Tugas-UAS-[Individu]- Pengembangan dan Penerapan Sistem

Disusun Oleh :

Rizqi Ramadhan 6026221002

Dosen Pengampu :

Dr. Ir. Aris Tjahyanto, M.Kom.

DEPARTEMEN SISTEM INFORMASI

FAKULTAS TEKNOLOGI ELEKTRO DAN INFORMATIKA CERDAS

INSTITUT TEKNOLOGI SEPULUH NOPEMBER

SURABAYA

2022

0. Jelaskan overview terkait projek secara keseluruhan dan bagian projek yang menjadi bagian tanggung jawab anda. Jelaskan bagian yang anda kerjakan. Penjelasan antara lain terkait latar belakang adanya projek ini, permasalahan yang perlu diselesaikan pada projek ini, tujuan, dsb. Hasil pekerjaan atau tugas sebelumnya, dapat dilampirkan pada bagian akhir laporan ini.
1. Soal yang ini terkait dengan bagaimana anda membagi pekerjaan pada anggota tim.
   a) Bagaimana kelompok anda membagi pekerjaan untuk masing-masing anggota tim?
   b) Apakah cara pembagiannya tsb berusaha mendukung prinsip cohesion dan coupling.
   c) Diagram UML yang mana yang kemungkinan dapat dipergunakan untuk memeriksa cohesion dan coupling? Jelaskan.
   d) Apa saja manfaat high cohesion dan low coupling pada saat develop software?
2. Salah satu pertimbangan dalam menentukan granularitas atau ukuran servis pada arsitektur SOA adalah message size. Layanan dengan granularitas kasar (coarse-grain) cenderung mengirimkan lebih banyak data daripada layanan berbutir halus (fine-grain), termasuk data yang tidak secara khusus diperlukan untuk tugas tersebut, sehingga tentu saja akan membebani sistem.
   (a) Jelaskan bagaimana teknik yang anda gunakan untuk mengirimkan message untuk mengimplementasi arsitektur SOA pada software yang anda buat.
   (b) Tunjukkan dan jelaskan pada salah satu kode program dari servis yang telah anda buat bagaimana suatu message dikirim dan diterima (pilih salah satu method GET/POST/PUT/…).
3. (a) Tunjukkan arsitektur final dari SOA yang anda rancang.
   (b) Tunjukkan servis yang akan dibuat dan bagaimana mengakses servis yang disediakan tersebut.
   (c) Tunjukkan bagaimana anda meng-orkestrasi servis-servis tersebut menjadi salah satu layanan use case.
   (d) Tunjukkan dan jelaskan source code yang dihasilkan dari 3(c).

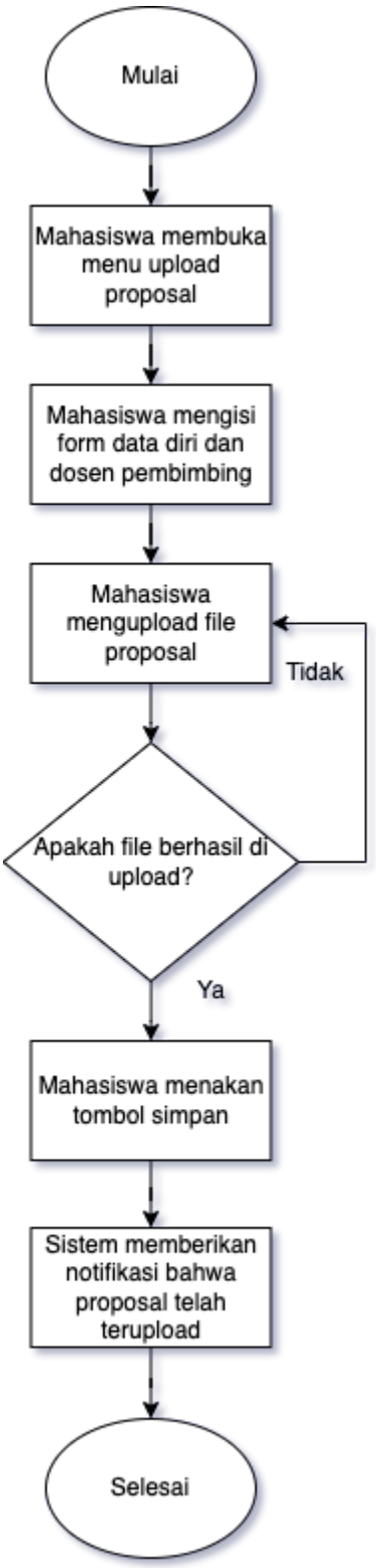
4. Anggap saja anda sudah menyelesaikan seluruh desain model yang menjadi bagian anda, kemudian memberikan penjelasan ke programmer agar dapat mewujudkan program computer/ kodingan berdasarkan desain model UML/DFD).
   (a) Jelaskan secara garis besar bagaimana anda melakukan transformasi dari MODEL ke CODE, mulai dari struktur menu paling atas sampai dengan ke database dari modul bagian anda. Tulis dalam bentuk kumpulan poin-poin penting.
   (b) Jelaskan bagaimana anda menyusun struktur menu berdasarkan model UML? Berikan contoh hasil struktur menunya beserta asalnya.
   (c) Struktur menu tsb dapat dipetakan langsung ke model UML yang mana? Tunjukkan.
   (d) Bagaimana anda menyusun skeleton atau kerangka program berdasarkan model?
   (e) Tunjukkan bahwa dengan activity diagram atau sequence diagram atau diagram lain serta user interface dapat digunakan untuk menjelaskan alur program (fokus ke Soal nomor 3).

(f) Jelaskan bagaimana anda merancang prototype berdasarkan use case diagram, use case specification, dsb yang telah disebutkan di jawaban soal nomor 0.

JAWABAN

0. Secara singkat project yang telah kami kerjakan merupakan project lanjutan dari SIMTA yakni SIMTA NG 2.0, yang merupakan pembaharuan dari SIMTA versi sebelumnya, yaitu versi yang pertama. Tujuan dari pengembangan proyek ini adalah penyempurnaan dari generasi yang pertama, seperti penambahan fitur-fitur yang dianggap perlu maupun penyederhanaan dari fitur sebelumnya sehingga menjadi SIMTA yang lebih *user friendly* dan mudah digunakan. Kemudian untuk bagian fitur-fitur yang saya kerjakan beberapa diantaranya yakni, Mendaftarkan Proposal TA, Revisi Proposal TA, Acc Lembar Persetujuan, UC009. Lihat Jadwal Sidang Proposal/TA, UC011. Isi Nilai & Kelulusan Pra TA/TA. Salah satu fitur yang perlu dibenahi adalah dalam pemberian sistem *tracking* pada status TA pada akun SIMTA mahasiswa. Fitur ini dirasa perlu untuk diperbarui karena selama ini dosen seringkali kesulitan untuk mengetahui status tracking pengerjaan tugas akhir dari mahasiswa yang dibimbingnya, sehingga beberapa kali pengajuan tugas akhir mahasiswa terlambat dikonfirmasi dan *accept* oleh dosen. Perbaikan fitur juga perlu diberikan untuk push notification pada SIM TA milik Dosen dan juga Mahasiswa bisa berupa push notification by E-Mail ataupun SMS/WA. Fitur ini diharapkan mampu membantu *user* untuk segera melakukan dan menyelesaikan proses yang diberikan.

Dan berikut merupakan flowchart dari pembagian masing-masing use case yang akan dijadikan kedalam service.



1. Soal yang ini terkait dengan bagaimana anda membagi pekerjaan pada anggota tim.
   a) Dalam pembagian tugas untuk project yang kami kerjakan, kami membagi rata semua fitur dan fungsi yang akan kami benahi, bersama-sama membuat functional requirements, dan riset mengenai project yang akan kami kerjakan.

b) Cara pembagian project dalam tim kami juga mendukung prinsip dari cohesion dan coupling, dimana coupling berpengaruh bahwa kita harus pastikan bahwa suatu modul seminimal mungkin tergantung atau berpengaruh terhadap modul lainnya. Tujuannya jika ada update secara internal di dalam suatu modul, modul-modul lainnya tidak akan terlalu kena pengaruh. Kemudian untuk cohesion juga merupakan suatu prinsip yang penting dalam membuat perangkat lunak karena modul yang dibuat memiliki fungsi-fungsi yang serupa untuk satu tanggung jawab.

c) Diagram UML bagian use case diagram dalam hal ini bisa dipergunakan untuk memeriksa cohesion dan coupling, dikarenakan yang menggambarkan hubungan interaksi antara sistem dan aktor. *Use Case* dapat mendeskripsikan tipe interaksi antara si pengguna sistem dengan sistemnya. *Use Case* merupakan sesuatu yang mudah dipelajari. Langkah awal untuk melakukan pemodelan perlu adanya suatu diagram yang mampu menjabarkan aksi aktor dengan aksi dalam sistem itu sendiri.

d) Coupling juga dapat diartikan sebagai sebuah ukuran untuk mengukur berapa kuatnya sebuah element terhubung dengan element lain.Ukuran ini dipakai juga untuk mengetahui seberapa kuat informasi yang dimilikinya,atau ketergantungan ke elemen lain. Sebagai contoh jika anda bayangkan jika anda mengubah 1 modul A, tapi karena modul lain memiliki ketergatungan terhadap modul A,maka efek perubahan ini mungkin saja punya impact terhadap modul lain Karena itu low coupling sangat penting dalam perencanaan software. kohesi adalah keterikatan fungsi-fungsi di dalam suatu modul. Maksudnya adalah modul yang dibuat memiliki fungsi-fungsi yang serupa untuk satu tanggung jawab. Karena itu high cohesion sangat penting dalam perancangan software. Modul kohesi melakukan suatu tugas tunggal pada suatu prosedur perangkat lunak yang memerlukan sedikit interaksi dengan prosedur yang sedang dilakukan di bagian lain dari suatu program. Karena itu high cohesion sangat penting dalam perancangan software.

2. Salah satu pertimbangan dalam menentukan granularitas atau ukuran servis pada arsitektur SOA adalah message size. Layanan dengan granularitas kasar (coarse-grain) cenderung mengirimkan lebih banyak data daripada layanan berbutir halus (fine-grain), termasuk data yang tidak secara khusus diperlukan untuk tugas tersebut, sehingga tentu saja akan membebani sistem.

   a) Pada penerapan arsitektur SOA, penulis menggunakan beberapa implementasi step-step yang telah menjadi requirement dalam proses pembuatan arsitektur SOA diantaranya.

      - Decompose Business Process (into Granular Actions) dimana pada step ini proses bisnis dipecah menjadi tindakan granular.
      - Filter Out Unsuitable Actions, pada step ini menyaring beberapa aksi dan tindakan yang tidak sesuai

- Define Entity Service Candidates, kemudian pada step ini mendefinisikan setiap entitas yang digunakan apakah service tersebut agnostik atau non agnostik
- Identify Process-Specific Logic, Logika khusus proses dipisahkan menjadi lapisan layanan logisnya sendiri. Untuk proses bisnis tertentu, jenis logika ini biasanya dikelompokkan menjadi layanan tugas atau konsumen layanan yang bertindak sebagai pengontrol komposisi.
- Identify Resources, Sumber daya yang diidentifikasi pada tahap ini dapat diekspresikan menggunakan garis miring ke depan sebagai pembatas.
- Associate Service Capabilities with Resources and Methods, menentukan resource dan method pada GET dan POST
- Apply Service-Orientation, Dokumentasi proses bisnis yang kami gunakan sebagai input untuk proses pemodelan layanan dapat memberi kami tingkat pengetahuan tentang pemrosesan dasar yang diperlukan oleh masing-masing kandidat kemampuan layanan REST yang teridentifikasi.
- Identify Service Composition Candidates, Dokumentasikan interaksi kemampuan layanan paling umum yang dapat terjadi selama eksekusi logika proses bisnis.
- Analyze Processing Requirements, dalam step ini Kita perlu mempertimbangkan hal-hal berikut:
    - Manakah dari sumber daya yang diidentifikasi sejauh ini yang dapat dianggap sebagai utilitas-sentris?
    - Dapatkah tindakan yang dilakukan pada sumber daya yang berpusat pada bisnis dianggap berpusat pada utilitas (seperti tindakan pelaporan)?
    - Logika aplikasi dasar apa yang perlu dijalankan untuk memproses tindakan dan/atau sumber daya yang dicakup oleh kandidat kemampuan layanan?
    - Apakah logika aplikasi yang diperlukan sudah ada?
    - Apakah ada logika aplikasi yang diperlukan yang mencakup batasan aplikasi? (Dengan kata lain, apakah diperlukan lebih dari satu sistem untuk menyelesaikan tindakan?)
- Define Utility Service Candidates (and Associate Resources and Methods), Menggali catatan dari langkah-langkah proses sebelumnya terkait tindakan yang berpusat pada utilitas yang telah didokumentasikan selama ini. Dikombinasikan dengan penelitian yang mereka kumpulkan dari langkah Analisis Persyaratan Pemrosesan, mereka melanjutkan untuk menentukan dua layanan utilitas berikut.
- Define Microservice Candidates (and Associate Resources and Methods), Logika pemrosesan non-agnostik (tujuan tunggal) yang diidentifikasi sebelumnya untuk menentukan apakah ada unit logika ini yang memenuhi syarat untuk enkapsulasi oleh layanan mikro terpisah.

- Model layanan mikro dapat memperkenalkan arsitektur implementasi layanan yang sangat independen dan otonom yang dapat cocok untuk unit logika dengan permintaan pemrosesan tertentu
- Apply Service-Orientation
- Revise Candidate Service Compositions
- Revise Resource Definitions and Capability Candidate Grouping, Logika pemrosesan baru apa pun yang diidentifikasi di langkah sebelumnya dapat menghasilkan peluang untuk menambah dan/atau merevisi lebih jauh kumpulan sumber daya yang dimodelkan sejauh ini.

b) Source Code Service dengan menggunakan method GET dan POST

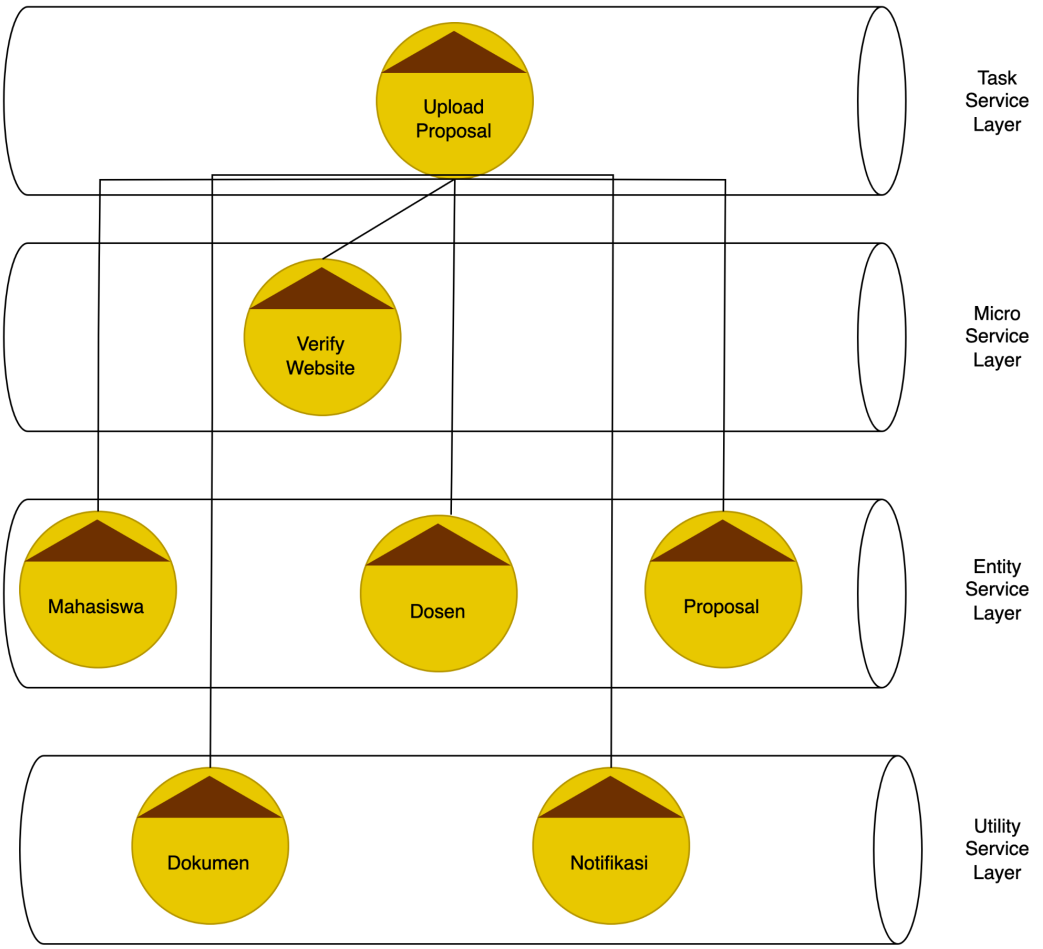
```
<form action="form_upload.php" method="get">
    <div>
        Dokumen: <input type="text"
name="Dokumen">
    <div>
        <input type="submit" value="Submit">
    </div>
</form>
```

Pada metode ini data yang dikirim adalah URL yang berupa rangkaian pasangan nama dan nilai yang dipisahkan oleh ampersand (&). File form_upload.php akan digunakan ketika tombol submit di tekan, dan akan menampilkan data dengan method get di file form_upload.php.

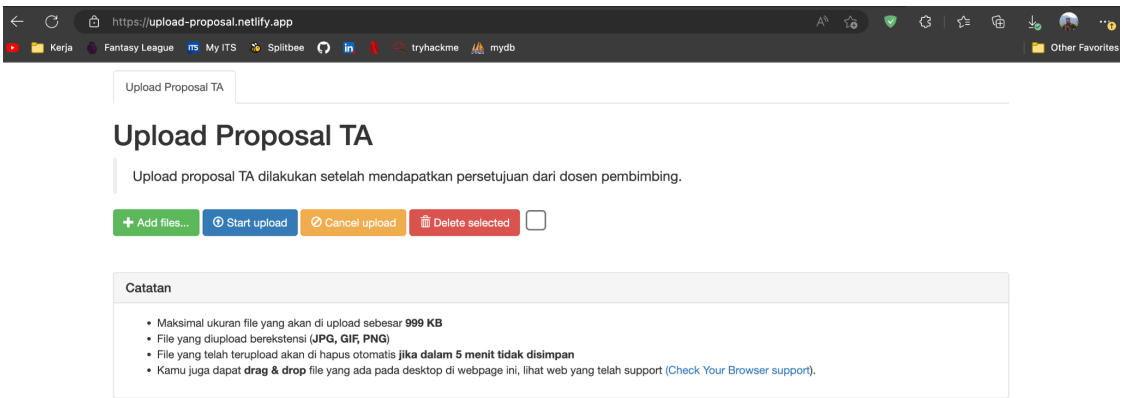
```
http://localhost/laravelproject/pps/upload_proposal.php?npm=12345678&Dokumen=submit
```

3. Arsitektur SOA yang telah dibuat.

a) Tunjukkan arsitektur final dari SOA yang anda rancang.



b) Service yang akan penulis buat yakni berupa form upload proposal untuk mahasiswa yang akan melaksanakan proses Sidang TA, dimana nantinya service ini hanya berisi service upload proposal, dapat dilihat pada gambar di bawah tampilan user interface untuk servie yang di tawarkan.



c)

d) wew

4. ss