proses *login* Operasional. Aktor mulai dengan memasukkan *username* dan *password*, kemudian sistem akan memvalidasi apakah *username* dan *password* yang dimasukkan *valid* atau tidak *valid*. Jika *valid* maka akan tampil menu utama, tetapi jika tidak *valid* akan kembali ke *form login*.

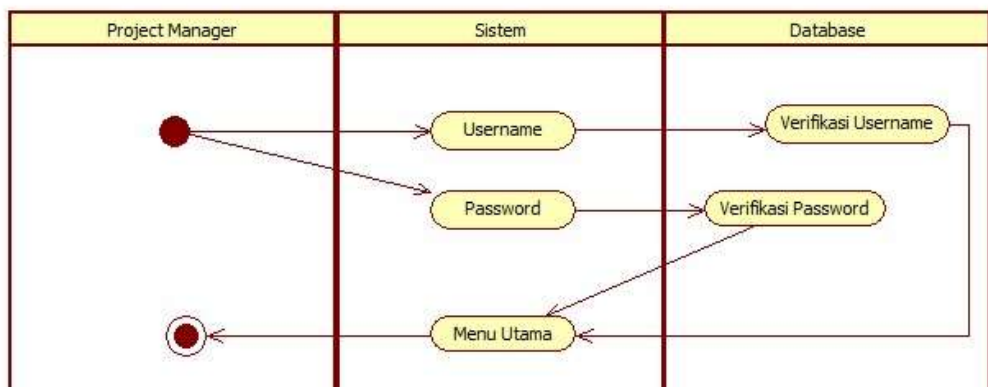
5.2.5.2 Activity Diagram Login Supervisor (UC2)



Gambar 5. 27 Activity Diagram Login Supervisor (UC2)

Pada *activity diagram* ini menjelaskan proses *login* Supervisor. Aktor mulai dengan memasukkan *username* dan *password*, kemudian sistem akan memvalidasi apakah *username* dan *password* yang dimasukkan *valid* atau tidak *valid*. Jika *valid* maka akan tampil menu utama, tetapi jika tidak *valid* akan kembali ke *form login*.

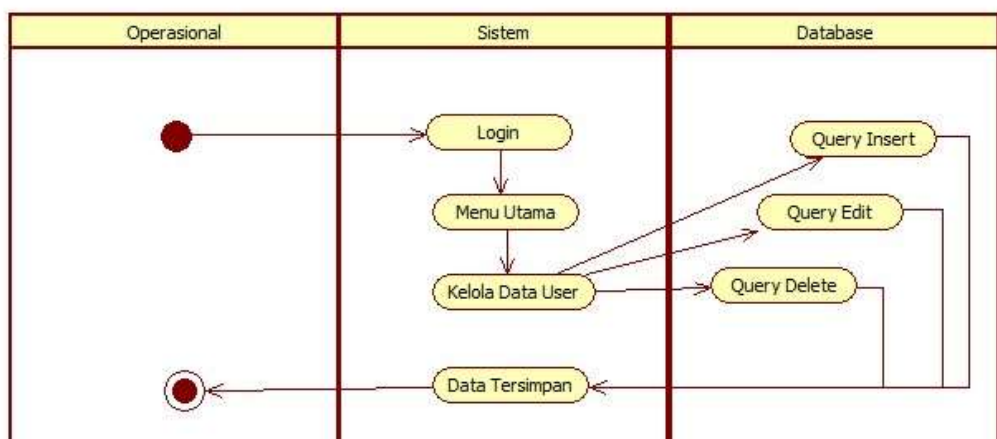
5.2.5.3 Activity Diagram Login Project Manager (UC3)



Gambar 5. 28 Activity Diagram Project Manager (UC3)

Pada *activity diagram* ini menjelaskan proses *login Project Manager*. Aktor mulai dengan memasukkan *username* dan *password*, kemudian sistem akan memvalidasi apakah *username* dan *password* yang dimasukkan *valid* atau tidak *valid*. Jika *valid* maka akan tampil menu utama, tetapi jika tidak *valid* akan kembali ke *form login*.

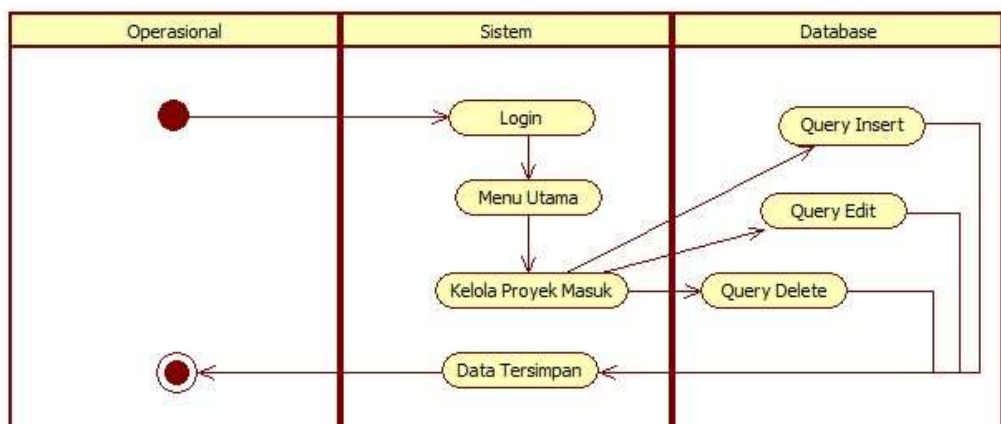
5.2.5.4 Activity Diagram Kelola Data User (UC4)



Gambar 5. 29 2 Activity Diagram Kelola Data User (UC4)

Setelah *login*, Admin memilih menu *kelola data user*. Kemudian sistem akan menampilkan *kelola data user*. Aktor akan melakukan proses *insert*, *update*, dan *delete* data meja, selanjutnya di sistem akan disimpan data *user*. Jika *valid* maka akan tampil data *user*, tetapi jika tidak *valid* akan kembali ke *kelola data user*. Admin dapat *logout* dan sistem menutup *database*.

5.2.5.5 Activity Diagram Kelola Proyek Masuk (UC5)

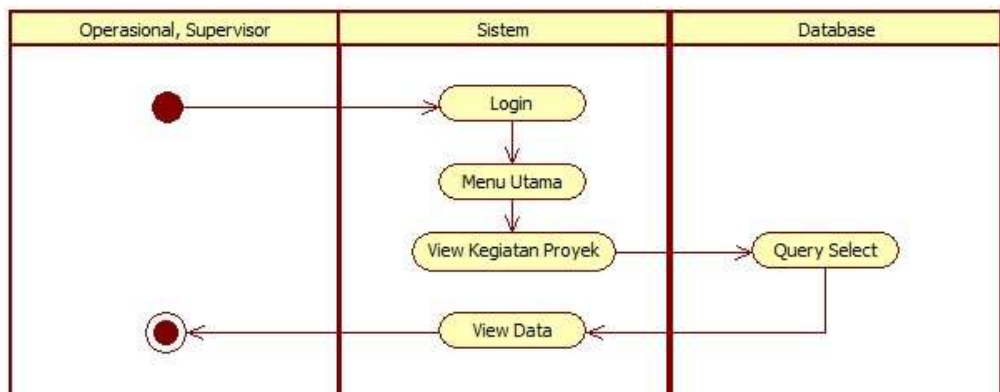


Gambar 5. 30 Activity Diagram Kelola Data Proyek Masuk (UC5)

Setelah *login*, Admin memilih menu kelola Proyek Masuk. Kemudian sistem akan menampilkan kelola data Proyek Masuk. Aktor akan melakukan proses *insert*, *update*, dan *delete* data Proyek Masuk, selanjutnya di sistem

akan disimpan data Proyek Masuk. Jika *valid* maka akan tampil data Proyek Masuk, tetapi jika tidak *valid* akan kembali ke kelola data Proyek Masuk. Admin dapat *logout* dan sistem menutup *database*.

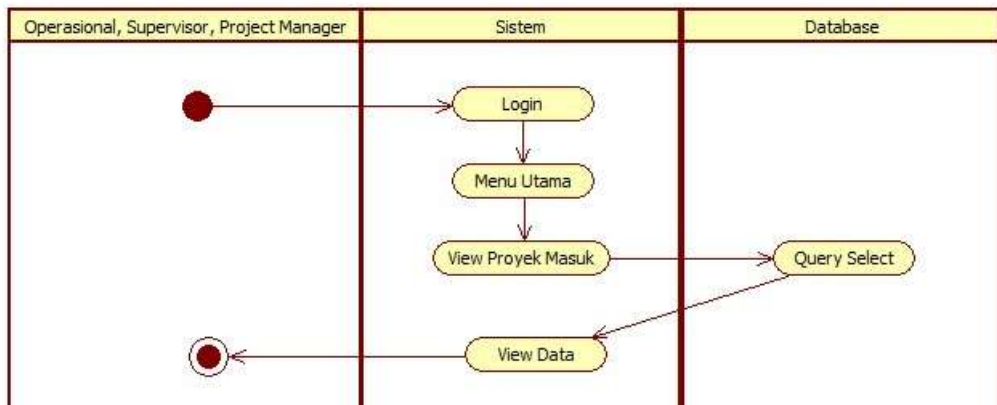
5.2.5.6 Activity Diagram View Kegiatan Proyek (UC6)



Gambar 5. 31 Activity Diagram View Kegiatan Proyek (UC6)

Setelah *login*, Operasional dan Supervisor memilih menu kegiatan proyek. Kemudian sistem akan menampilkan data kegiatan proyek. Aktor akan melakukan proses *view data* kegiatan proyek, selanjutnya di sistem akan ditampilkan.

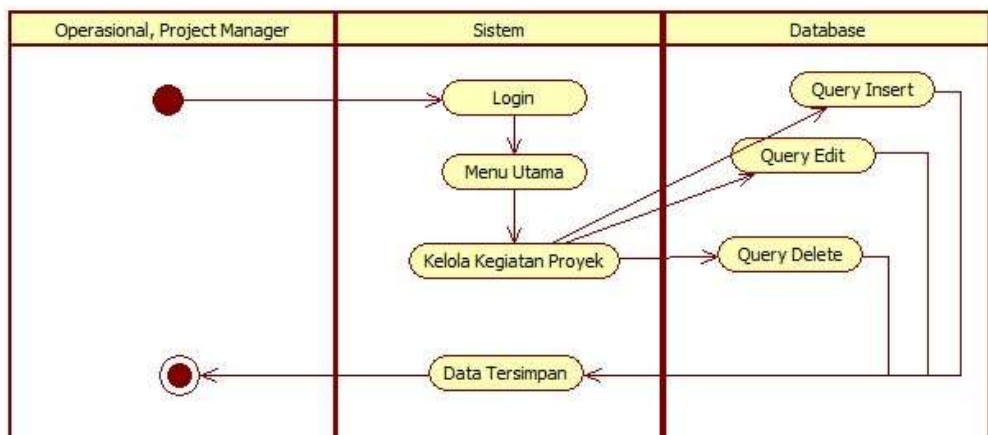
5.2.5.7 Activity Diagram View Proyek Masuk (UC7)



Gambar 5. 33 Activity Diagram View Proyek Masuk (UC7)

Setelah *login*, Operasional dan *Project Manager* memilih menu proyek masuk di. Kemudian sistem akan menampilkan data proyek masuk. Aktor akan melakukan proses *view data* proyek masuk.

5.2.5.8 Activity Diagram Kelola Kegiatan Proyek (UC8)

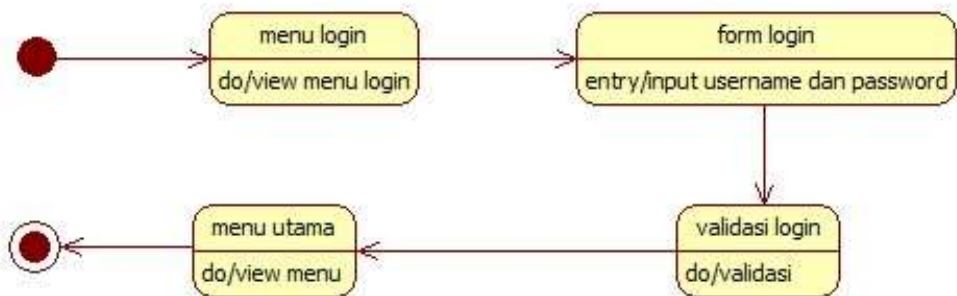


Gambar 5. 33 Activity Diagram Kelola Kegiatan Proyek (UC8)

Setelah *login*, Operasional dan *Project Manager* memilih menu kelola Kegiatan Proyek. Kemudian sistem akan menampilkan kelola Kegiatan Proyek. Aktor akan melakukan proses *insert*, *update*, dan *delete* data Kegiatan Proyek, selanjutnya di sistem akan disimpan data Kegiatan Proyek. Jika *valid* maka akan tampil data Kegiatan Proyek, tetapi jika tidak *valid* akan kembali ke kelola data Kegiatan Proyek. Aktor dapat *logout* dan sistem menutup *database*.

5.2.6 Statechart Diagram

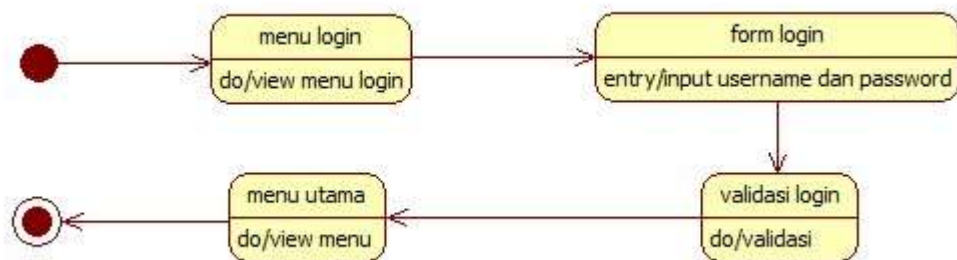
5.2.6.1 Statechart Diagram Login Operasional (UC1)



Gambar 5. 51 Statechart Diagram Login Operasional (UC1)

Operasional masuk ke menu *login* menginputkan *username* dan *password*. Jika *valid* maka akan masuk ke menu Operasional dan jika tidak *valid* akan masuk kembali ke *form login*.

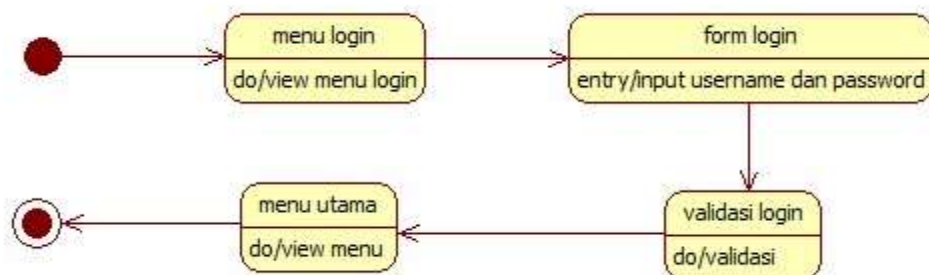
5.2.6.2 Statechart Diagram Login Supervisor (UC2)



Gambar 5. 52 Statechart Diagram Login Supervisor (UC2)

Supervisor masuk ke menu *login* menginputkan *username* dan *password*. Jika *valid* maka akan masuk ke menu Supervisor dan jika tidak *valid* akan masuk kembali ke *form login*.

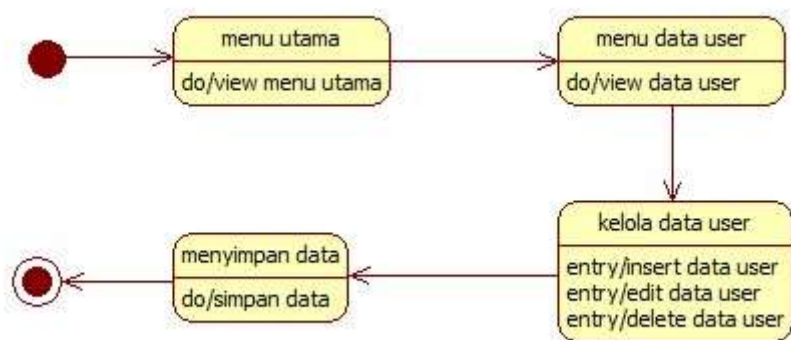
5.2.6.3 Statechart Diagram Login Project Manager (UC3)



Gambar 5. 53 Statechart Diagram Project Manager (UC3)

Project Manager masuk ke menu *login* menginputkan *username* dan *password*. Jika *valid* maka akan masuk ke menu *Project Manager* dan jika tidak *valid* akan masuk kembali ke *form login*.

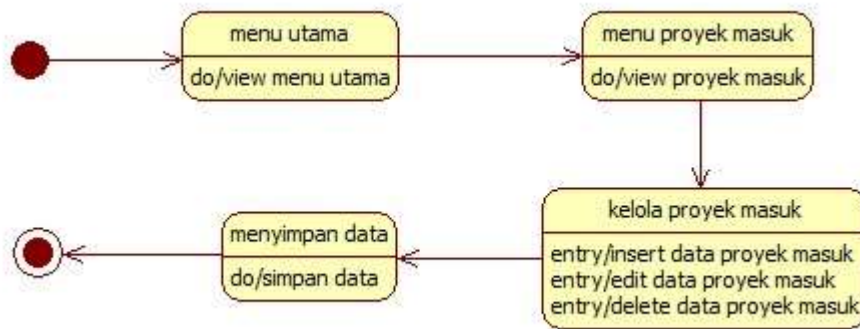
5.2.6.4 Statechart Diagram Kelola Data User (UC4)



Gambar 5. 54 Statechart Diagram Kelola Data User (UC4)

Operasional mengelola menu data user yang nanti di validasi ke *database* jika benar maka *query* berhasil.

5.2.6.5 Statechart Diagram Kelola Proyek Masuk (UC5)



Gambar 5. 55 Statechart Diagram Kelola Data Proyek Masuk (UC5)

Operasional mengelola menu data proyek masuk yang nanti di validasi ke *database* jika benar maka *query* berhasil.

5.2.6.6 Statechart Diagram View Kegiatan Proyek (UC6)



Gambar 5. 56 Statechart Diagram View Kegiatan Proyek (UC6)

Operasional dan Supervisor melihat kegiatan proyek yang sebelumnya sudah tersimpan di *database*.

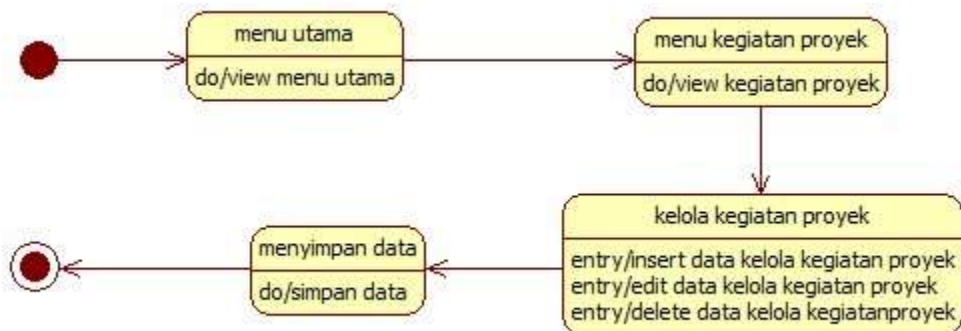
5.2.6.7 Statechart Diagram View Proyek Masuk (UC7)



Gambar 5. 57 Statechart Diagram View Proyek Masuk (UC7)

Operasional, Supervisor, dan *Project Manager* melihat proyek masuk yang sebelumnya sudah tersimpan di *database*.

5.2.6.8 Statechart Diagram Kelola Kegiatan Proyek (UC8)

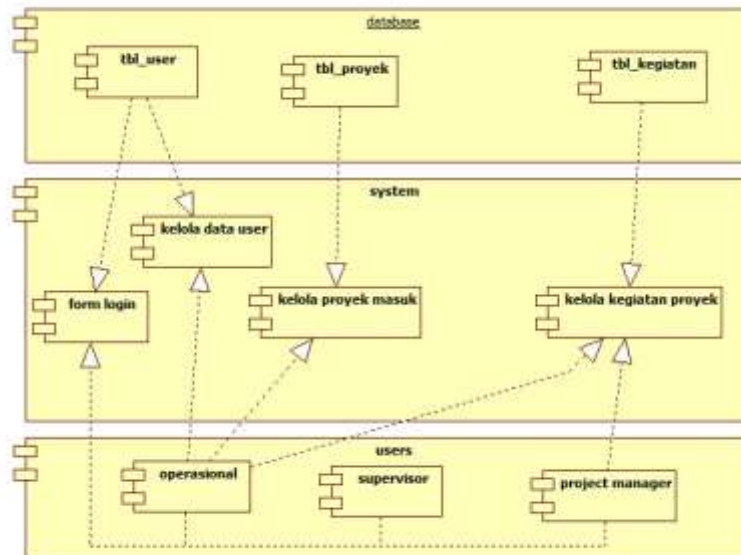


Gambar 5. 58 Statechart Diagram Kelola Kegiatan Proyek (UC8)

Operasional dan *Project Manager* mengelola menu kegiatan proyek yang nanti di validasi ke *database* jika benar maka *query* berhasil.

5.2.7 Component Diagram

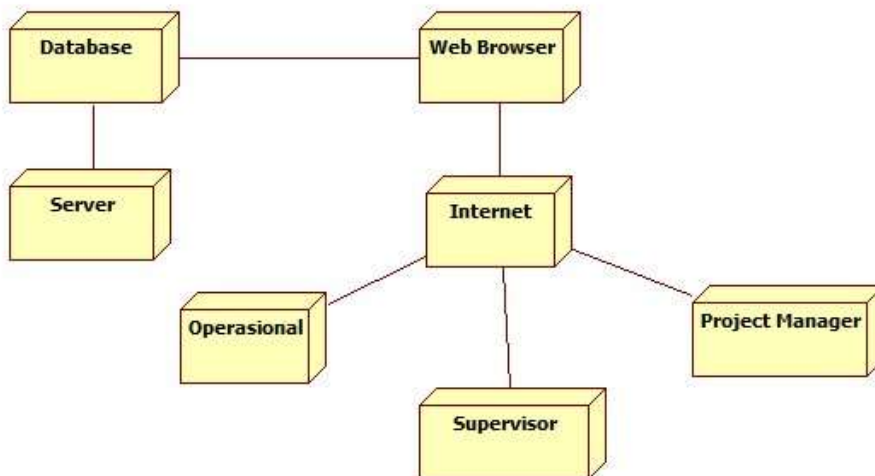
Hal penting pada *component* adalah *component* mewakili potongan-potongan yang independen yang bisa dipesan dan diperbaharui sewaktu-waktu. *Component Diagram* adalah sebuah bentuk pemodelan sebuah sistem kedalam komponen-komponen yang lebih rendah.



Gambar 5. 59 Component Diagram

5.2.8 Deployment Diagram

Deployment diagram menunjukkan tata letak sebuah sistem secara fisik, menampakkan bagian-bagian *software* yang berjalan pada bagian *hardware*. Bagian utama *hardware* atau perangkat keras adalah node yaitu nama umum untuk semua jenis sumber komputasi.



Gambar 5. 60 Deployment Diagram

6.1 Pengkajian

Dalam pengkajian ini, penulis akan mengkaji kembali data yang diperoleh dari hasil pengumpulan data yang telah dilakukan sebelumnya untuk mendukung analisis dari penelitian ini, untuk memberikan solusi. Pada Bab IV bagian 4.2 telah dijelaskan bahwa penulis melakukan analisis dengan beberapa tahapan dan metode yang digunakan yaitu *Critical Path Method (CPM)*.

6.2 Identifikasi Masalah

Pada identifikasi masalah yang telah dipaparkan sebelumnya yaitu untuk menentukan total durasi pengerjaan proyek dengan menggunakan *critical path method* untuk membantu mempermudah *Project Manager* dalam melakukan perencanaan dengan mendapatkan data *Project Completion*. Untuk data dan hasil dari pengolahan data tersebut dapat dilihat pada table dibawah ini :

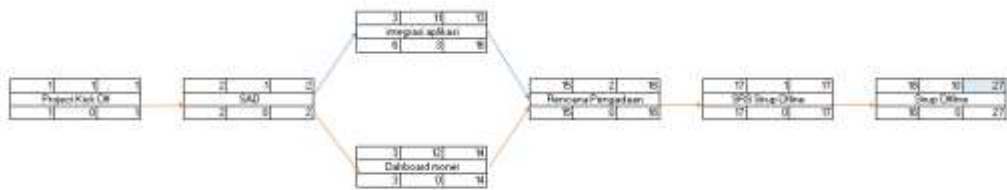
Tabel 6. 2 Data Project Completion LKPP-Sirup

Modul	Estimated Man Days	Predecessor
GENERAL	1	-
PROJECT KICK OFF		
SAD	1	Project Kick Off
INTEGRASI APLIKASI	11	SAD
Web Service SIMDA		
Create API untuk insert ke Database		
Update data RKA dan buatkan menjadi history		
display data RKA yang diupload dalam bentuk Tab		
Web Service SIMRAL		
Create API untuk insert ke Database		
Update Data RKA dan insert ke table History		
Fixing WS Om SPAN		
Integrasi KRISNA		
Update Data SIMDA menggunakan API		
Update Data SIMRAL menggunakan API		
Create paket menggunakan API		
DASHBOARD MONER	12	SAD
Rekap paket sesuai tagging krisna		
Optimasi query fitur statistic		
Bug Fixing OLAP		
Dashboard notifikasi tahun aktif satker /OPD		
Dashboard notifikasi perubahan pagu PKOR/PKOK		
Status upload file RKAD		
Rekap pengumuman RUP		
Notif perubahan paket		
Notif verifikasi akun PPK		

Rekap pengumuman RUP di PPK yang sama		
Rekap pengumuman RUP seluruh KLPD		
Model Rekapitulasi		
RENCANA PENGADAAN	2	SAD
Notifikasi Perubahan struktur anggaran		
Create paket dengan lintas anggaran kegiatan		
SRS SIRUP OFFLINE	1	Integrasi, Dashboard Moner, Rencana Pengadaan
SIRUP OFFLINE	10	SRS Sirup Offline
Sirup Installer		
Create Paket		
Download daftar paket		
Upload daftar paket		
Auto Upload daftar paket		
TOTAL MAIN DAYS	41	

6.3 Pengujian Metode

Pada tahap pengujian metode ini, dilakukan pengujian metode untuk mengetahui apakah *Critical Path Method* dapat menghasilkan total durasi pengerjaan proyek yang lebih optimal untuk membantu *Project Manajer* dalam melakukan perencanaan.



Gambar 6. 2 Perhitungan Critical Path Method

6.4 Evaluasi Metode

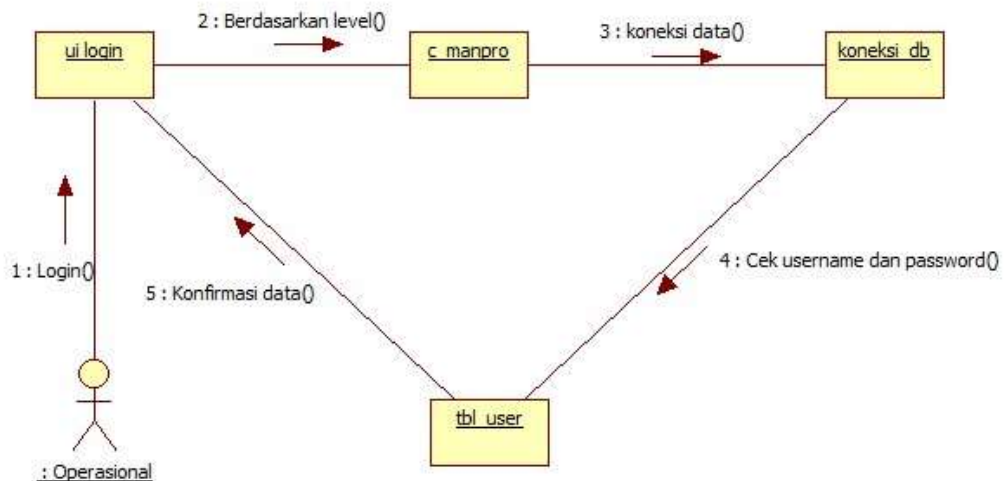
Pada data yang diperoleh hasil total waktu pengerjaan pada proyek LKPP-Sirup adalah 41 hari, sedangkan ketika diolah menggunakan Critical Path Method menghasilkan durasi waktu yang lebih optimal yaitu 27 hari.

e diagram ini menjelaskan proses kelola kegiatan proyek. Aktor melakukan *login* terlebih dahulu. Selanjutnya sistem akan memuat *form* menu utama. Untuk memulai kelola ini, masuk ke menu kelola kegiatan proyek, kemudian sistem akan melakukan proses koneksi *database* dan dilanjutkan dengan proses *OpenTable* ke tabel yang dituju yaitu *tbl_kegiatan* dengan melakukan operasi *update()*. Selanjutnya sistem akan memvalidasi hasil dari operasi, apakah berhasil atau gagal. Dan yang terakhir sistem akan melakukan proses *CloseDB* untuk menutup koneksi.

5.2.4 Communication Diagram

Communication diagram adalah sebuah interaksi yang memberi tekanan pada hubungan-hubungan data antar partisipan yang berbeda dalam sebuah interaksi.

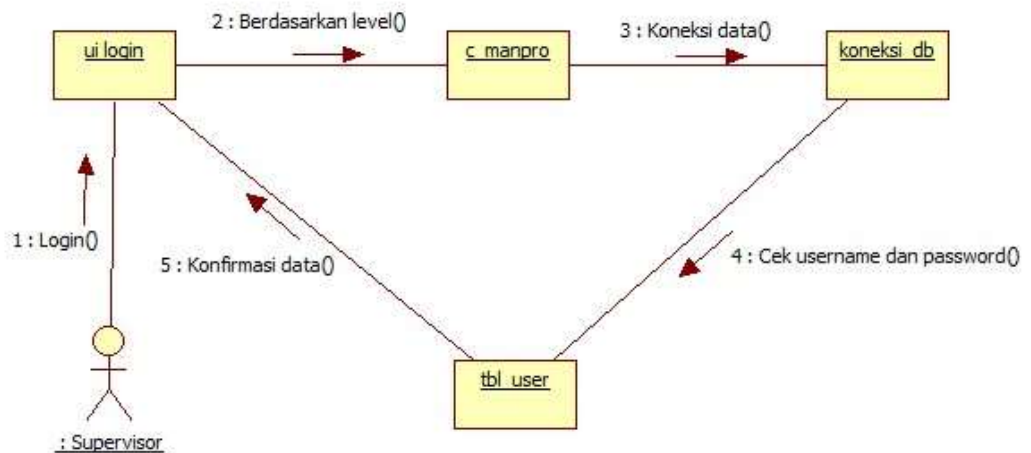
5.2.3.1 Communication Diagram Login Operasional (UC1)



Gambar 5. 61 Communication Diagram Login Operasional (UC1)

Pada *Communication diagram* ini menjelaskan proses *login*. Aktor mulai menjalankan aplikasi dan akan tampil *form login*. Kemudian aktor memasukkan *username dan password*, pada *interface login* data yang di *input* akan di cek di tabel `tbl_user` di *database*. Jika sesuai maka akan mendapatkan tampilan form utama ke pengguna.

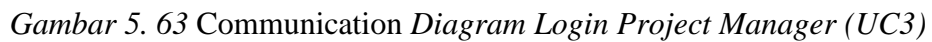
5.2.3.2 Communication Diagram Login Supervisor (UC2)



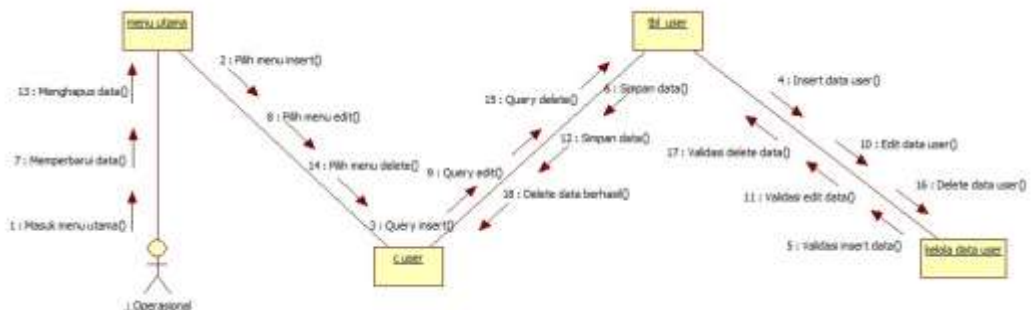
Gambar 5. 62 Communication Diagram Login Supervisor (UC2)

Pada *Communication diagram* ini menjelaskan proses *login*. Aktor mulai menjalankan aplikasi dan akan tampil *form login*. Kemudian aktor memasukkan *username dan password*, pada *interface login* data yang di *input* akan di cek di tabel *tbl_user* di *database*. Jika sesuai maka akan mendapatkan tampilan form utama ke pengguna.

5.2.3.3 Communication Diagram Login Project Manager (UC3)



5.2.3.4 Communication Diagram Kelola Data User (UC4)



Pada *Communication diagram* ini menjelaskan proses kelola data *user*. Aktor melakukan *login* terlebih dahulu. Selanjutnya sistem akan memuat *form* menu utama. Untuk memulai kelola ini, masuk ke menu kelola data *user*, kemudian sistem akan melakukan proses koneksi database dan dilanjutkan dengan proses *OpenTable* ke tabel yang dituju yaitu *tbl_user* dengan melakukan operasi *insert()*, *update()*, *delete()*. Selanjutnya sistem akan memvalidasi hasil dari operasi, apakah berhasil atau gagal. Dan yang terakhir sistem akan melakukan proses *CloseDB* untuk menutup koneksi.

```

graph TD
    Operator[Operator]
    Admin[Admin]
    Customer[Customer]
    Project[Sebelum proyek masuk]
    MainMenu[Menu Utama]
    SubMenu[SubMenu]
    ProjectMenu[Menu proyek masuk]

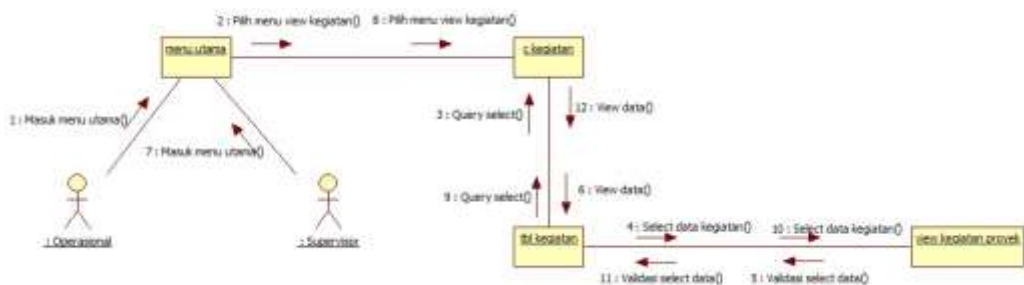
    Operator -- "1 : Masuk menu utama()" --> MainMenu
    MainMenu -- "7 : Menampilkan data()" --> Operator
    Operator -- "12 : Menghapus data()" --> MainMenu
    MainMenu -- "2 : Pilih menu insert()" --> SubMenu
    SubMenu -- "8 : Pilih menu edit()" --> MainMenu
    MainMenu -- "14 : Pilih menu delete()" --> SubMenu
    SubMenu -- "3 : Query insert()" --> Customer
    Customer -- "9 : Query edit()" --> Admin
    Admin -- "15 : Query delete()" --> Project
    Project -- "6 : Simpan data()" --> Admin
    Admin -- "17 : Validasi delete data()" --> Project
    Project -- "4 : Insert data proyek masuk()" --> ProjectMenu
    ProjectMenu -- "20 : Edit data proyek masuk()" --> ProjectMenu
    ProjectMenu -- "11 : Validasi edit data()" --> Admin
    Admin -- "18 : Delete data berhasil()" --> Customer
    ProjectMenu -- "16 : Delete data proyek masuk()" --> ProjectMenu
    ProjectMenu -- "5 : Validasi insert data()" --> Admin
  
```

The diagram illustrates the interactions between an Operator, Admin, and Customer with the SIM system. The Operator interacts with the Main Menu (Menu Utama) for login, data display, deletion, and menu selection. The Admin interacts with the Sub Menu (SubMenu) for data insertion, editing, and deletion, and with the Project Menu (Menu proyek masuk) for project data management. The Customer interacts with the Main Menu for data insertion and deletion, and with the Sub Menu for data editing and deletion. The Project Menu handles project-specific data operations like insertion, editing, and deletion, with validation steps involving the Admin.

Pada *Communication diagram* ini menjelaskan proses kelola proyek masuk. Aktor melakukan *login* terlebih dahulu. Selanjutnya sistem akan memuat *form* menu utama. Untuk memulai kelola ini, masuk ke menu proyek

masuk kemudian sistem akan melakukan proses koneksi *database* dan dilanjutkan dengan proses *OpenTable* ke tabel yang dituju yaitu *tbl_proyek* dengan melakukan operasi *insert()*, *update()*, *delete()*. Selanjutnya sistem akan memvalidasi hasil dari operasi, apakah berhasil atau gagal. Dan yang terakhir sistem akan melakukan proses *CloseDB* untuk menutup koneksi.

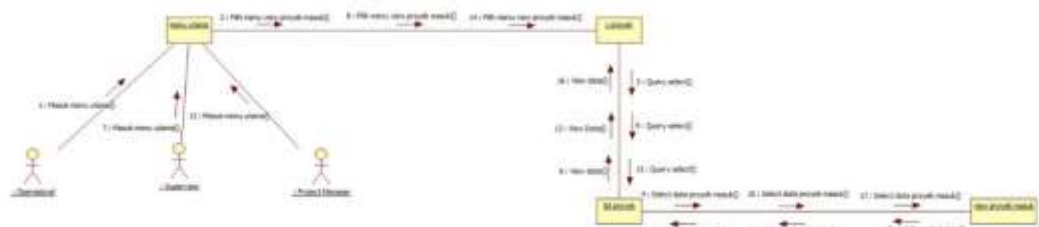
5.2.3.6 Communication Diagram View Kegiatan Proyek (UC6)



Gambar 5. 66 Communication Diagram View Kegiatan Proyek (UC6)

Pada *Communication diagram* ini menjelaskan proses *view data* kegiatan proyek. Aktor melakukan *login* terlebih dahulu. Selanjutnya sistem akan memuat *form* menu utama. Untuk memulai proses ini, masuk ke menu data kegiatan, kemudian sistem akan melakukan proses koneksi *database* dan dilanjutkan dengan proses *OpenTable* ke tabel yang dituju yaitu *tbl_kegiatan* dengan melakukan operasi *select()*. Selanjutnya sistem akan memvalidasi hasil dari operasi, apakah berhasil atau gagal. Dan yang terakhir sistem akan melakukan proses *CloseDB* untuk menutup koneksi.

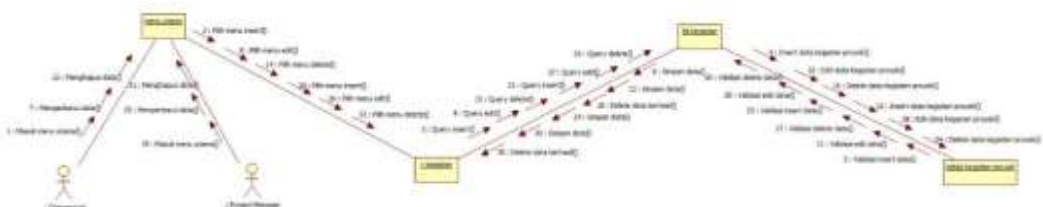
5.2.3.7 Communication Diagram View Proyek Masuk (UC7)



Gambar 5. 67 Communication Diagram View Proyek Masuk (UC7)

Pada *Communication diagram* ini menjelaskan proses *view data* proyek masuk. Aktor melakukan *login* terlebih dahulu. Selanjutnya sistem akan memuat *form* menu utama. Untuk memulai proses ini, masuk ke menu data proyek, kemudian sistem akan melakukan proses koneksi *database* dan dilanjutkan dengan proses *OpenTable* ke tabel yang dituju yaitu *tbl_proyek* dengan melakukan operasi *select()*. Selanjutnya sistem akan memvalidasi hasil dari operasi, apakah berhasil atau gagal. Dan yang terakhir sistem akan melakukan proses *CloseDB* untuk menutup koneksi.

5.2.3.8 Communication Diagram Kelola Kegiatan Proyek (UC8)



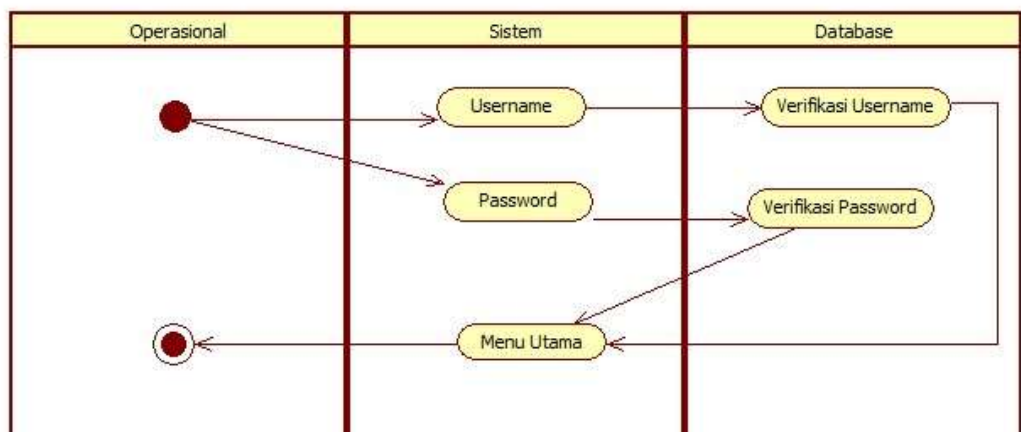
Gambar 5. 68 Communication Diagram Kelola Kegiatan Proyek (UC8)

Pada *Communication diagram* ini menjelaskan proses kelola kegiatan proyek. Aktor melakukan *login* terlebih dahulu. Selanjutnya sistem akan memuat *form* menu utama. Untuk memulai kelola ini, masuk ke menu kelola kegiatan proyek, kemudian sistem akan melakukan proses koneksi *database* dan dilanjutkan dengan proses *OpenTable* ke tabel yang dituju yaitu *tbl_kegiatan* dengan melakukan operasi *update()*. Selanjutnya sistem akan memvalidasi hasil dari operasi, apakah berhasil atau gagal. Dan yang terakhir sistem akan melakukan proses *CloseDB* untuk menutup koneksi.

5.2.5 Activity Diagram

Activity Diagram menggambarkan berbagai alir aktivitas alam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir [33].

5.2.5.1 Activity Diagram Login Operasional (UC1)

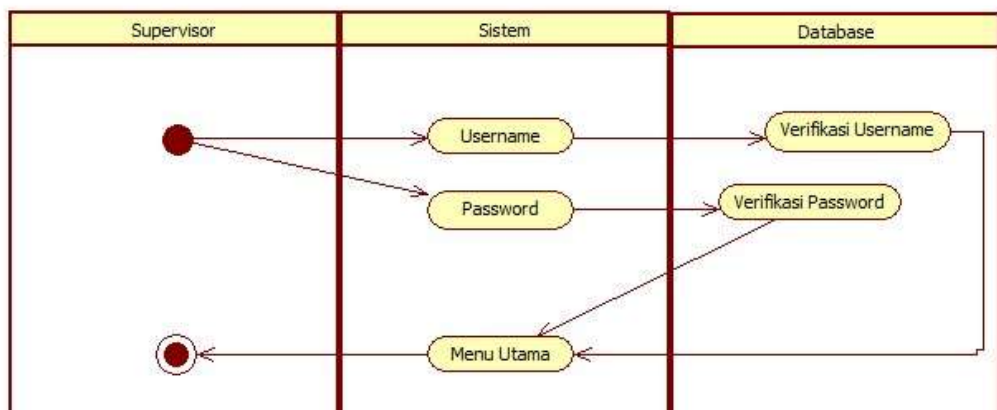


Gambar 5. 26 Activity Diagram Login Operasional (UC1)

Pada *activity diagram* ini menjelaskan proses *login* Operasional. Aktor mulai dengan memasukkan *username* dan *password*, kemudian sistem akan

memvalidasi apakah *username* dan *password* yang dimasukkan *valid* atau tidak *valid*. Jika *valid* maka akan tampil menu utama, tetapi jika tidak *valid* akan kembali ke *form login*.

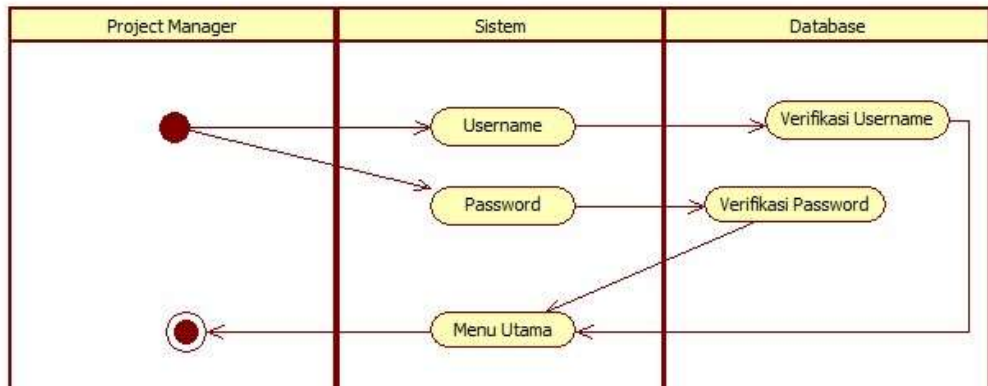
5.2.5.2 Activity Diagram Login Supervisor (UC2)



Gambar 5. 27 Activity Diagram Login Supervisor (UC2)

Pada *activity diagram* ini menjelaskan proses *login* Supervisor. Aktor mulai dengan memasukkan *username* dan *password*, kemudian sistem akan memvalidasi apakah *username* dan *password* yang dimasukkan *valid* atau tidak *valid*. Jika *valid* maka akan tampil menu utama, tetapi jika tidak *valid* akan kembali ke *form login*.

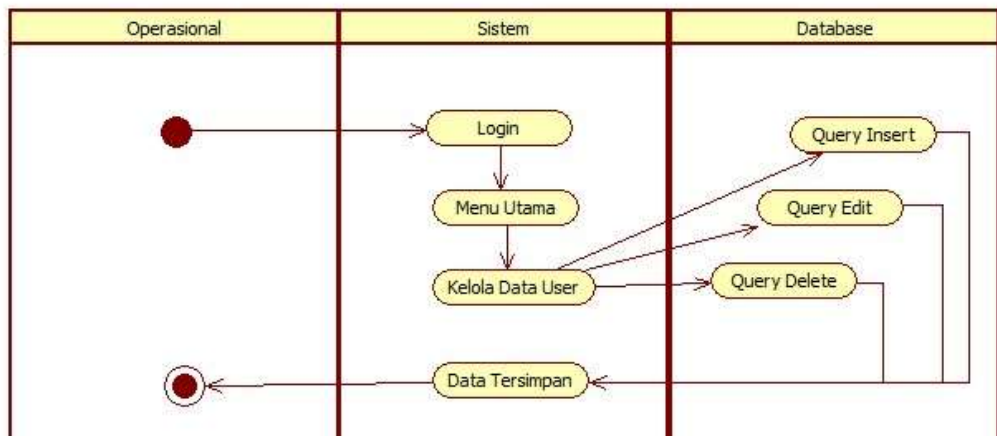
5.2.5.3 Activity Diagram Login Project Manager (UC3)



Gambar 5. 28 Activity Diagram Project Manager (UC3)

Pada *activity diagram* ini menjelaskan proses *login Project Manager*. Aktor mulai dengan memasukkan *username* dan *password*, kemudian sistem akan memvalidasi apakah *username* dan *password* yang dimasukkan *valid* atau tidak *valid*. Jika *valid* maka akan tampil menu utama, tetapi jika tidak *valid* akan kembali ke *form login*.

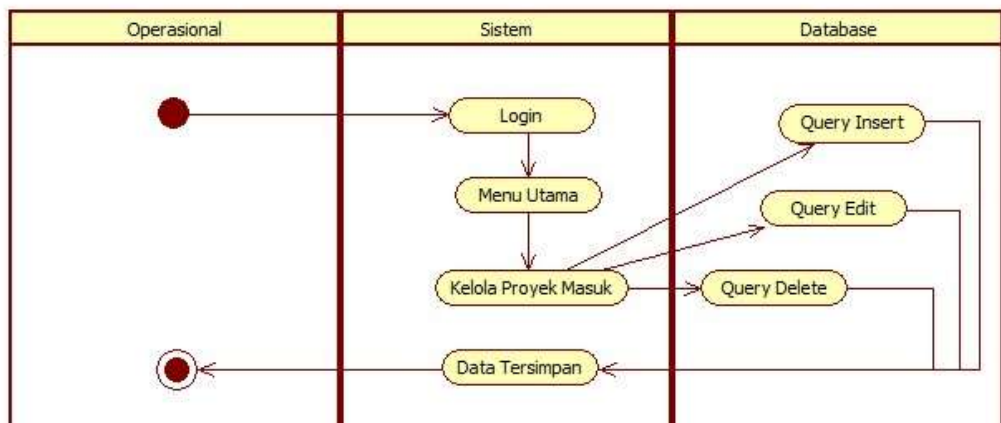
5.2.5.4 Activity Diagram Kelola Data User (UC4)



Gambar 5. 29 3 Activity Diagram Kelola Data User (UC4)

Setelah *login*, Admin memilih menu kelola data *user*. Kemudian sistem akan menampilkan kelola data *user*. Aktor akan melakukan proses *insert*, *update*, dan *delete* data meja, selanjutnya di sistem akan disimpan data *user*. Jika *valid* maka akan tampil data *user*, tetapi jika tidak *valid* akan kembali ke kelola data *user*. Admin dapat *logout* dan sistem menutup *database*.

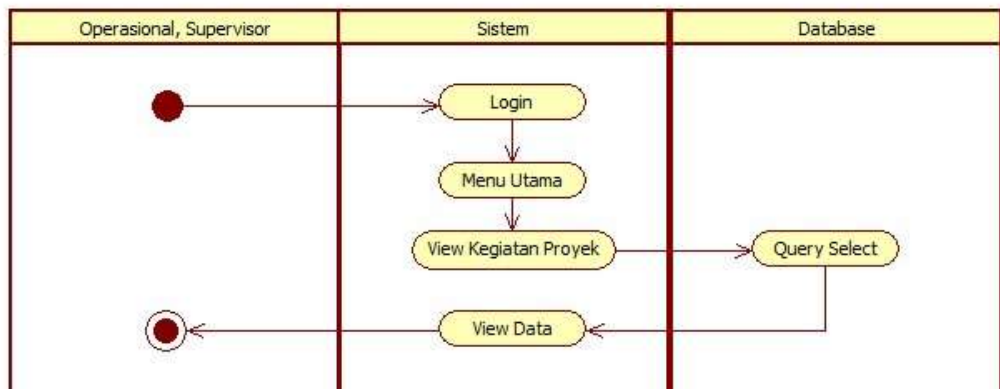
5.2.5.5 Activity Diagram Kelola Proyek Masuk (UC5)



Gambar 5. 30 Activity Diagram Kelola Data Proyek Masuk (UC5)

Setelah *login*, Admin memilih menu kelola Proyek Masuk. Kemudian sistem akan menampilkan kelola data Proyek Masuk. Aktor akan melakukan proses *insert*, *update*, dan *delete* data Proyek Masuk, selanjutnya di sistem akan disimpan data Proyek Masuk. Jika *valid* maka akan tampil data Proyek Masuk, tetapi jika tidak *valid* akan kembali ke kelola data Proyek Masuk. Admin dapat *logout* dan sistem menutup *database*.

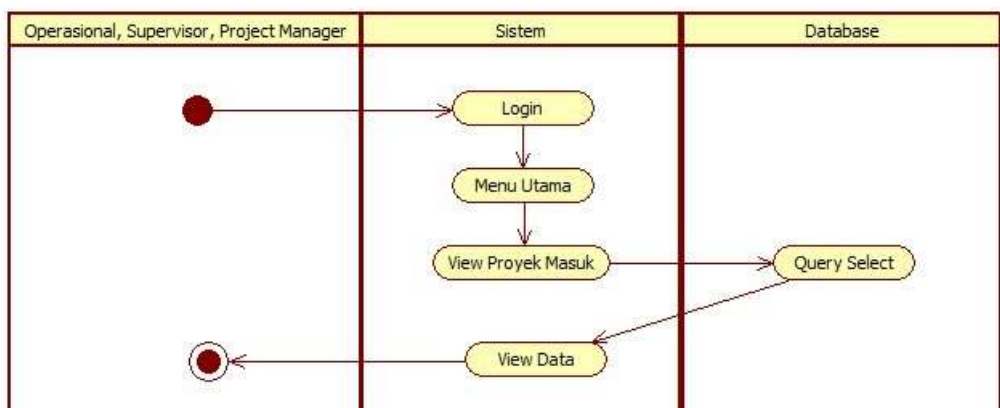
5.2.5.6 Activity Diagram View Kegiatan Proyek (UC6)



Gambar 5. 31 Activity Diagram View Kegiatan Proyek (UC6)

Setelah *login*, Operasional dan Supervisor memilih menu kegiatan proyek. Kemudian sistem akan menampilkan data kegiatan proyek. Aktor akan melakukan proses *view data* kegiatan proyek, selanjutnya di sistem akan ditampilkan.

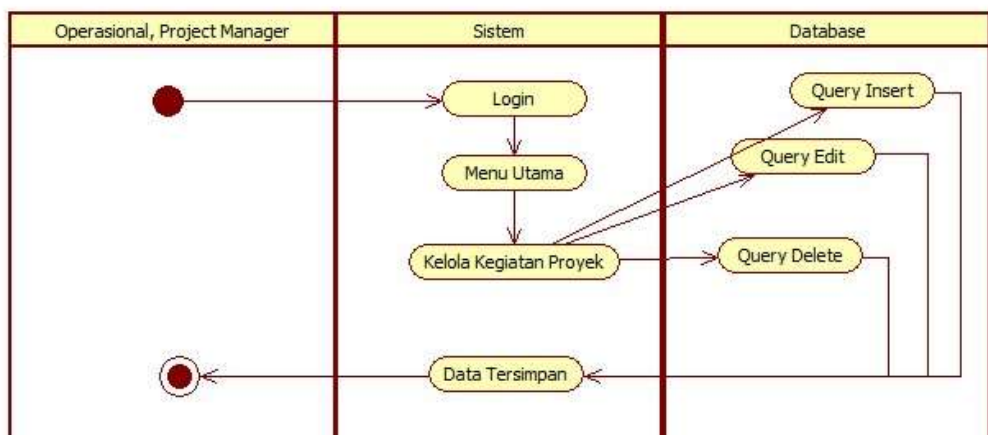
5.2.5.7 Activity Diagram View Proyek Masuk (UC7)



Gambar 5. 33 Activity Diagram View Proyek Masuk (UC7)

Setelah *login*, Operasional dan *Project Manager* memilih menu proyek masuk di. Kemudian sistem akan menampilkan data proyek masuk. Aktor akan melakukan proses *view* data proyek masuk.

5.2.5.8 Activity Diagram Kelola Kegiatan Proyek (UC8)

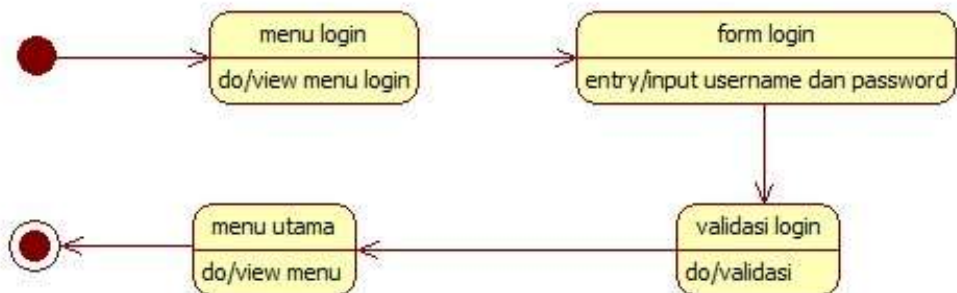


Gambar 5. 33 Activity Diagram Kelola Kegiatan Proyek (UC8)

Setelah *login*, Operasional dan *Project Manager* memilih menu kelola Kegiatan Proyek. Kemudian sistem akan menampilkan kelola Kegiatan Proyek. Aktor akan melakukan proses *insert*, *update*, dan *delete* data Kegiatan Proyek, selanjutnya di sistem akan disimpan data Kegiatan Proyek. Jika *valid* maka akan tampil data Kegiatan Proyek, tetapi jika tidak *valid* akan kembali ke kelola data Kegiatan Proyek. Aktor dapat *logout* dan sistem menutup *database*.

5.2.6 Statechart Diagram

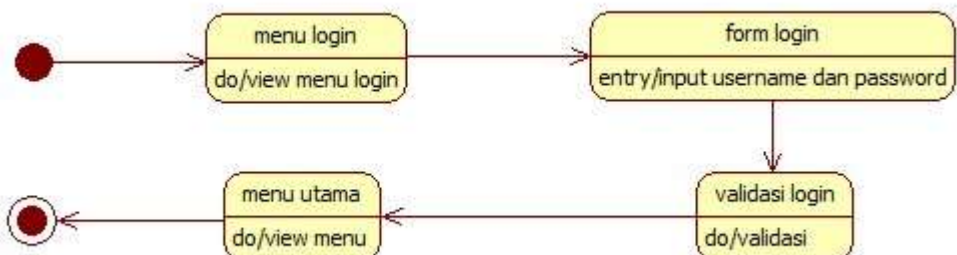
5.2.6.1 Statechart Diagram Login Operasional (UC1)



Gambar 5. 69 Statechart Diagram Login Operasional (UC1)

Operasional masuk ke menu *login* menginputkan *username* dan *password*. Jika *valid* maka akan masuk ke menu Operasional dan jika tidak *valid* akan masuk kembali ke *form login*.

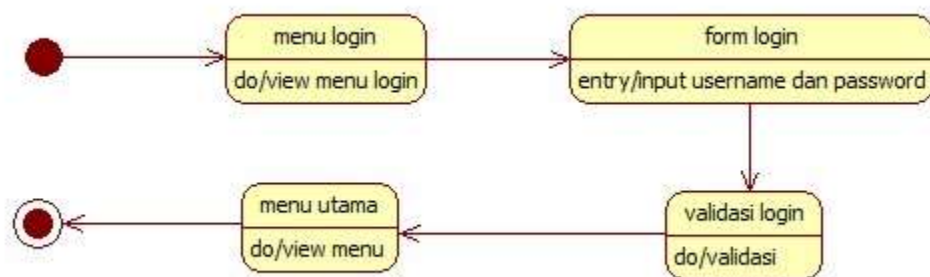
5.2.6.2 Statechart Diagram Login Supervisor (UC2)



Gambar 5. 70 Statechart Diagram Login Supervisor (UC2)

Supervisor masuk ke menu *login* menginputkan *username* dan *password*. Jika *valid* maka akan masuk ke menu Supervisor dan jika tidak *valid* akan masuk kembali ke *form login*.

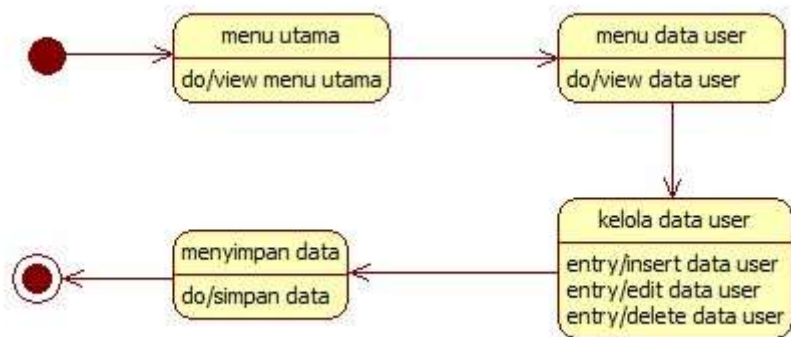
5.2.6.3 Statechart Diagram Login Project Manager (UC3)



Gambar 5. 71 Statechart Diagram Project Manager (UC3)

Project Manager masuk ke menu *login* menginputkan *username* dan *password*. Jika *valid* maka akan masuk ke menu *Project Manager* dan jika tidak *valid* akan masuk kembali ke *form login*.

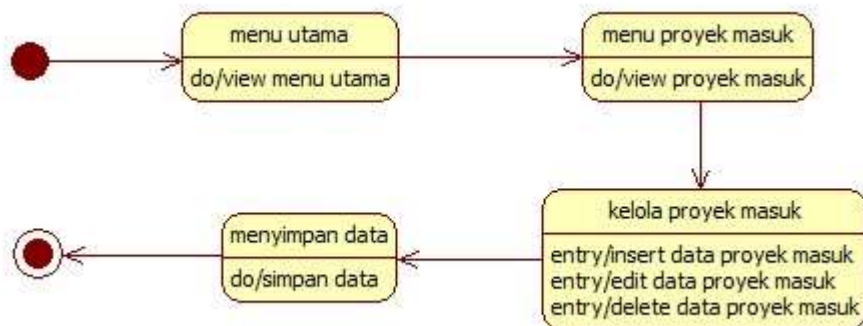
5.2.6.4 Statechart Diagram Kelola Data User (UC4)



Gambar 5. 72 Statechart Diagram Kelola Data User (UC4)

Operasional mengelola menu data user yang nanti di validasi ke *database* jika benar maka *query* berhasil.

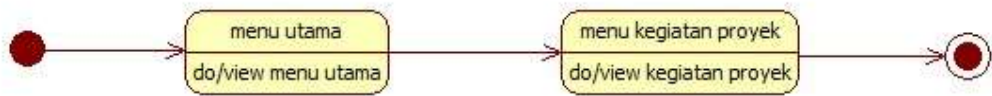
5.2.6.5 Statechart Diagram Kelola Proyek Masuk (UC5)



Gambar 5. 73 Statechart Diagram Kelola Data Proyek Masuk (UC5)

Operasional mengelola menu data proyek masuk yang nanti di validasi ke *database* jika benar maka *query* berhasil.

5.2.6.6 Statechart Diagram View Kegiatan Proyek (UC6)



Gambar 5. 74 Statechart Diagram View Kegiatan Proyek (UC6)

Operasional dan Supervisor melihat kegiatan proyek yang sebelumnya sudah tersimpan di *database*.

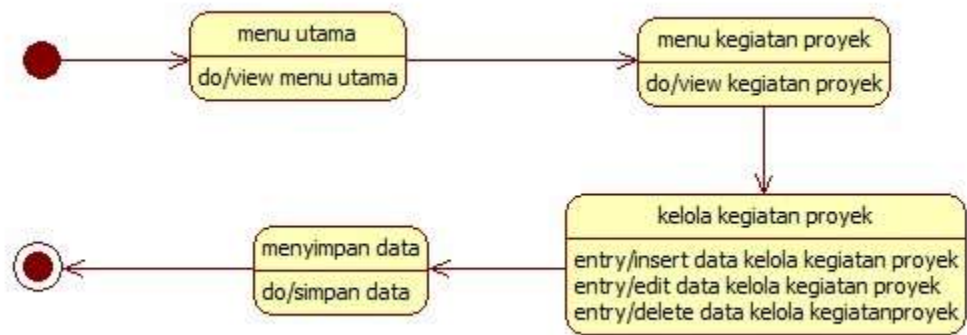
5.2.6.7 Statechart Diagram View Proyek Masuk (UC7)



Gambar 5. 75 Statechart Diagram View Proyek Masuk (UC7)

Operasional, Supervisor, dan *Project Manager* melihat proyek masuk yang sebelumnya sudah tersimpan di *database*.

5.2.6.8 Statechart Diagram Kelola Kegiatan Proyek (UC8)

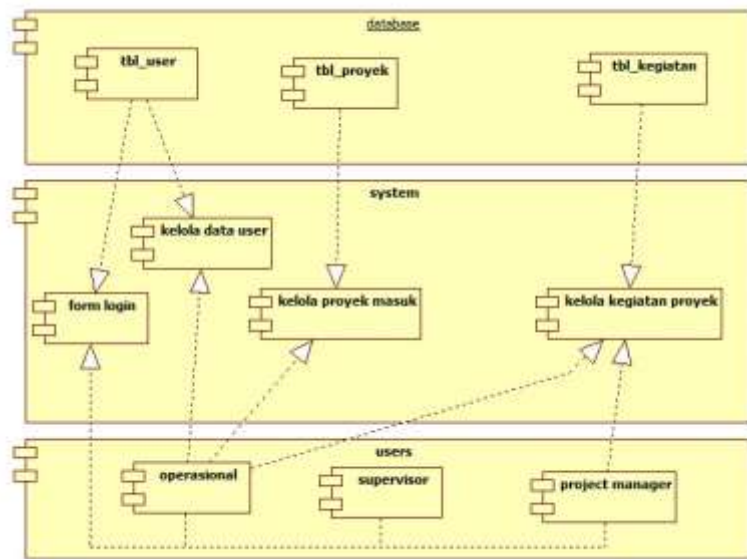


Gambar 5. 76 Statechart Diagram Kelola Kegiatan Proyek (UC8)

Operasional dan *Project Manager* mengelola menu kegiatan proyek yang nanti di validasi ke *database* jika benar maka *query* berhasil.

5.2.7 Component Diagram

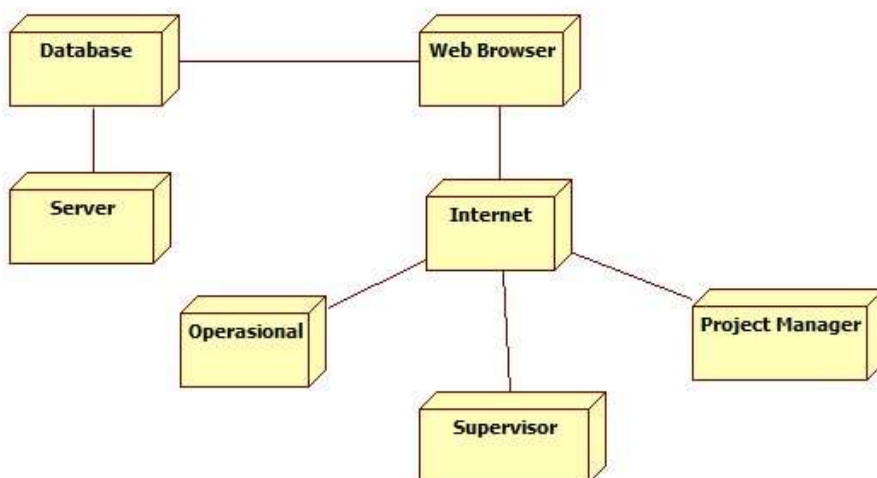
Hal penting pada *component* adalah *component* mewakili potongan-potongan yang independen yang bisa dipesan dan diperbaharui sewaktu-waktu. *Component Diagram* adalah sebuah bentuk pemodelan sebuah sistem kedalam komponen-komponen yang lebih rendah.



Gambar 5. 77 Component Diagram

5.2.8 Deployment Diagram

Deployment diagram menunjukkan tata letak sebuah sistem secara fisik, menampakkan bagian-bagian *software* yang berjalan pada bagian *hardware*. Bagian utama *hardware* atau perangkat keras adalah node yaitu nama umum untuk semua jenis sumber komputasi.



Gambar 5. 78 Deployment Diagram

6.1 Pengkajian

Dalam pengkajian ini, penulis akan mengkaji kembali data yang diperoleh dari hasil pengumpulan data yang telah dilakukan sebelumnya untuk mendukung analisis dari penelitian ini, untuk memberikan solusi. Pada Bab IV bagian 4.2 telah dijelaskan bahwa penulis melakukan analisis dengan beberapa tahapan dan metode yang digunakan yaitu *Critical Path Method (CPM)*.

6.2 Identifikasi Masalah

Pada identifikasi masalah yang telah dipaparkan sebelumnya yaitu untuk menentukan total durasi pengerjaan proyek dengan menggunakan *critical path method* untuk membantu mempermudah *Project Manager* dalam melakukan perencanaan dengan mendapatkan data *Project Completion*. Untuk data dan hasil dari pengolahan data tersebut dapat dilihat pada table dibawah ini :

Tabel 6. 3 Data Project Completion LKPP-Sirup

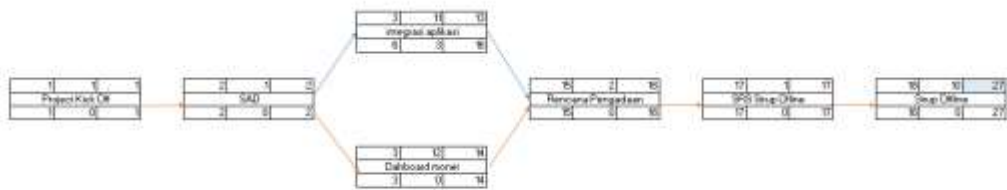
Modul	Estimated Man Days	Predesesor
GENERAL	1	-
PROJECT KICK OFF		
SAD	1	Project Kick Off
INTEGRASI APLIKASI	11	SAD
Web Service SIMDA		
Create API untuk insert ke Database		
Update data RKA dan buatkan menjadi history		
dispaly data RKA yang diupload dalam bentuk Tab		
Web Service SIMRAL		

Create API untuk insert ke Database		
Update Data RKA dan insert ke table History		
Fixing WS Om SPAN		
Integrasi KRISNA		
Update Data SIMDA menggunakan API		
Update Data SIMRAL menggunakan API		
Create paket menggunakan API		
DASHBOARD MONER	12	SAD
Rekap paket sesuai tagging krisna		
Optimasi query fitur statistic		
Bug Fixing OLAP		
Dashboard notifikasi tahun aktif satker /OPD		
Dashboard notifikasi perubahan pagu PKOR/PKOK		
Status upload file RKAD		
Rekap pengumuman RUP		
Notif perubahan paket		
Notif verifikasi akun PPK		
Rekap pengumuman RUP di PPK yang sama		
Rekap pengumuman RUP seluruh KLPD		
Model Rekapitulasi		
RENCANA PENGADAAN	2	SAD
Notifikasi Perubahan struktur anggaran		
Create paket dengan lintas anggaran kegiatan		
SRS SIRUP OFFLINE	1	Integrasi, Dashboard Moner, Rencana Pengadaan
SIRUP OFFLINE	10	SRS Sirup Offline
Sirup Installer		
Create Paket		

Download daftar paket		
Upload daftar paket		
Auto Upload daftar paket		
TOTAL MAIN DAYS	41	

6.3 Pengujian Metode

Pada tahap pengujian metode ini, dilakukan pengujian metode untuk mengetahui apakah *Critical Path Method* dapat menghasilkan total durasi pengerjaan proyek yang lebih optimal untuk membantu *Project Manajer* dalam melakukan perencanaan.



Gambar 6. 3 Perhitungan Critical Path Method

6.4 Evaluasi Metode

Pada data yang diperoleh hasil total waktu pengerjaan pada proyek LKPP-Sirup adalah 41 hari, sedangkan ketika diolah menggunakan Critical Path Method menghasilkan durasi waktu yang lebih optimal yaitu 27 hari.