**LAPORAN GELADI**

**KEMENTRIAN KOMUNIKASI DAN INFORMATIKA**

**JAKARTA PUSAT**

**Disusun untuk memenuhi**

**Tugas Geladi pada Semester 4**

****

**Disusun Oleh :**

**Samuel Edi Ronal 6306130007**

**TELKOM UNIVERSITY**

**BANDUNG**

**2015**

**Lembar Pengesahan**

*Laporan ini disusun untuk memenuhi syarat mata kuliah Geladi Program Studi Diploma III Teknik Informatika*

*Telkom University*

Disusun Oleh :  
Samuel Edi Ronal (6306130007)

Mengetahui,

(Pembimbing Akademik Telkom University) (Pembimbing Lapangan)

Yuyun Doni Marshal Rangga

NIP : 13831156-1 NIK : 1987032720110110006

Geladi adalah program kegiatan yang dirancang untuk memberikan pengalaman dan sosialisasi kerja kepada peserta gelada. Hal ini sangat penting, karena peserta geladiakan mengenal dan menghayati kondisidalam ruang lingkup lapangan kerja yang akan mereka pilih setelah lulus, serta dapat beradaptasi pada perubahan lingkungan pekerjaan.

Berdasarkan pengertian diatas peserta gelada disebar ke beberapa lokasi di Indonesia da nada juga yang memilih tempat sendiri untuk melaksanakan gelada. Pelaksanaan Geladi pada tahun ajaran 2015 ini dilaksanakan selama 6 minggu yang dimulai pada tanggal 1 Juni 2015 dan berakhir pada tanggal 10 Juli 2015 bertempat di Kementrian Komunikasi dan Informatika yang menangani penyusunan dokumen SRS, SPM, Software User, dan Test Document beberapa proyek pemerintahan

Dengan masa gelada selama 6 minggu, saya memahami apa saja yang diperlukan pada saat pembuatan sebuah aplikasi pendukung sarana publik. Saya memahami bagaimana pembuatan UML yang baik dan benar, serta mengerti apa yang telah dilakukan pemerintah untuk kemajuan bangsa dan negara.

1. **PENDAHULUAN**
   1. **Latar Belakang**

Dewasa ini kebutuhan memperoleh informasi sangat meningkat dan semua ini dapat dirasakan oleh seluruh elemen masyarakat. Masyarakat berhak mengetahui tentang apa saja yang sedang terjadi di lingkungan sekitar. Masyarakat tidak boleh pasif terhadap proyek pemerintahan tetapi harus ikut serta dalam proyek tersebut agar meningkatkan pembangunan sarana publik.

Dalam hal ini, pemerintah khususnya bagian Kominfo memiliki beberapa kendala untuk membuat sarana public yang memadai bagi seluruh masyarakat. Seluruh elemen masyarakat dituntut agar tidak pasif terhadap perkembangan teknologi informasi tetapi turut serta menciptakan masyarakat Indonesia yang modern. Dengan masyarakat modern maka Indonesia akan menjadi sasaran utama pasar global di dunia. Dalam hal ini, maka angka kemiskinan yang ada di Indonesia dapat berkurang seiring dengan kemajuan teknologi informasi.

Pada kesempatan kali inilah, Telkom University memberangkatkan mahasiswa mereka untuk melaksanakan Kegiatan Geladi 2015. Kegiatan praktek ini dilakukan di Kominfo yang tersebar diseluruh Indonesia. Kegiatan praktek ini menguntungkan bagi mahasiswa dan mahasiswi karena dapat menambah pengetahuan khususnya dalam bidang teknologi pengalaman serta keprofesionalan dalam melakukan suatu bidang pekerjaan. Disamping itu, kegiatan praktek ini sangat berpengaruh terhadap nilai Mata Kuliah “Geladi” yang dicanangkan oleh Telkom University.

* 1. **Ruang Lingkup**

Semenjak pertama kali program kegiatan Geladi dilaksanakan pada tahun 1992, mahasiswa dan mahasiswi yang dahulunya bernama Institut Teknologi Telkom yang sekarang berubah nama menjadi Telkom University ditugaskan untuk melakukan kerja praktek dilingkungan PT. TELKOM maupun diluar PT. TELKOM dengan pekerjaan setingkat jurusan. Dalam pelaksanaannya, setiap mahasiswa dituntuk untuk dapat memberikan manfat untuk tempat kerjanya. Ini berarti bahwa setiap pengiriman mahasiswa untuk Geladi di suatu instansi harus disesuaikan dengan kebutuhan yang ada di instransi tersebut.

Kegiatan gelada dapat dilanjutkan dengan hal-hal yang lebih spesifik yang tergantung pada program studi mahasiswa.

* 1. **Rumusan Masalah**

1. Apa yang dimaksud dengan Software Requirement Specification (SRS)?
2. Apa yang dimaksud dengan Software Project Management (SPM)?
3. Apa yang dimaksud UML?
4. Sebutkan jenis-jenis UML !
   1. **Tujuan**
5. Tujuan Umum

Dalam kegiatan kurikuler di Telkom University, Geladi merupakan suatu kegiatan yang harus ditempuh oleh setiap mahasiswa dari semua jurusan, baik untuk program S1 maupun D3. Tujuan dilaksanakannya gelada tersebut adalah

1. Untuk memberikan pengalaman kerja praktek secara langsung serta menggali berbagai masalah atau pekerjaan yang timbul di lapangan
2. Untuk meningkatkan keterampilan dan wawasan, baik secara teknik maupun hubungan kemanusiaan
3. Untuk membentuk perilkasu positif bagi para mahasiswa peserta Geladi melalui penyesuaian diri dengan lingkungan kerja tempat Geladi
4. Tujuan Khusus

Beberapa tujuan yang ingin dicapai melalui laporan ini sebagai berikut

1. Mendeskripsikan apa itu SRS
2. Mendeskripsikan apa itu SPM
3. Mendeskripsikan apa itu UML
4. Mendeskripsikan jenis-jenis UML
   1. **Manfaat**

Melalui laporan ini diharapkan wawasan pembaca terhadap pengenalan dunia telekomunikasi khususnya validasi data beserta perangkat-perangkatnya semakin bertambah sehingga dapat memberikan solusi dan ide yang cemerlang untuk dunia pertelekomunikasian di Indonesia

1. **TINJAUAN TEORI**

**2.1 Deskripsi Software Requirement Specification (SRS)**

Software Requirement Specification (SRS) adalah dokumen yang menjelaskan tentang berbagai kebutuhan yang harus dipenuhi oleh suatu software. Dokumen ini dibuat oleh developer (pembuat software) setelah menggali informasi dari calon pemakai software. Pembuatannya pun seharusnya mengikuti standar yang ada dan paling diakui oleh para praktisi rekayasa software di dunia. Oleh karena itu, standar yang akan dibahas disini adalah standar dari IEEE

IEEE membuat standar SRS agar dokumen pentin itu tidak ambigu dan tentu saja lengkap. Dengan standar itu, si pengguna dapat mencurahkan semua keinginannya terkait software tersebut dengan jelas dan akurat sehingga developer pun dapat memahami apa yang dinginkan pengguna yang tepat. Bahkan, bagi perorangan, standar ini dapat membantunya dalam mengembangkan outline SRS yang baku khususnya untuk perusahaan, membantu membuat dokumen SRS dengan format dan isi yang standar (minimal), serta membantunya mengembangkan rincian-rincian pendukung lainnya.

Berikut adalah manfaat dari penggunaan SRS, antara lain :

1. Sebagai bentuk perjanjian antara customer dan supplier tentang software apa yang akan dibuat
2. Mengurangi beban dalam proses pengembangan software
3. Sebagai bahan perkiraan biaya dan rencana penjadwalan
4. Sebagai dasar validasi dan verifikasi software di ujung penyelesaian proyek nantinya
5. Memfasitasi transfer, semisal software tersebut ingin ditransfer ke pengguna ataupun mesin. Customer akan merasa mudah jika ingin mentransfer software ke bagian-bagian lain dalam orginasasinya. Bahkan, jika pergantian personil developer, proyek dapat mudah ditransfer ke personil baru dengan memahami isi SRS tersebut
6. Mendasari perbaikan produk software di kemudian hari.

Ada beberapa istilah yang digunakan dan harus diketahui untuk memahami standar SRS yang dibuat IEEE ini. Istilah tersebut adalah

1. Kontrak

Dokumen yang mengikat secara hokum dan disepakati oleh customer dan supplier, termasuk syarat-syarat teknologi dan organisasi, biaya, serta jadwal pengerjaan. Kontrak bisa mengandung sesuatu yang kurang formal tetapi bermanfaat, seperti komitmen atau harapan

1. Customer

Pihak yang membayar untuk produk dan biasanya yang menentukan persyaratan

1. Supplier

Pihak yang membuat produk software untuk customer

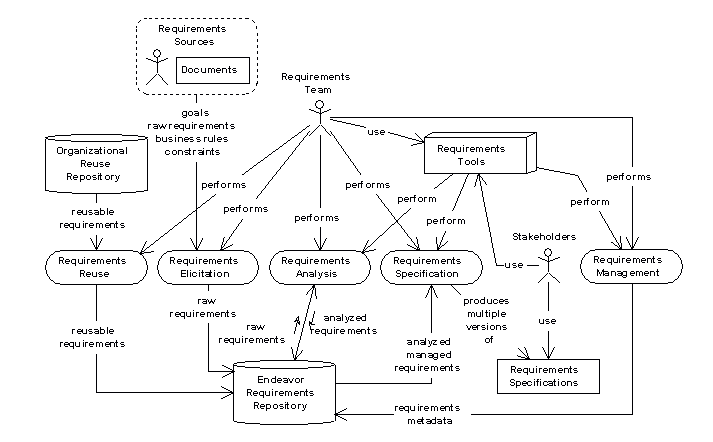
1. Pengguna

Pihak yang mengoperasikan atau berinteraksi langsung dengan software. Pengguna dan customer biasanya bukan orang yang sama

Beberapa pertimbangan dalam penyusunan SRS, yaitu

1. Sifat SRS
2. Lingkungan SRS
3. Karakteristik dari SRS yang baik, yaitu
4. Correct (benar)
5. Unambiguous (tidak ambigu)
6. Complete (lengkap)
7. Consisten (konsisten)
8. Ranked for importance (prioritas yang penting)
9. Verifiable (dapat diverifikasi)
10. Modifiable (dapat dimodifikasi)
11. Traceable (bisa dilacak)
12. Penyusunan SRS secara bersama-sama
13. Evolusi SRS
14. Membuat prototype, seperti model atau contoh
15. Mencantumkan desain system
16. Pencantuman persyaratan proyek di SRS

Berikut adalah gambaran SRS :



**2.2 Deskripsi Software Project Management (SPM)**

Software Project Management merupakan bidang yang menarik untuk dipelajari dan pekerjaan saya disini lebih difokuskan kepada apa yang telah saya lakukan selama ini. Suatu Software Project dapat dilakukan oleh Konsultan atas permintaan User atau juga bisa merupakan suatu internal development Project, dan dalam tulisan ini, yang dibahas adalah software project yang dilakukan oleh konsultan yang telah ditunjuk oleh user.

Apa yang saya lakukan dalam suatu software project management selama ini adalah dengan menerapkan SDLC (Software Development Life Cycle) yang terdiri dari :

1. Project Kick Off
2. Project Plan (Menentukan Jadwal dan Resource)
3. Requirement Gathering and Analysis
4. Design
5. Development / Coding
6. System Integration Test
7. User Acceptance Test
8. Data Migration
9. Go Live

Berikut adalah penjelasan singkat terkait hal-hal tersebut di atas

1. Project Kick Off

Tahap ini merupakan awal dari project. Tahapan ini dilakukan dalam suatu rapat yang dihadiri oleh Tim User dan Konsultan. User menyiapkan summary cakupan project serta tahapan dan deliverables yang diharapkan. Tahap ini dilakukan dalam bentuk rapat resmin disertai dengan notulensi. Catatan rapat ini akan didistribusikan ke seluruh pihak yang terkait dengan project dan menjadi landasan kegiatan selanjutnya. Pada tahap ini juga ditentukana Person In Charge (PIC) baik dari sisi User maupun dari sisi Konsultan. Project Kick Off dilakukan dengan merujuk kepada kontrak atau SPK yang telah ditandatangani.

1. Project Plan

Perencanaan project merupakan tahap yang sangat penting. Pada tahap ini, project manajer membuat draft jadwal atas keseluruhan project, sehingga dapat memberikan gambaran kepada setiap orang yang terlibat dalam project. Sehingga target utama dari Plan Project ini adalah untuk mendapatkan gambaran kapan setiap tahapan project dilakukan dan kapan selesainya serta siapa saja personal yang terlibat dalam project. Hal ini dapat dibuat berdasarkan kontrak pekerjaan yang telah dibuat dan ditandatangani sebelumnya.

1. Business Requirement Gathering and Analisys

Pendefenisian masalah merupakan hal yang esensi dari sebuah Software Project. Setiap bagian/unit/divisi yang akan menjadi pengguna software, wajib mengirimkan perwakilan pada proses ini. Tanpa keterwakilan dari salah satu bagian/unit/divisi, assasment kebutuhan menjadi tidak tepat yang pada ujungnya akan memberikan solusi software yang tidak sesuai dengan kebutuhan.

Pada tahap ini sering kali terjadi konflik kepentingan antara pekerjaan operasional dengan menghadiri meeting requirement assessment, untuk itu sangat diperlkukan dukungan penuh dari pimpinan perusahaan dari sisi user untuk memberikan prioritas utama pada project ini. Solusi konflik kepentingan ini sering kali dengan menetapkan minimal satu orang perwakilan dari setiap bagian/unit/divisi yang terlibat secara penuh project ini dari awal sampai akhir. Jadi walaupun masih ada tanggung jawab operasional, PIC ini tetap memprioritaskan waktunya diproject. Untuk melengkapi hal tersebut, penentuan PIC sebaiknya juga disertai dengna penentapan KPI (Key Performance Indicator) tambahan atas karyawan tersebut atas keterlibatannya di Project Software ini.

1. Design

Tahap desain sangat menetukan kualitas atas software yang akan dibuat. Pada tahap desain dilakukan pembuatan Flow Process, Data Flow Diagram, Entity Relationship Diagram, Program Framework dan Struktur Class dan aspek teknik lainnya. Seluruh pekerjaan pada tahap desain dibuat berdasarkan FSD yang telah disepakati pada tahap sebelumnya. Desain solusi yang baik akan sangat memudahkan dalam pembuatan program yang akan dikembangkan. Data Flow Diagram yang efektif dan efesien akan membuat solusi menjadi lebih cepat dan mudah direalisasikan. Entity Relationship Diagram menentukan kualitas database yang akan dibangun. Kesalahan yang sering terjadi adalah ERD tidak dibuat secara akurat sehingga menghasilkan kualitas database yang redundan dan tidak efisien. DFD merupakan alur data dari business process yang sedang dipelajari sedangkan ERD merupakan tipe hubungan antara 2 atau lebih entitas di dalam business process tersebut

1. Coding / Development

Inti dari project software adalah coding / development program. Umumnya, kualitas dari program sering berdasarkan pada kualitas si programmer yang bersangkutan. Software Project Management yang baik akan membuatkan struktur class yang lengkap dan stabil sebagai framework utama. Dengan pembuatan strukturr call dan framework yang baku, variasi dan kesalahan programmer dapat diminimalisasi. Tanpa itu akan terjadi tingkat variasi dan kesalahan yang sering kali tinggi dan mengurangi kualitas software yang dibangun, untuk itu peran senior developer dalam suatu project software akan sangat penting, terutama saat ini banyak sekali pilihan bahasa program serta Integrated Development Environmentnya (IDE). Namun yang paling banyak digunakan adalah Java, PHP, Microsoft Visual Studio, serta untuk mobile yaitu Android, Apple, dan BlackBerry. Terlepas dari perbedaan pemilihan tools development, penerapan konsep Object Oriented Programming (OOP) merupa suatu keharusan dan tetap dijaga untuk memberikan kualitas software yang efektif dan efisien, konsep class dan framework yang baik akan memberikan kemudahan dan keseragaman dalam pengembangan software.

1. Pembuatan Test Script

Tahap ini dilakukan bersamaan dengan tahapan Coding / Development yang dilakukan oleh Business Analysist bersama tim User, tujuannya adalah membuat suatu scenario test yang lengkap dan komprehensif sesuai dengan proses real yang diinginkan user sehingga bisa menggambarkan kondisi proses sebenarnya. Test script ini harus benar benar mewakili cerita sebenarnya yang dilengkapi dengan contoh nilai-nilai masukan beserta hasil yang diharapkan. Dengan adanya test script yang lengkap dan komprehensif, maka testing dilakukan oleh siapa saja, walaupun tidak terlibat di tahap awal project software, dimana orang yang akan melakukan testing, cukup memasukkan nilai awal sesuai test script dan melihat apakah nilai yang dihasilkan sudah sesuai dengan hasil perhitungan manual yang tercantum pada test script. Kasus-kasus test script harus divariasikan sesuai dengan kemungkinan variasi pada proses sebenarnya.

1. User Acceptance Test (UAT)

Tahap ini kurang lebih sama dengan yang dilakukan pada tahap SIT, hanya saja yang melakukan adalah bagian/unit/divisi terkait. Tantangan yang muncul pada tahapan ini adalah memastikan agar user terkait dapat menghadiri jadwal UAT sesuai dengan waktu yang disepakati. Suatu system yang diintegrasikan biasanya melibat beberapa bagian/unit/divisi, segingga ketidakhadiran salahsatu perwakilan dari bagian/unit/divisi tertentu akan memundurkan jadwal UAT sehingga jadwal keseluruhan juga jadi terganggu. Untuk itu perlu dilakukan pendekatan secara dini kesetiap pimpinan bagian/unit/divisi terkait sehingga mempunyai kesadaran dan persepsi yang sma mengenai pentingnya software ang sedang dikembangkan. Walaupun batasan pekerjaan dan batasan proses ujicoba sudah digrasikan dengan test script yang disepakati sebelumnnya, namun demikian tidak jarang pada saat UAT, saat menemukan suatu variasi testing yang ternyata tidak ada di test cript dan hal terburuknya adalah feature yang tidak tersedia tersebut harus didevelop oleh programmer

1. Data Migration

Software dibuat dengan tujuan untuk menggantikan proses manual yang selama ini terjadi menjadi suatu proses yang otomatis, atau mengganti system lama dengan system baru yang dianggap lebih baik atau melakukan penambahan program atas existing program. Jenis manapun dari pengembangan software tersebut di atas harus melalui tahap yang dinamakan Data Migration atau Migrasi Data. Untuk pelaksanaan data migration, beberapa hal yang harus disiapkan yaitu penyiapan existing data yang digunakan secara manual atau yang digunakan oleh system yang lama, pembuatan script untuk melakukan one time migration dimana data yang lama akan diupload system baru dengan menggunakan script migrasi tersebut.

1. Go Live

Ini adalah suatu tahap dimana semua proses SDLC sudah selesai dan user sudah bisa menggunakan system baru dengan existing data. setelahGo Live bukan berarti tidak ada problem lagi, sering suatu system akan terlihat masalahnya pada saat system Go Live di production server, namun dengna penanganan proses project manajemen yang handal seharusnya problem ini dapat diminimalisasi.

1. Support

Untuk menjamin agar system berjalan bagus dan stabil setelah production, diperlukan mekanisme support yang efektif, dengan Service Level Agreemnt (SLA) yang disepakati oleh User dan Vendor. Issue biasanya digolongkan menjadi Critical/Stopper, Urgent, Important dan Nice to Have dan masing-masing kriteria issue ini akan disolusikan dengan SLA yang berbeda-beda.

Berikut adalah gambaran untuk SPM :



**2.3 Deskripsi Unified Modeling Language (UML)**

UML merupakan salah satu alat bantu yang dapat digunakan dalam bahasa pemograman yang beriorientasi objek, saat ini UML akan mulai menjadi standar masa depan bagi industry pengembangan system/perangkat lunak yang berorientasi objek sebab pada dasarnya UML digunakan oleh banyak perusahaan raksasa seperti IBM, Microsoft, dll.

Langkah-langkah penggunaan Unified Modeling Language (UML)

1. Buatlah daftar business process dari level tertinggi untuk mendefenisikan aktivitas dan proses yang mungkin muncul
2. Petakan *use case* untuk setiap business process untuk mendefenisikan dengan tepat fungsional yang harus disediakan oleh system, kemudian perhalus use case diagram dan lengkapi dengan requirement, constraints dan catatan-catatan lain
3. Buatlah deployment diagram secara kasar untuk mendefinisikan arsitektur fisik system
4. Defenisikan requirement lain non fungsional, security dan sebagainya yang juga disediakan oleh system
5. Berdasarkan use case diagram, mulailah membuat activity diagram
6. Defenisikan obyek-obyek level atas package atau domain dan buatlah sequence dan/atau collaboration untuk setiap alir pekerjaan, jika sebuah use case memiliki kemungkinan alir normal dan error, buat lagi satu diagram untuk masing-masing alir
7. Buat rancangan user interface model yang menyediakan antarmuka bagi pengguna untuk menjalankan scenario use case
8. Berdasarkan model-model yang sudah ada, buatlah class diagram setiap package atau domain dipecah menjadi hirarki class lengkap dengan atribut dan metodenya. Akan lebih baik jika untuk setiap class dibuat unit test untuk menguji fungsionalitas class dan interaksi dengan class lain
9. Setelah class diagram dibuat, kita dapat melihat kemungkinan pengelompokkan class menjadi komponen-kompnen karena itu buatlah component diagram pada tahap ini. Selain itu, definisikan test integrasio setiap komponen untuk meyakinkan ia dapat bereaksi dengan baik
10. Perhalus deployment diagram yang sudah dibuat. Detailkan kemampuan dan requirement piranti lunak, system operasi, jaringan dan sebagainya. Petakan komponen ke dalam node
11. Mulailah membangun system. Ada dua pendekatan yang tepat digunakan
12. Pendekatan use case dengan mengassign setiap use case kepada tim pengembang tertentu untuk mengembangkan unit kode yang lengkap dengan test
13. Pendekatan komponen yaitu mengassign setiap komponen kepada tim pengembangan tertentu
14. Lakukan uji modul dan uji integrasi serta perbaiki model beserta codenya. Model harus selalu sesuai dengan code yang actual
15. Perangkat lunak siap dirilis

**2.3 Deskripsi Jenis – Jenis Diagram UML**

Jenis – jenis diagram UML :

1. Use Case Diagram

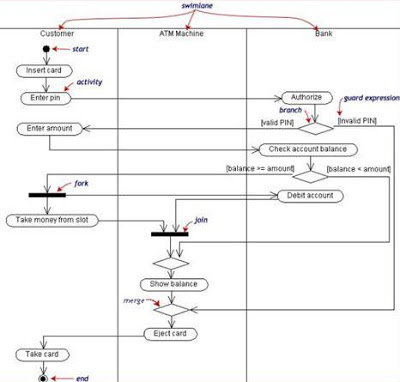
Use case adalah abstraksi dari interaksi antara system dan actor. User case bekerja dengan cara mendeskripsikan tipe interaksi antara user sebuah system dengan systemnya sendiri melalui sebuah cerita bagaimana sebuah system dipakai.



Diagram use case berguna dalam 3 hal, yaitu :

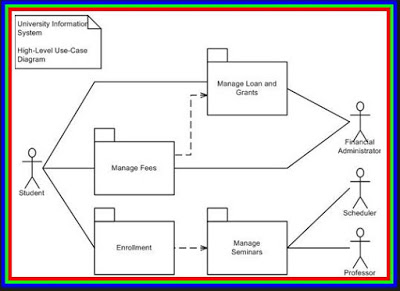
1. Menjelaskan fasilitas yang ada (requirement)
2. Komunikasi dengan klien
3. Membuat test dari kasus-kasus secara umum
4. Activity Diagram

Activity Diagram menyediakan analisis dengan kemampuan untuk memodelkan proses dalam suatu system informasi. Activity Diagram dapat digunakan untuk alur kerja model use case individual atau logika keputusan yang terkandung dalam metode individual



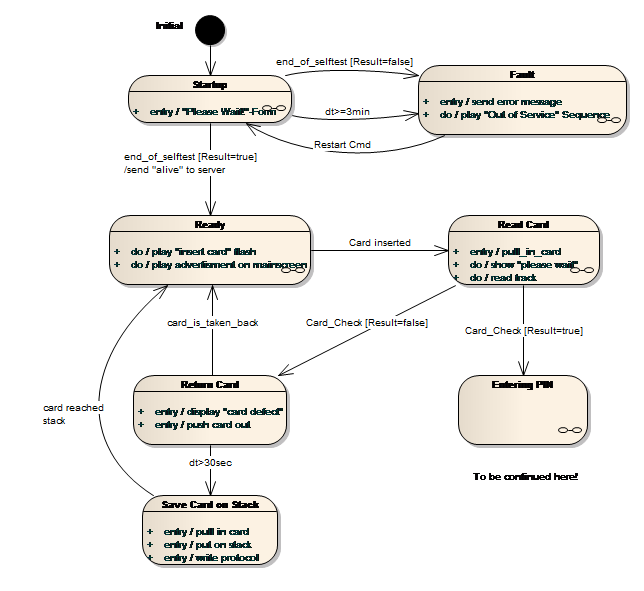
1. Package Diagram

Package Diagram utamanya digunakan untuk mengelompokkan elemen diagram UML yang berlainan secara bersama-sama ke dalam tingkat pembangunan yang lebih tinggi yaitu berupa sebuah paket. Package diagram pada dasarnya adalah class diagram yang hanya menampilkan package, disamping class, dan hubungan khas yang ditampilkan pada class diagram.



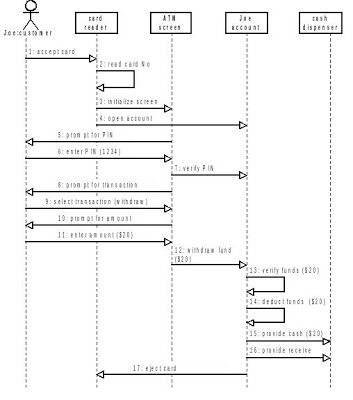
1. State Machines Diagram

State diagram menggambarkan urutan keadaan yang dilalui objek dalam suatu kelas, kerena suatu kejadian menyebabkan suatu perpindahan aktivitas/state. State dari objek adalah penggolongan dari satu atau lebih nilai attribute pada kelas.



1. Sequence Diagram

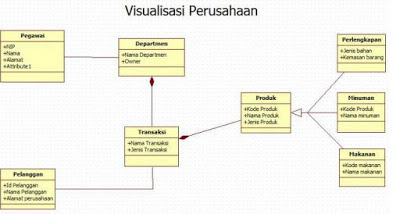
Sequence diagram menjelaskan interaksi objek yang disusun berdasarkan urutan waktu. Secara mudahnya sequence diagram adalah gambaran tahap demi tahap yang seharusnya dilakukan untuk menghasilkan sesuatu sesuai dengan use case diagram. Sequence diagram menekankan penyusunan berbasis waktu untuk kegiatan yang dilakukan dengan satu set dari objek yang berkolaborasi. Sequence diagram sangat berguna dalam membantu analisis, memahami spesifikasi real-time dan menggunakan kasus yang rumit.



1. Class diagram

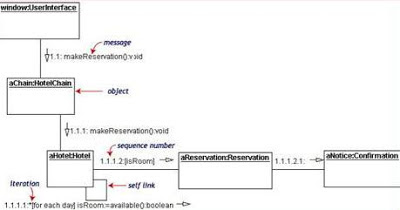
Class adalah deskripsi kelompok objek-objek dengan property, perilaku (operasi) dan relasi yang sama. Sehingga dengan adanya classdiagram dapat memberikan pandangan global atas sebuah system. Hal tersebut tercermin dari class-class yang ada dan relasinya satu dengan yang lainnya. Sebuah system biasanya mempunyai beberapa class diagram. Class diagram sangat membantu dalam visualisasi struktur kelas dari suatu system

Class diagram berfungsi untuk menjelaskan tipe dari objek system dan hubungannya dengan objek lain. Objek adalah nilai tertentu dari setiap attribute kelas entity.



1. Communication Diagram

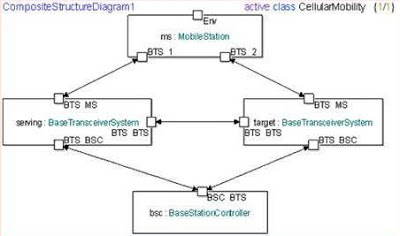
Communication diagram menggambarkan interaksi antar objek seperti sequence diagram, tetapi lebih menekankan pada peran masing-masing objek. Setiap message memiliki sequence number, dimana message dari lever tertinggi memiliki nomor 1.



1. Composite Structure Diagram

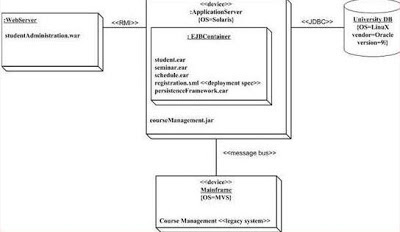
Diagram struktur komposit adalah diagram yang menunjukkan struktur internal classifier, termasuk point interaksinya ke bagian lain dari system. Hal ini menunjukkan konfigurasi dan hubungan bagian, yang bersama-sama melakukan perilaku classifier.

Diagram struktur komposit merupakan jenis diagram struktur yang statis dalam UML, yang menggambarkan struktur internal kelass dan kolaborasi.



1. Object Diagram

Object diagram merupakan sebuah gambaran tentang objek dalam sebuah system pada satu titik waktu. Karena lebih menonjolkan perintah-perintah dari pada class, object diagram lebih sering disebut sebagai sebuah diagram perintah.



1. Timing Diagram

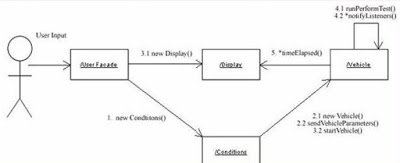
Memperlihatkan interaksi ketika tujuan utama diagram adalah waktu. Menggambakan perubahan dalam state atau kondisis dari pengelompokan instance atau tugas berlebihan.

Biasanya dipakai untuk memperlihatkan perilaku dari sebuah/beberapa object melalui periode waktu



1. Component Diagram

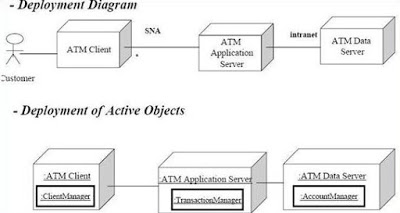
Diagram ini bila dikombinasikan dengan diagram penyebaran dapat digunakan untuk menggambarkan distribusi fisik dari modul perangkat lunak melalui jaringan. Misal, ketika merancang system client-server,hal ini berguna untuk menunjukkan mana kelas atau paket kelass akan berada pada node klien dan mana yang akan berada di server.



1. Deployment Diagram

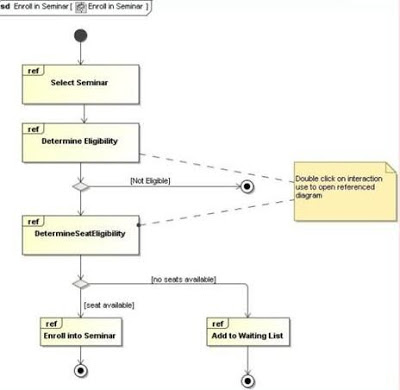
Menggambarkan detail bagaimana komponen di deploy dalam infrastruktur system, diaman komponen akan terletak (pada mesin, server , atau piranti keras), bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi server, dan hal-hal lain yang bersifat fisikal.

Hubungan antar node (misalnnya TCP/IP) dan requiremen dapat juga didefenisikan dalam diagram ini



1. Interaction Overview Diagram

Merupakan pecangkolan secara bersama antara activity diagram dengan sequence diagram. Interaction Overview Diagram dapat dianggap sebagai activity diagram dimana semua aktivitas diganti dengan sedikit sequence diagram, atau bisa dianggap sebagai sequence diagram yang dirincikan dengan notasi activity diagram yang digunakan untuk menunjukkan aliran pengawasan.



1. **PELAKSANAAN GELADI**
   1. **Rencana Kegiatan**

Rencana kegiatan yang akan dilaksanakan pada pelaksanaan gelada yang berlangsung selama enam minggu pada Kominfo Jakarta Pusat adalah sebagai berikut:

1. Pembukaan Geladi dan Pembagian Kelompok
2. Pengenalan tentang SRS, SPM, Git, Software User
3. Pembuatan SRS, SPM, Software User dan Test Document
4. Validasi Data
   1. **Pelaksanaan**

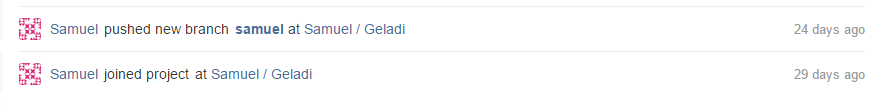
Kegiatan yang dilakukan di Kominfo Jakarta Pusat merupakan kegiatan rutin untuk mempermudah pekerjaan.

Adapun kegiatan yang dilakukan yaitu :

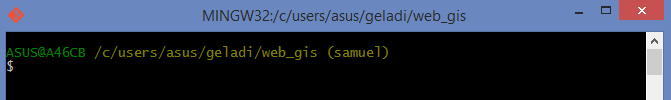
1. Membuat Dokumen SRS, SPM, Software User, dan Test Document
2. Login Username dalam Gitlab Kominfo
3. Membuat branch dalam Gitlab Kominfo
4. Melakukan push ke dalam Gitlab tersebut

Langkah-langkah push ke dalam Gitlab Kominfo :

1. Untuk melakukan push kedalam Gitlab Kominfo maka terlebih dahulu kita membuat branch di dalam Gitlab.



1. Setelah itu, lakukan penginstalan Git dan dapat didownload di <https://git-scm.com/downloads>
2. Setelah selesai diinstal, buka Git Bash dan masuk kedalam directory folder yang telah selesai dikerjakan



1. Setelah itu lakukan add pada folder yang dikerjakan



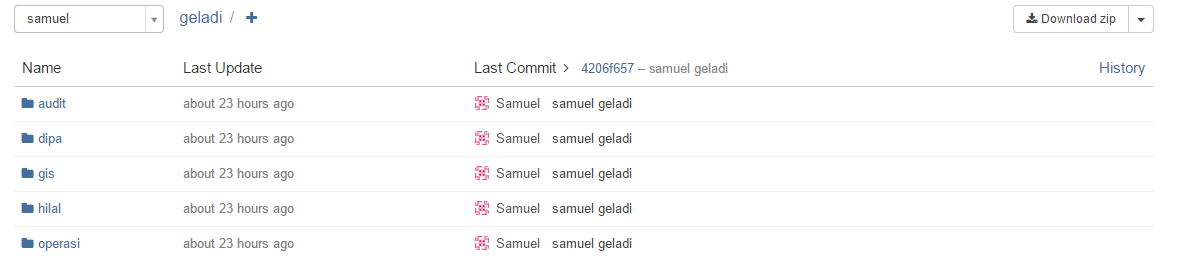
1. Lakukan commit terhadap folder tersebut



1. Lakukan push kedalam directory yang sudah dibuat pada Gitlab Kominfo dan masukkan username beserta password pada Gitlab



1. Tunggu sampai data selesai terupload
2. Lihat kedalam branch Gitlab yang sudah terupload tadi



1. Selesai
2. **PENUTUP**
   1. **Kesimpulan**

Dari hasil kegiatan Geladi ini didapatkan beberapa kesimpulan sebagai berikut :

1. Program gelada merupakan program pendekatan mahasiswa dengan pihak industry dalam menyiapkan lulusan yang memiliki kemampuan yang utuh siap kerja
2. Validasi data adalah proses pengencekan seluruh komponen dalam data SRS maupun SPM
3. Proses validasi bertujuan untuk memvalidkan data dengan membenarkan error yang terdeteksi pada UML. Oleh karena itu, tingkat error/kesalahan yang terjadi dapat ditekankan serendah-rendahnya guna memperoleh validitas data yang tinggi.
4. Mempelajari dan mengetahui seluk beluk dokumen SRS maupun SPM, software user dan test document.
   1. **Saran**

Ada beberapa hal yang perlu didalami untuk membuat kulitas Geladi maupun system kerja di Kominfo menjadi lebih baik, antara lain :

1. Pencapaian target yang diharapkan sebaiknya disampaikan diawal Geladi sehingga membantu mahasiswa untuk mengatur waktu yang diberikan
2. Pembagian kelompok, baik komposisi dan anggota, agar lebih disesuaikan berdasarkan kebutuhan
3. Peserta gelada maupun siswa PKL yang berada di wilayah Kominfo Jakarta Pusat khsusnya untuk dibekali terlebih dahulu mengenai dsar-dasar yang akan dilakukan selama Geladi maupun PKL
4. Sebaiknya pihak Institusi dapat memantau secara rutin pelaksanaan Geladi melalui pembimbing lapangan agar mahasiswa dapat serius dan bertanggung jawab dalam melaksanakan semua kegiatan yang diberikan pada waktu Geladi