

1. Tentukan pilihan anda sendiri, apakah lebih cenderung ke *Plain-Drive Development* ataukah *Agile Development*? Jelaskan alasan anda!
2. Buatlah satu desain *class object* yang terlibat di dalam sebuah *component process* menggunakan UML notaion!
3. Apa saran anda untuk mengurangi biaya pada tahapan validasi dan evaluasi sebuah pembangunan *software* ?

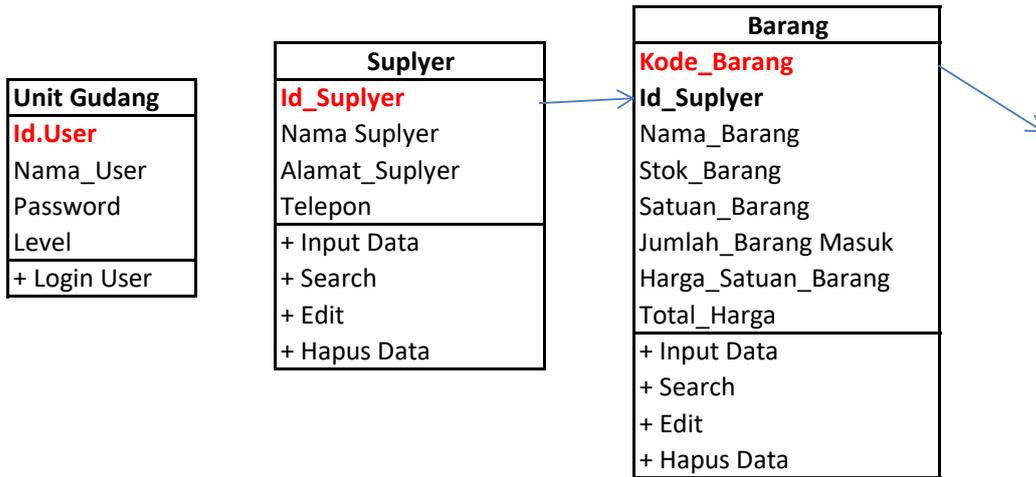
**1. Alasan kenapa pada saat saya membangun suatu project system atau software dari klien, saya lebih memilih Agile Development?**

- A. Karena dengan memakai metode agile development pada saat pembuatan project system/ software, klien terlibat langsung dalam pembuatan project tersebut agar project yang dibangun sesuai dengan harapan klien. Karena inti dari metode agile development adalah mementingkan kepuasan klien, Apabila Kepuasan klien terpenuhi maka klien siap membayar tinggi demi mewujudkan system/software yg diharapkan klien, sehingga kepuasan klien menjadi sumber pendapatan yang cukup besar bagi perusahaan.
- B. Jika pada saat pembangunan system terjadi kegagalan dan kerugian dari segi materi relatif kecil karena testing dapat dilakukan sesering mungkin sehingga memperkecil ditemukannya bug atau error yang fatal.
- C. Bagi perusahaan sendiri dengan menerapkan metode agile development perusahaan mendapatkan pengalaman dan tantangan yang baru. Karena semakin banyaknya pengalaman dan tantangan yang kita dapatkan akan menjadikan perusahaan kita dikenal banyak orang dan bisnis perusahaan semakin meningkat.
- D. Proses pembuatan system/ software lebih cepat selesainya dikarenakan banyak team yang terlibat dan interaksi antar team lebih intensif.
- E. Dapat memprediksi berapa biaya yang dikeluarkan pada saat proses pembuatan system/ software karena klien terlibat langsung dari awal pembangunan system atau project tersebut.

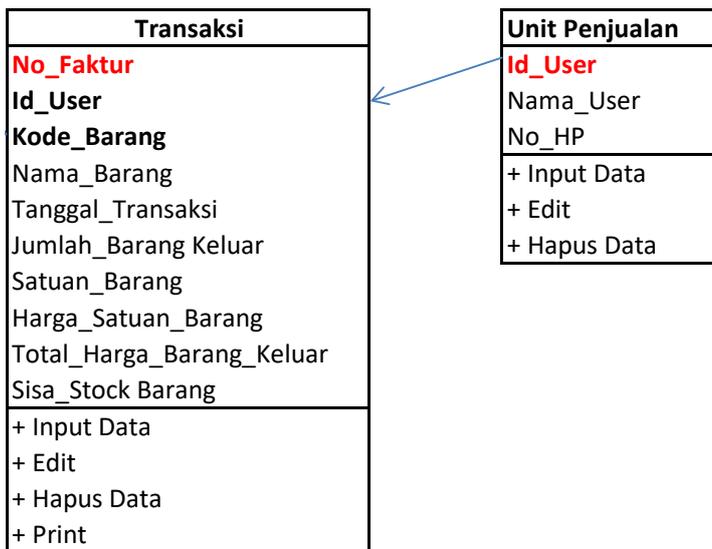
**2. Saran saya mengurangi biaya pada tahap validasi dan evaluasi dalam membangun software adalah:**

Dengan cara melibatkan langsung klien pada saat proses pembuatan software karena pada saat tahap penyelesaian pembuatan software tidak ada lagi validasi yang salah dan evaluasi yang akan dibuat dikarenakan klien sudah terlibat langsung dari awal proses pembuatan software.

## UML DIAGRAM CLASS PADA APLIKASI



## STOCK BARANG



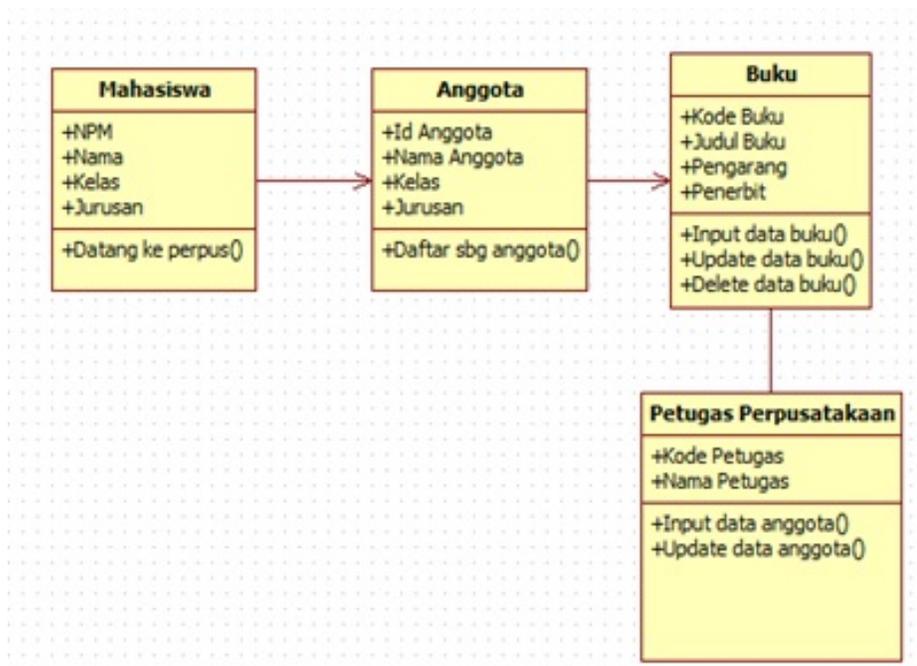
## Tugas Mata Kuliah Advanced Analysis dan Design

### Soal

1. Tentukan pilihan anda sendiri, apakah lebih cenderung ke *Plain-Drive Development* ataukah *Agile Development*? Jelaskan alasan anda!
2. Buatlah satu desain *class object* yang terlibat di dalam sebuah *component process* menggunakan UML notaion!
3. Apa saran anda untuk mengurangi biaya pada tahapan validasi dan evaluasi sebuah pembangunan *software* ?

### Jawaban

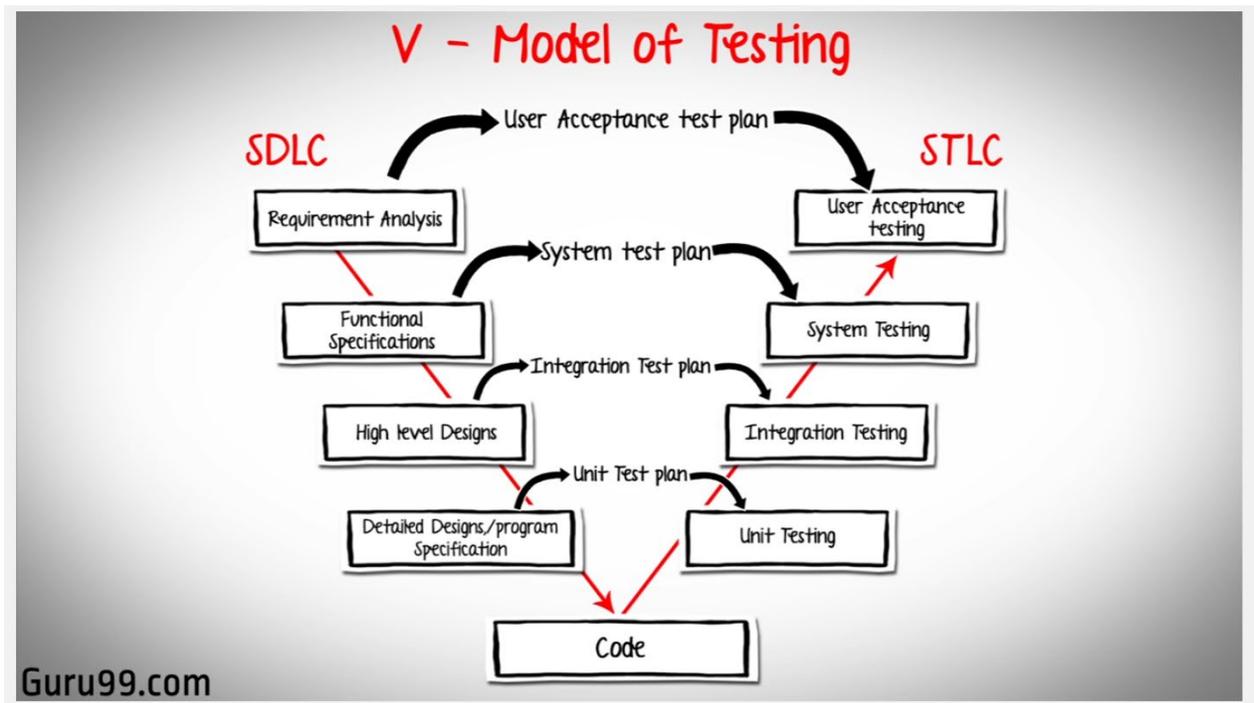
1. Lebih cenderung kepada *Agile Development* karena kepuasan pelanggan adalah nomor satu bagi perusahaan saya. Metode *Agile Development* ini memungkinkan untuk melakukan review pelanggan mengenai software yang dibuat lebih awal. Selanjutnya menghemat waktu karena pembangunan system dibuat lebih cepat, dan mengurangi resiko kegagalan implementasi software dari segi non teknis sehingga kerugian dari segi materi relative lebih kecil.
2. UML manajemen perpustakaan



3. Saran untuk mengurangi biaya pada tahapan validasi dan evaluasi pembangunan software

### Software Test Life Cycle

Berlawanan dengan kepercayaan populer, Pengujian Perangkat Lunak bukan hanya aktivitas tunggal. Kegiatan ini terdiri dari serangkaian kegiatan yang dilakukan secara metodologis untuk membantu mengesahkan produk perangkat lunak Anda. Kegiatan-kegiatan ini (tahapan) merupakan Software Test Life Cycle (STLC). Gambar 2 menunjukkan contoh model pengujian yaitu V-Model of Testing. Model pengujian yang lainnya pun banyak, menyesuaikan pada model pengembangan perangkat lunak yang digunakan.



Gambar 2 Flow V-Model of Testing

#### 1. Requirement Analysis

selama fase ini, tim uji mempelajari requirement dari sudut pandang pengujian untuk mengidentifikasi requirement yang dapat diuji. Tim QA dapat berinteraksi dengan berbagai pemangku kepentingan (Client, Business Analyst, Technical Leads, System Architects, dll) untuk memahami requirement secara terperinci. Requirement dapat berupa Fungsional (menentukan apa yang harus dilakukan perangkat lunak) atau Non Fungsional (mendefinisikan kinerja sistem / ketersediaan keamanan). Kelayakan otomatisasi untuk proyek pengujian yang

diberikan juga dilakukan pada tahap ini. Aktivitas yang dilakukan pada tahap requirement analysis antara lain:

Mengidentifikasi jenis tes yang akan dilakukan.

Mengumpulkan secara detail tentang prioritas dan fokus pengujian.

Menyiapkan Requirement Traceability Matrix (RTM).

Mengidentifikasi rincian lingkungan pengujian di mana pengujian seharusnya dilakukan.

analisis kelayakan otomatisasi (jika diperlukan).

Tahap requirement analysis ini menghasilkan RTM dan dokumen analisis kelayakan otomatisasi.

## 2. Test Planning

Fase ini juga disebut fase Strategi Tes. Biasanya, dalam tahap ini, seorang manajer senior QA akan menentukan perkiraan upaya dan biaya untuk proyek dan akan menyiapkan dan menyelesaikan Rencana Tes. Aktivitas yang dilakukan pada tahap requirement analysis antara lain:

Persiapan dokumen rencana uji / strategi untuk berbagai jenis pengujian

Pemilihan alat uji

Estimasi upaya pengujian

Perencanaan sumber daya dan menentukan peran dan tanggung jawab

Kebutuhan pelatihan

Tahap test planning ini menghasilkan dokumen rencana uji / strategi dan dokumen estimasi upaya pengujian.

## 3. Test Case Development

Tahap ini melibatkan pembuatan, verifikasi, dan pengerjaan ulang kasus uji dan skrip pengujian. Data uji, diidentifikasi / dibuat dan ditinjau dan kemudian dikerjakan ulang juga. Aktivitas yang dilakukan pada tahap test case development antara lain:

Membuat test case, skrip otomatisasi (jika ada)

Meninjau tes case dan skrip

Membuat data uji (jika uji lingkungan tersedia)

Tahap test case development ini menghasilkan test case/script dan test data.

## 4. Test Environment Setup

Lingkungan pengujian menentukan kondisi perangkat lunak dan perangkat keras tempat work product diuji. Pengaturan lingkungan pengujian adalah salah satu aspek penting dari proses

pengujian dan dapat dilakukan secara paralel dengan Tahap Pengembangan Test Case. Tim uji mungkin tidak terlibat dalam kegiatan ini jika pelanggan / tim pengembangan menyediakan lingkungan pengujian, dalam hal ini tim uji diminta untuk melakukan pemeriksaan kesiapan (smoke testing) dari lingkungan yang diberikan. Aktivitas yang dilakukan pada tahap test environment setup antara lain:

Memahami arsitektur yang diperlukan, pengaturan environment, dan menyiapkan daftar requirement perangkat keras dan perangkat lunak untuk Test Environment.

Pengaturan test environment dan test data.

Lakukan smoke test dalam pembangunan.

Tahap test environment ini menghasilkan environment yang siap dengan pengaturan test data dan hasil smoke test.

## 5. Test Execution

Selama fase ini tim uji akan melakukan pengujian berdasarkan rencana pengujian dan kasus uji yang disiapkan. Bug akan dilaporkan kembali ke tim pengembangan untuk koreksi dan pengujian ulang akan dilakukan. Aktivitas yang dilakukan pada tahap test execution antara lain:

Jalankan tes sesuai rencana

Dokumentasikan hasil tes, dan log cacat untuk kasus yang gagal

Petakan defect ke test case dalam RTM

Tes ulang perbaikan defect

Lacak defect yang akan ditutup

## 6. Test Cycle Closure

Tim pengujian akan bertemu, membahas dan menganalisis artefak pengujian untuk mengidentifikasi strategi yang harus diterapkan di masa depan, mengambil pelajaran dari siklus pengujian saat ini. Idanya adalah untuk menghilangkan hambatan proses untuk siklus pengujian di masa depan.

Kesimpulan

Agar hemat biaya, pengujian harus dikonsentrasikan pada area yang paling efektif. Tidak adanya kebijakan pengujian organisasi dapat mengakibatkan terlalu banyak usaha dan uang yang akan dihabiskan untuk pengujian, sehingga berusaha untuk mencapai tingkat kualitas yang tidak mungkin atau tidak perlu. Testing juga dipengaruhi oleh people, cost, organization, dan measure.

## Advanced is analysis and design

Nama : Nizar Firliansa

Kelas : Magister teknik informatika R1.

### 1. Lebih cenderung ke *Plain- Drie Deelopment*

Alasan :

1. Karna dapat memberikan perangkat lunak yang nyata dan berfungsi berulang kali secara tepat waktu sesuai dengan prinsip.
2. Perencanaan, penjadwalan, dan pelacakan proyek didorong oleh hierarki fitur, dan bukannya kumpulan tugas rekayasa perangkat lunak yang diadopsi secara sewenang-wenang.
3. Sebagai model proses praktis untuk rekayasa perangkat lunak berorientasi objek.
4. Menjamin kualitas perangkat lunak dengan mendorong strategi pengembangan tambahan, penggunaan desain dan inspeksi kode, penerapan audit jaminan kualitas perangkat lunak, pengumpulan metrik, dan penggunaan pola (untuk analisis, desain, dan konstruksi).

### 2. Contoh desain Class object

Objek Manusia.	
	
Atribut :	Method :
- Umur	- Jalan
- Tinggi	- Lari
- Berat	- Lompat.

3. Saran untuk mengurangi biaya pada tahapan validasi dan evaluasi sebuah pembangunan software adalah dengan Perbaikan proses.

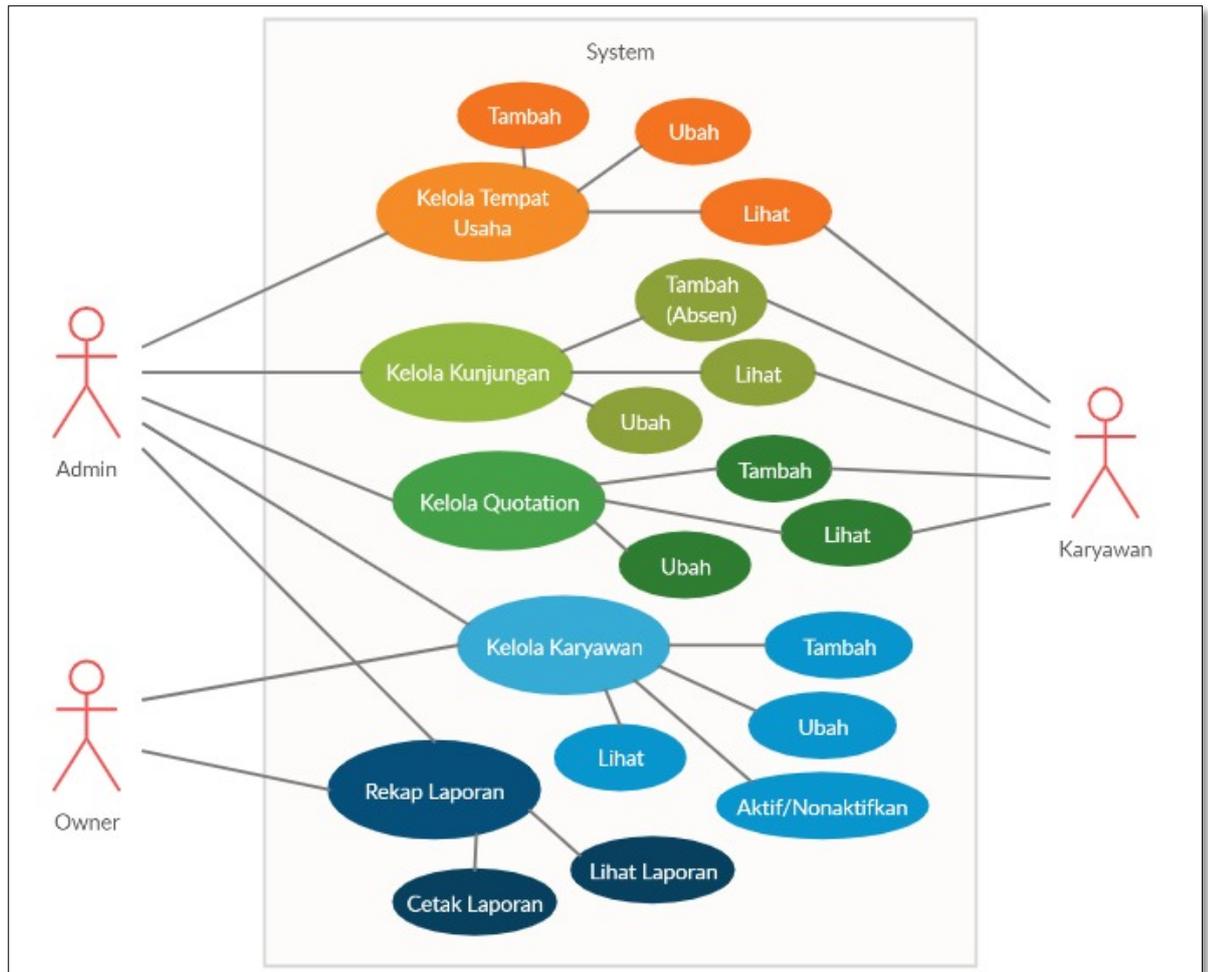
Perbaikan proses adalah Penerapan praktik-praktik secara konsisten dan mengubah praktik-praktik yang menjadi penyebab masalah. Tujuan dari perbaikan proses pengembangan perangkat lunak adalah untuk memahami proses yang ada saat ini, memperkenalkan perubahan proses untuk meningkatkan kualitas, mengurangi biaya, atau mempercepat jadwal, tuntutan industri terhadap peningkatan kualitas produksinya, dan untuk mengurangi resiko cacat perangkat lunak yang diakibatkan dari kesalahan yang terjadi saat proses pengembangan.

**NOVITA ANGGRAINI**  
**192420025**  
**ADVANCE ANALISIS & DESIGN**

---

1. Tentukan pilihan anda sendiri, apakah lebih cenderung ke plan drive development ataukah agile development? Jelaskan alasan anda
  - ⇒ Agile Development, alasan saya memilih adalah karena agile development merupakan method yang flexible. Artinya suatu project resiko mengalami kegagalan jika produk yang dipesan tidak sesuai dengan yang diharapkan konsumen. oleh karena itu perlunya ada komunikasi antar team & konsumen untuk membangun suatu relasi yang intim. Perubahan yang terjadi apabila adanya kesalahan atau error jika diketahui terlambat maka akan berpengaruh pada waktu dan cost. Agile development adalah method yang bertujuan untuk memberikan proses komunikatif dimana setiap unit dikerjakan dengan sistem 1 pengerjaan unit > feedback > maintenance > development > kemudian re-delivery kembali hingga dinyatakan diterima. Maka hal ini lebih efisien untuk menekan biaya maintenance dari pada semua unit selesai barulah maintenance. Kemungkinan error atau kesalahan produk sesuai dengan pemesanan konsumen itu lebih kecil dari pada plain drive yang memang method dengan produk skala besar. Agile bisa dilakukan dengan dukungan team solid dan kostumer yang bisa diajak komunikatif. Hal ini serupa dengan satu saja perwakilan kostumer yang bisa diajak komunikasi intens, dimana dia mengerti akan kebutuhan system. Maka agile method adalah alasan saya memilihnya dimana ini adalah method yang paling flexible untuk sebuah projek.

2. Buatlah satu design class object yang terlibat didalam sebuah component process menggunakan UML notation  
 ⇒ System sales call



3. Apa semua saran anda untuk mengurangi biaya pada tahapan validasi dan evaluasi sebuah pembangunan software?  
 ⇒ Solusinya adalah unit testing. Unit testing diadakan karena bertujuan untuk meminimalisirkan pembengkakan biaya dan waktu pembangunan sebuah system, hal ini berkaitan dengan bagaimana atau apakah pada setiap unit kebutuhan dari system telah terpenuhi atau belum. Adanya unit testing ini dapat menangkap error / kesalahan lebih cepat sehingga proses maintenance dapat dilakukan sesegera mungkin dari pada semua unit selesai lalu error ditemukan belakangan malah akan membebani dengan perubahan cukup besar atau perlu adanya maintenance yang besar hal ini mempengaruhi penggunaan biaya dan waktu. Unit testing adalah solusi yang digunakan untuk memberikan kemungkinan produk tidak gagal dan sesuai dengan pesanan konsumen.

Nama : Rachmat Akbar  
Kelas : MTI AR2  
Nim : 192420036

1. Tentukan pilihan anda sendiri, apakah lebih cenderung ke plan-driven development ataukah agile development ? jelaskan alasan anda !
2. buatlah satu design class object yang terlibat dalam sebuah component process menggunakan UML notation!
3. Apa saran anda untuk mengurangi biaya pada tahapan validasi dan evaluasi sebuah pembangunan software

JAWAB :

1. Saya akan cenderung menggunakan metode *agile development* , hal ini dikarenakan *agile development* mempunyai banyak keuntungan yang menurut saya cocok diterapkan pada masa sekarang. Hal-hal yang mendasari saya memilih *agile development* adalah :
  - a. Keterlibatan *Stakeholder*

Agile memberikan banyak kesempatan dalam keterlibatan stakeholder dan team - sebelum, selama dan setelah setiap Sprint. Dengan melibatkan client di setiap langkah proyek, ada kolaborasi yang tinggi antara client dan team proyek. Hal ini juga memberi lebih banyak kesempatan bagi team untuk benar-benar memahami visi client. Mempresentasikan software yang bekerja dengan lebih awal dan sering meningkatkan kepercayaan para stakeholder dalam kemampuan team untuk membuat software berkualitas tinggi dan mendorong mereka untuk lebih terlibat dalam proyek ini.
  - b. Transparansi

Pendekatan Agile memberikan kesempatan unik bagi client untuk lebih terlibat dalam keseluruhan proyek. Ini beralih dari kemampuan memprioritaskan fitur untuk merencanakan dan meninjau pembaruan software yang berisi fitur baru.
  - c. Pengiriman awal dan mudah diprediksi

Dengan menggunakan jadwal tetap Sprints 1-4 minggu, fitur baru bisa dibuat dengan cepat dan sering, juga dengan tingkat prediktabilitas yang tinggi. Ini juga memberi kesempatan untuk memasang atau menguji beta dari software lebih awal dari yang direncanakan jika ada nilai bisnis yang cukup.
  - d. Biaya dan jadwal yang mudah ditebak

Karena setiap Sprint memiliki durasi yang tetap, biayanya dapat diprediksi dan sudah ditentukan di dalam pekerjaan yang akan dilakukan dalam jangka waktu yang ada membuat mudahnya perkiraan biaya yang ada. Dikombinasikan dengan perkiraan yang diberikan kepada client sebelum setiap Sprint, client dapat lebih mudah

memahami biaya setiap fitur yang meningkatkan kemampuan pengambilan keputusan mengenai prioritas setiap fitur.

e. Memungkinkan perubahan

Sementara team perlu tetap fokus untuk memberikan subset fitur produk yang disepakati di setiap Sprint, ada kesempatan untuk terus memperbaiki dan memprioritaskan isi backlog produk. Item backlog baru atau yang telah diubah dapat direncanakan dalam pertemuan Sprint berikutnya, memberikan kesempatan untuk mengenalkan perubahan dalam beberapa minggu.

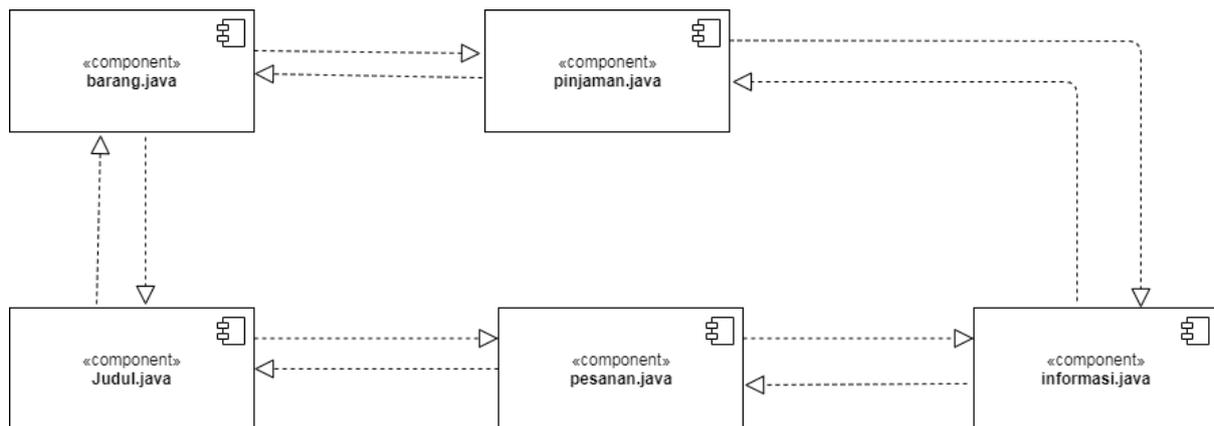
f. Fokus kepada stakeholder

Agile biasanya menggunakan feedback user dengan kriteria yang berfokus pada bisnis untuk menentukan fitur produk. Dengan memfokuskan fitur pada kebutuhan pengguna nyata, setiap fitur memberikan nilai, tidak hanya sekedar komponen IT. Ini juga memberikan kesempatan untuk menguji beta software setelah setiap Sprint, mendapatkan umpan balik yang berharga di awal proyek dan memberikan kemampuan untuk melakukan perubahan seperlunya.

g. Meningkatkan kualitas

Dengan membagi proyek menjadi unit-unit yang dapat dikelola, team proyek dapat berfokus pada pengembangan, pengujian dan kolaborasi berkualitas tinggi. Selain itu, dengan memproduksi sering membangun dan melakukan pengujian dan ulasan selama setiap Sprint, kualitas ditingkatkan dengan menemukan dan memperbaiki cacat dengan cepat dan mengidentifikasi kesalahan ekspektasi.

## 2. Contoh komponen diagram



## 3. Salah satu upaya yang dapat dilakukan untuk mengurangi biaya pada tahapan validasi dan evaluasi dalam pembangunan software adalah melakukan Test secara lebih awal dan berulang pada setiap tahapan daripada hanya dilakukan pada saat akhir. Hal ini juga yang mendasari saya memilih *agile development* karena pada setiap *sprint* telah dilakukan validasi dan evaluasi sehingga akan mengurangi *cost* pada saat validasi dan evaluasi akhir software yang dibuat

**Nama : Ria Aprinda**

**NIM : 192420022**

**Kelas : MTI 21 A R1**

**Mata Kuliah : Advanced IS Analysis and Design**

1. Tentukan pilihan anda sendiri, apakah lebih cenderung ke Plan-Drive Development ataukah Agile Development? Jelaskan alasan anda!

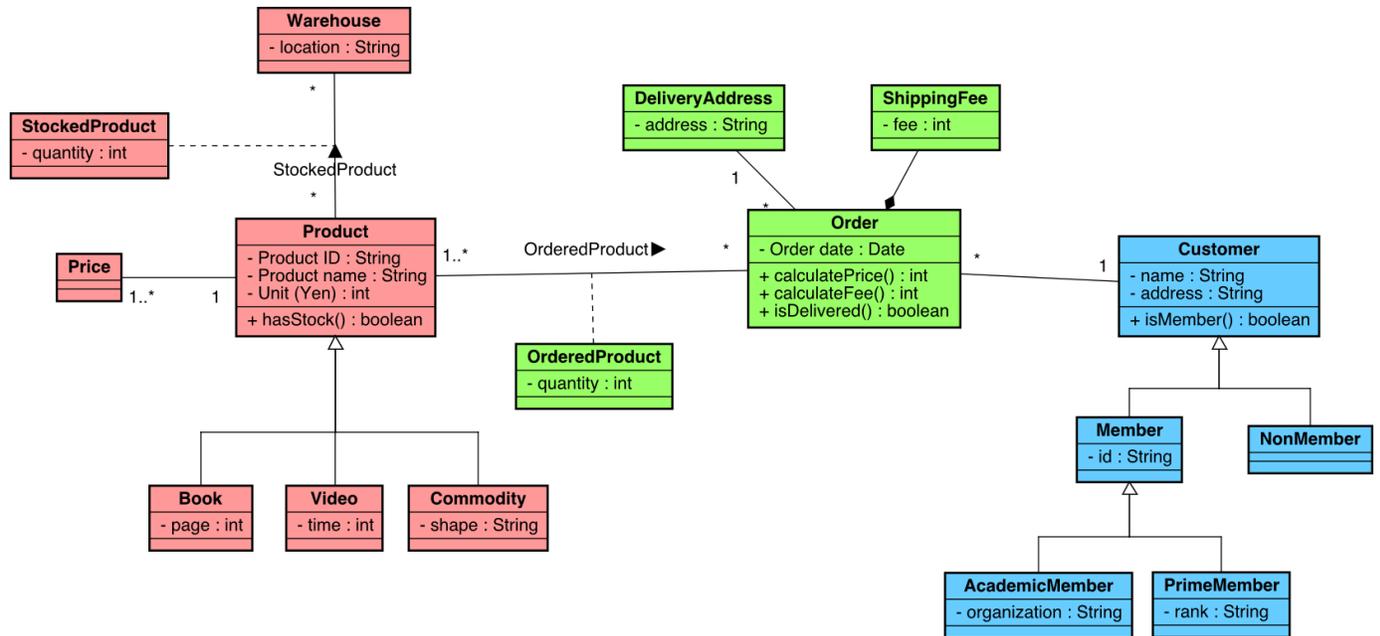
**Jawab :**

Plan Driven Development berfokus pada BDUF (Big Design Up Front) – dimana kita memiliki suatu desain besar, rencana komprehensif yang dilakukan, disepakati, dan disegel antara kita dan pelanggan. Kemungkinan adanya perubahan minimal, siklus feedback minimal yang menghasilkan closure yang menantang. Di sisi lain, Agile mengikuti EDUF (Just Enough Design Upfront) - ini memungkinkan kita untuk lebih fokus pada kebutuhan langsung, kita menyesuaikan perencanaan (perencanaan adaptif). Secara keseluruhan, dengan Agile kita melakukan perencanaan terus-menerus dibandingkan hanya datang dengan suatu rencana besar, sekali saja. Seiring berjalannya waktu, dengan Agile kita terus merencanakan dan beradaptasi dengan situasi yang berubah.

Plan Driven Development baik ketika kita dapat memprediksi kebutuhan dan perkembangan dengan baik sebelumnya, tetapi kenyataannya di banyak lingkungan bisnis rencana terus berubah dan kita membutuhkan pendekatan yang membantu kita dalam melakukan berbagai hal secara progresif. Agile cocok dengan konteks itu. Jadi, saya lebih cenderung ke Agile Development.

2. Buatlah satu desain class object yang terlibat di dalam sebuah component process menggunakan UML notation!

**Jawab :**



3. Apa saran anda untuk mengurangi biaya pada tahapan validasi dan evaluasi sebuah pembangunan software ?

**Jawab :**

Saran saya dengan melakukan pengujian software berupa unit testing, pengujian integrasi, pengujian sistem, dan pengujian penerimaan yang fungsinya untuk menghindari *bug* atau kesalahan pada proses-proses tertentu. Hal ini dapat menjaga kualitas software yang dibangun supaya dapat bertahan diantara software yang lain yang terus berkembang mengefektifkan biaya agar tidak banyak membuang software karena kegagalan pemasaran atau kegagalan produksi.

**Pertanyaan :**

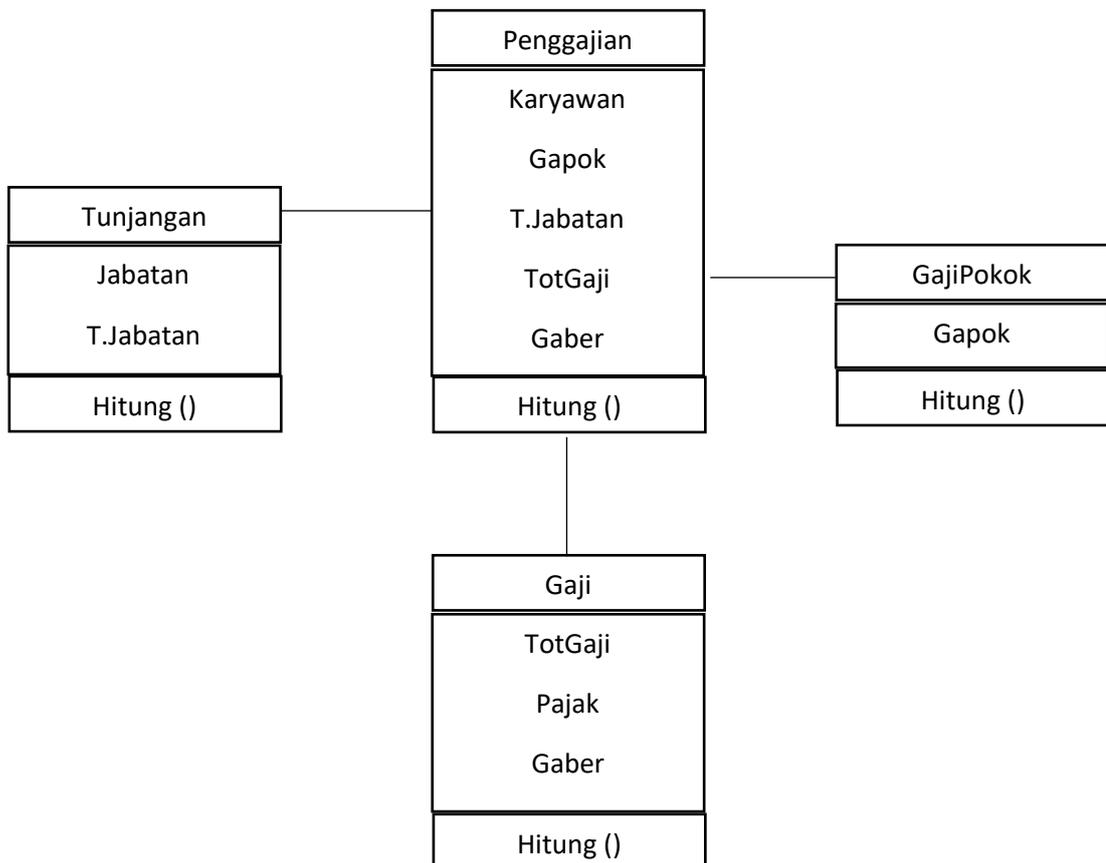
1. Tentukan pilihan anda sendiri, apakah lebih cenderung ke *Plain-Drive Development* ataukah *Agile Development*? Jelaskan alasan anda!
2. Buatlah satu desain *class object* yang terlibat di dalam sebuah *component process* menggunakan UML notaion!
3. Apa saran anda untuk mengurangi biaya pada tahapan validasi dan evaluasi sebuah pembangunan *software* ?

**Jawaban :**

1. Pilihan saya cenderung memilih *Agile Development*, Karna menurut saya *Agile Development* menjadi perhatian yang baik bagi pengembang perangkat lunak yang apabila suatu proses yang akan dibangun belum selesai tetapi boleh lanjut ke proses lainnya. Dan *Agile Development* lebih melibatkan *client* agar apabila setiap proses ada perbuahan dalam melakuai proses tahapannya maka akan bias diubah sesuai dengan apa yang diminta oleh *client*, ini membantu *programmer* untuk tetap bisa *maintenance* suatu system yang akan dibangun nantinya. Kemudian proses yang dilaluinya mempercepat penyelesaian. Dan juga *Agile Development* selalu mengikuti *trend* IT serta berfokus pada *coding/* Implementasi dari pada desain.
2. *Component process* : Gaji Karyawan → 

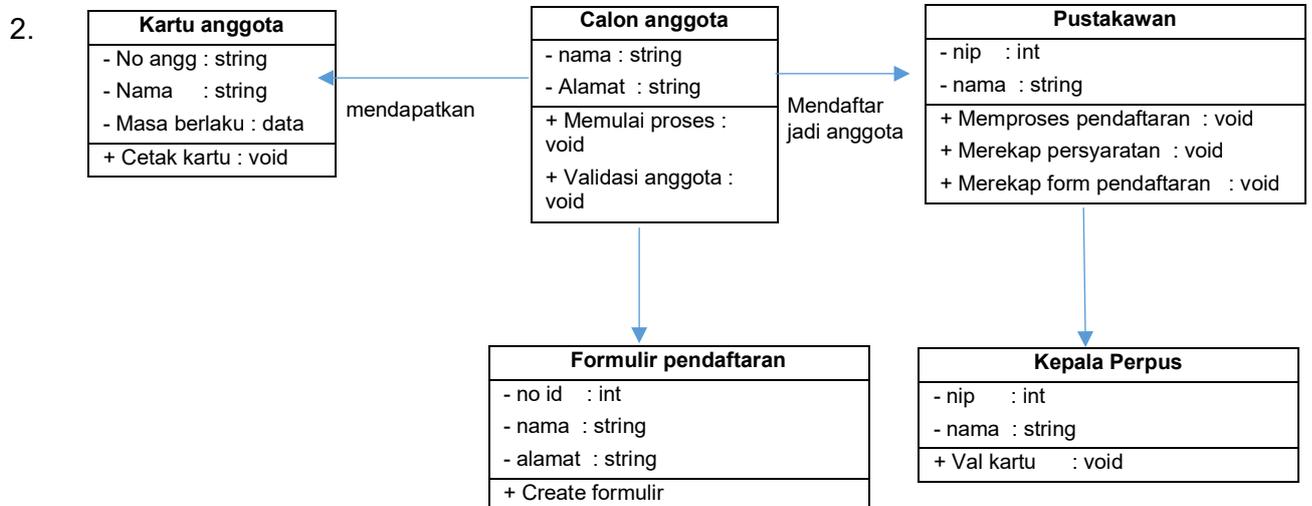
Penggajian
------------

 → OOP



3. Saran saya untuk mengurangi biaya tahapan validasi dan evaluasi yaitu dengan cara menentukan dengan baik metode apa yang akan digunakan dalam pengembangan atau pembangunan suatu system nantinya agar tahu dan dapat memastikan serta memilih apa saja yang diperlukan atau yang tidak diperlukan dalam pengembangan atau pembangunan suatu system. Misalnya dalam persentase biayanya yaitu 60% *Development cost* dan 40% *Testing cost*.

1. Metode Agile cocok untuk proyek yang memberikan kepuasan pelanggan karena resiko kegagalan yang kecil dan lebih cepat dalam penyelesaian software serta perubahan yang cepat dari software yang dibangun untuk meningkatkan kualitas software yang lebih baik. Alasannya dikarenakan Agile memiliki beberapa Methods yaitu :
  - a. Acceptance Test Driven Development (ATDD)
  - b. Agile Modeling
  - c. Adaptive Software Development (ASD)
  - d. Agile Unified Process (AUP)
  - e. Continuous integration (CI)
  - f. Crystal Clear
  - g. Crystal Methods
  - h. Dynamic Systems Development Method (DSDM)
  - i. Extreme Programming (XP)
  - j. Feature Driven Development (FDD)
  - k. Graphical System Design (GSD)
  - l. Kanban
  - m. Lean software development
  - n. Rational Unified Process (RUP)
  - o. Scrum
  - p. Scrum-ban
  - q. Story-driven modeling
  - r. Test-driven development (TDD)
  - s. Velocity tracking
  - t. Software Development Rhythms



3. Pengurangan biaya pada tahapan validasi akan berakibat kegagalan dalam pengujian

Nama : Ryan Andrian

NIM : 192420006

Kelas : MTI A R1

Mata Kuliah : Advanced IS Analysis and Design

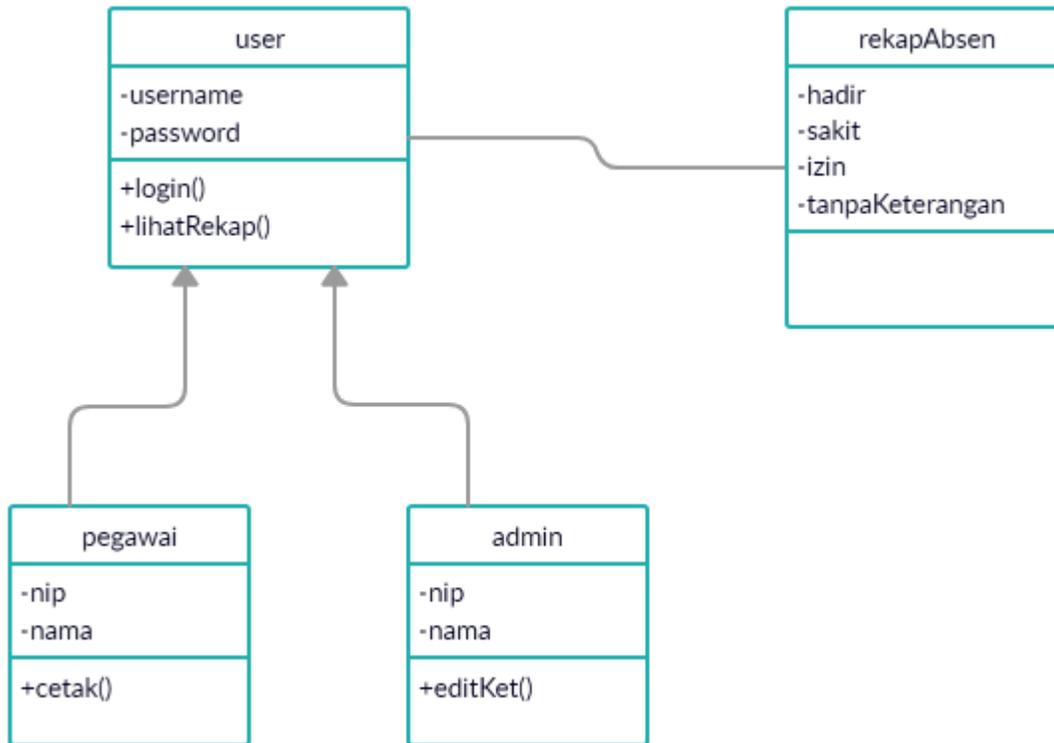
1. Tentukan pilihan anda sendiri, apakah lebih cenderung ke Plain-Drive Development ataukah Agile Development? Jelaskan alasan anda!

Jawab :

lebih cenderung ke Agile Development karena dalam proses pengembangannya lebih dinamis dan memungkinkan user untuk segera mencoba menggunakan *initial release* (versi beta) terlebih dahulu untuk dapat dievaluasi oleh user. Kemudian apabila user menginginkan penambahan / perubahan kebutuhan (fitur), dengan Agile Development dapat mengakomodir keinginan user tersebut dengan waktu yang relatif singkat dibandingkan Plan Driven Development. Disamping itu, di zaman sekarang dimana kompetisi semakin ketat, efisiensi biaya dan hasil maksimal dibutuhkan dalam waktu singkat sehingga Agile Development lebih cocok digunakan walaupun tidak tertutup kemungkinan Plan Driven Development masih tetap digunakan untuk kondisi tertentu (misal pada proses pengembangan tidak begitu mempermasalahkan segi biaya dan waktu penyelesaian yang relatif lebih panjang).

2. Buatlah satu desain class object yang terlibat di dalam sebuah component process menggunakan UML notaion!

Jawab :



3. Apa saran anda untuk mengurangi biaya pada tahapan validasi dan evaluasi sebuah pembangunan software ?

Jawab : melakukan uji coba (beta testing) untuk umum

Nama : Sapardi  
Nim : 192420026  
Kelas : MTI

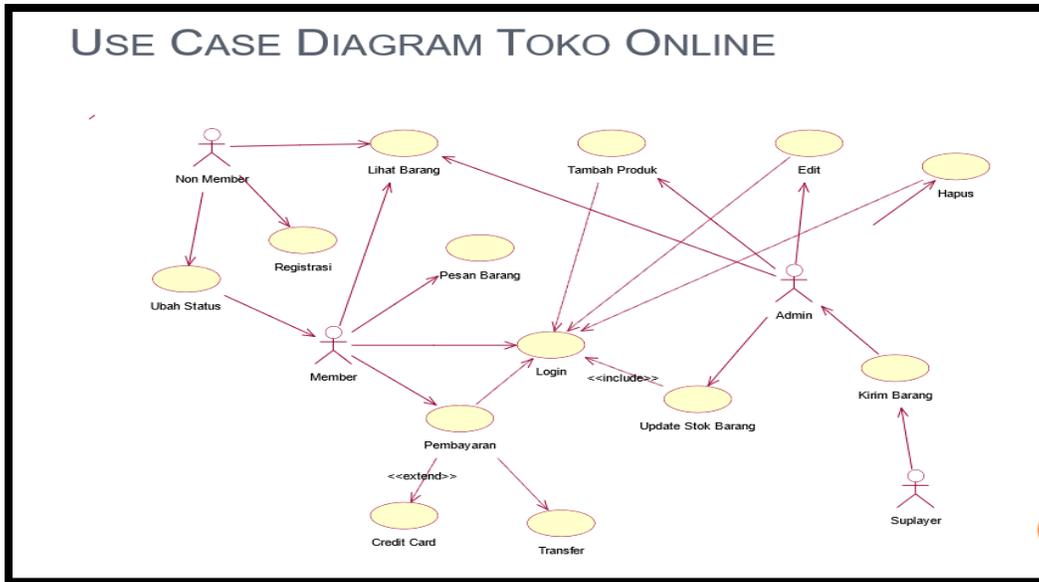
1. apakah lebih cenderung ke plain drive Development atau Agile Development ?

A. Saya lebih cenderung memakai Agile Development karna:

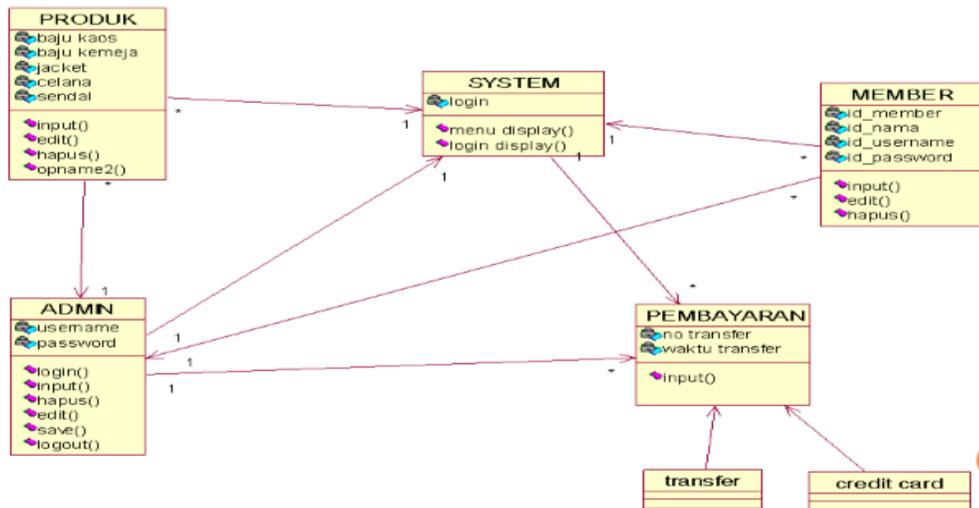
Menurut *Agile Alliance*, terdapat 12 prinsip yang harus dipenuhi bagi mereka yang ingin berhasil dalam penerapan *Agile Software Development*. Prinsip-prinsip tersebut antara lain :

1. Kepuasan klien adalah prioritas utama dengan menghasilkan [produk](#) lebih awal dan terus menerus.
2. Menerima perubahan kebutuhan, sekalipun diakhir pengembangan.
3. Penyerahan hasil/software dalam hitungan waktu dua minggu sampai dua bulan.
4. Bagian bisnis dan pembangun kerja sama tiap hari selama proyek berlangsung.
5. Membangun proyek dilindungi orang-orang yang bermotivasi tinggi yang bekerja dalam lingkungan yang mendukung dan yang dipercaya untuk dapat menyelesaikan proyek.
6. [Komunikasi](#) dengan berhadapan langsung adalah komunikasi yang efektif dan efisien.
7. Software yang berfungsi adalah ukuran utama dari kemajuan proyek.
8. Dukungan yang stabil dari sponsor, pembangun, dan pengguna diperlukan untuk menjaga perkembangan yang berkesinambungan.
9. Perhatian kepada kehebatan teknis dan desain yang bagus meningkatkan sifat agile.
10. Kesederhanaan sangat penting.
11. Arsitek. Kebutuhan dan desain yang bagus muncul dari tim yang mengatur dirinya sendiri.
12. Secara periodik tim evaluasi diri dan mencari cara untuk lebih efektif dan segera melakukannya.

B. Buatlah desain class Objek yang terlibat dalam sebuah komponentproses menggunakan UML Notation?



### CLASS DIAGRAM TOKO ONLINE



- C. Apa saran anda untuk mengurangi Biaya pada tahapan validasi dan evaluasi sebuah pembangunan Software?
- D. Pengurangan Bug pada system Software sehingga Bisa mengurangi Banyak nya tahap Testing
  - E. Dalam pengembangan system Software Harus Lebih Teliti Sehingga dapat menghindari banyak nya perubahan system Software
  - F. Sebelum ke tahap pengembangan harus lebih Teliti dalam membuat Kontark pembangunan Software ini berisikan Point –point dalam pengurangan biaya
  - G. Mengetahui kebutuhan hal terpenting dalam kebutuhan User

Nama : Sela Taramita

NIM : 192420038

“Tugas”

Soal :

1. Tentukan Pilihan anda sendiri, apakah lebih cenderung ke *Plain Drive Development* ataukah *Agile Development* ? Jelaskan alasan anda ?
2. Buatlah satu desain *class object* yang terlibat didalam sebuah *component process* menggunakan UML Notation !
3. Apa saran anda untuk mengurangi biaya pada tahapan validasi dan evaluasi sebuah pembangunan *software* ?

Jawaban :

1. *Plain Drive Development*

Dalam pengerjaan suatu proyek saya lebih memilih *Plain Drive Development* dikarenakan metodologi ini merupakan suatu metode yang lebih formal untuk membuat aplikasi. Dalam penggunaan Metodologi *Plain Drive Development* dalam suatu proyek dapat dilakukan pengulangan dan terprediksi, proses yang terorganisir, dokumen yang berkelanjutan, system muka arsitekturnya, mempunyai rencana yang terperinci, adanya proses pemantauan, pengendalian dan pengetahuan, manajemen resiko, verifikasi dan validasi.

Ada 3 hal yang paling menonjol dalam *Plain Drive Development* yaitu Proses Personal Software (PSP), Proses Perangkat Lunak dan Rational Unified Process (RUP).

Penggunaan metode *Plain Drive Development* cocok digunakan dalam berbagai hal, antara lain :

1. Jika customer lepas tangan terhadap pembuatan perangkat lunak, customer tersebut tidak mau ikut campur dalam pembuatannya, baik itu karena sibuk atau sebagainya
2. Anggota banyak (Timnya besar)
3. Analisis yang terperinci
4. Perangkat lunak tersebut dirancang agar bisa digunakan dalam waktu yang lama, tidak hanya 1 atau 2 tahun saja
5. Teknologi yang kurang mendukung
6. Tim pengembangannya bisa dilakukan dimana saja, tidak harus dalam satu lokasi

Untuk *Plain Drive Development* terdapat kelebihan dan kekurangannya yaitu :

Kelebihan :

1. Pembuatan program tersebut terencana, mulai dari spesifikasi, perancangan, validasi dan perubahan
2. Customer tidak perlu repot-repot untuk memikirkan pembuatan perangkat lunak tersebut karena Cuma membuat kontrak kebutuhan diawal saja

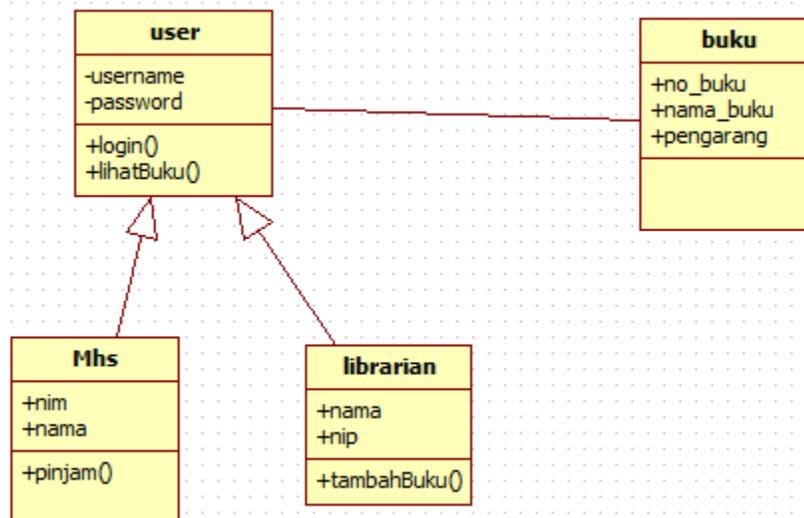
3. Perangkat lunak tersebut bisa digunakan dalam jangka waktu yang panjang sampai puluhan tahun
4. Bisa dikerjakan oleh tim dimana mereka tidak harus berkumpul dalam satu lokasi
5. Biaya relatif yang ringan karena pekerjaannya dilakukan berurutan
6. Hasilnya sangat bagus karena pengolahannya sangat terperinci

Kekurangan :

1. Tidak bisa dilakukan perubahan terhadap perangkat lunak ditengah-tengah pekerjaan sehingga tidak bisa menyesuaikan kemajuan dan keinginan masyarakat saat itu
2. Hasil nya kurang sesuai dengan kebutuhan dan keinginan masyarakat pada saat perangkat lunak tersebut selesai

Menurut saya walaupun ada beberapa kekurangan untuk menggunakan metode *Plain Drive Development* tetapi masih bisa diatasi dengan customer nya.

2. Class object didalam component proses dalam menggunakan UML Notation :  
Contoh Class Object :



3. Saran saya untuk mengurangi biaya pada tahapan validasi dan evaluasi sebuah pembangunan *software* :

Dalam pembangunan *software* untuk mengurangi biaya dalam tahap validasi yaitu :

1. Memastikan bahwa *software* yang akan digunakan sudah selesai sehingga tahap validasi nya tidak dilakukan berulang
2. Membentuk komite validasi yang bertanggung jawab dalam pelaksanaan validasinya
3. Membentuk *Validasi Master Plan* yaitu dokumen yang menguraikan (secara garis besar) pedoman pelaksanaan validasi dalam suatu *software* tersebut
4. Membuat validasi prosedur tetap, protocol serta laporan validasi

5. Pelaksanaan validasi yang jelas
6. Melaksanakan peninjauan produk pada software tersebut

Dalam pembangunan *software* untuk mengurangi biaya dalam tahap evaluasi yaitu :

1. Memastikan informasi yang valid tentang *software* yang akan digunakan
2. Memastikan strategi yang digunakan apakah telah benar
3. Ketepatan pemilihan cara operasi *software*
4. Pemilihan cara penggunaan yang lebih baik pada *software* tersebut
5. Pelaksanaan pengawasan terhadap kegiatan rutin yang sedang berjalan serta pengawasan untuk mengumpulkan informasi terhadap keluaran/hasil dan indikator yang digunakan untuk mengukur kinerja program
6. Pelaksanaan evaluasi dilaksanakan secara periodic dan berkala, dapat bersifat internal dan eksternal atau partisipatif sebagai umpan balik periodic.

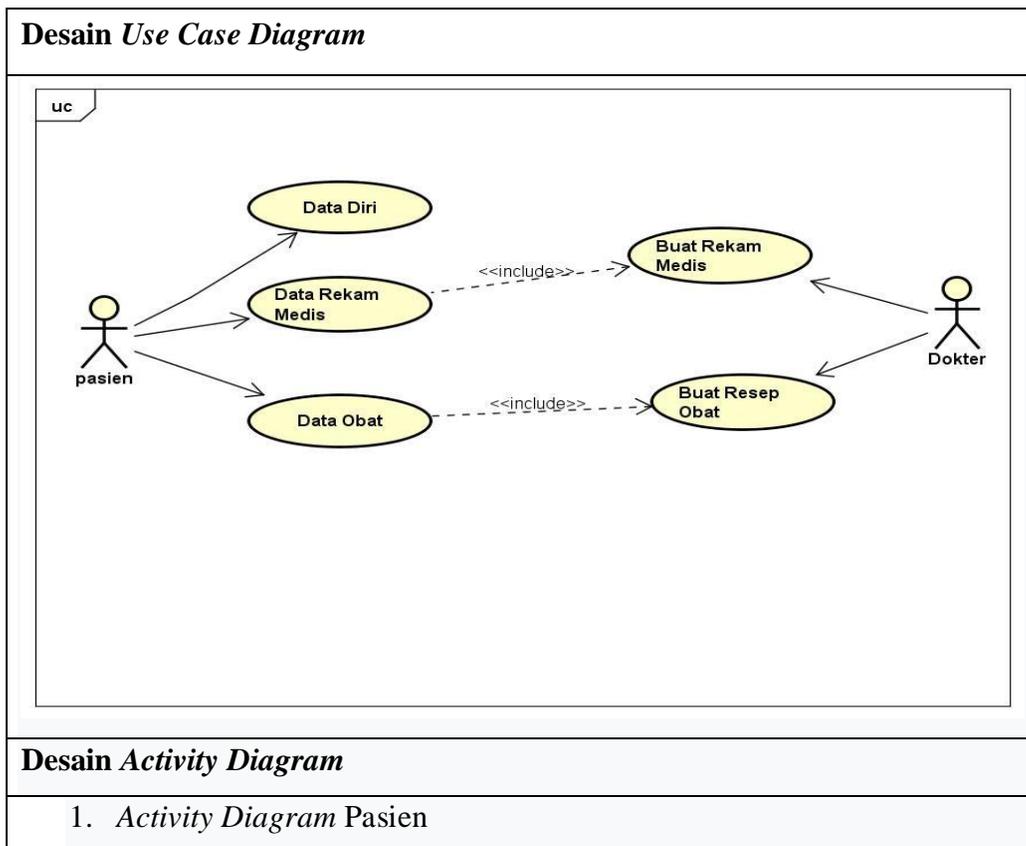
Nama : sulistiyani

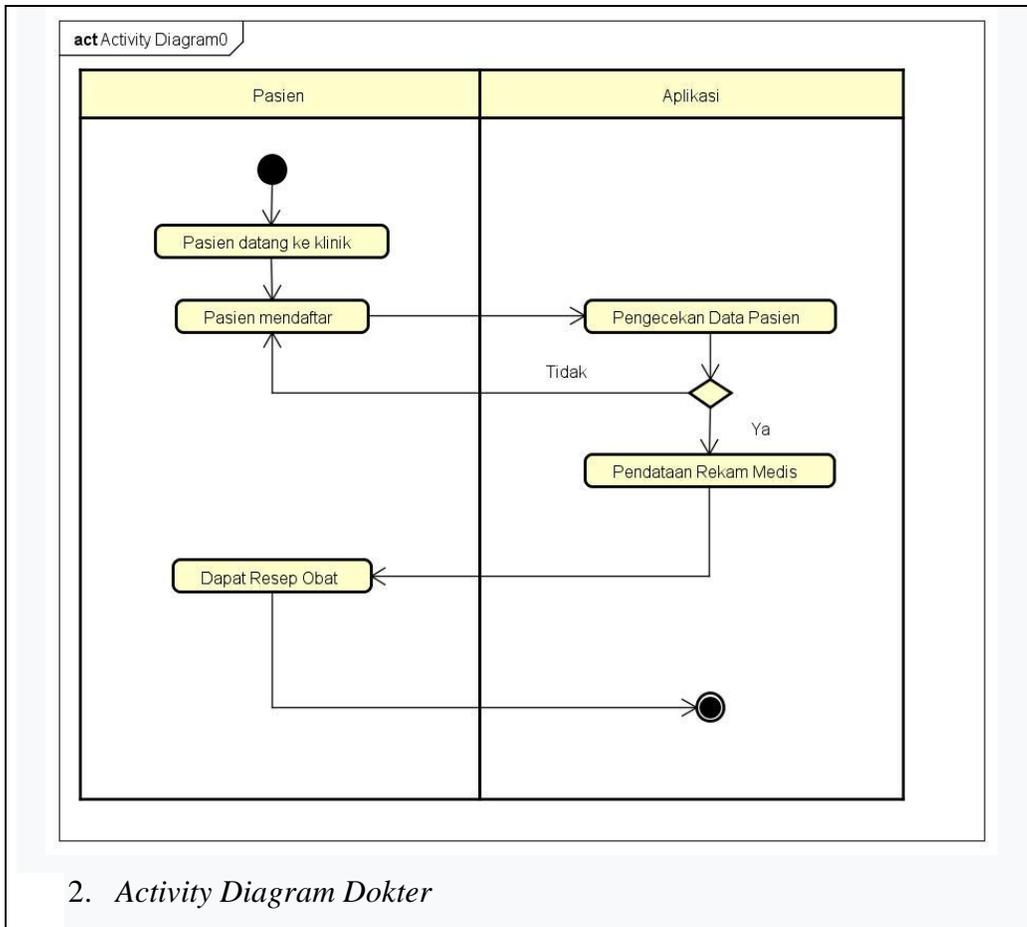
Mk : advanced is analysis and design

1. Saya lebih cenderung ke Agile Development methods dikarenakan metode ini memiliki keuntungan berupa:

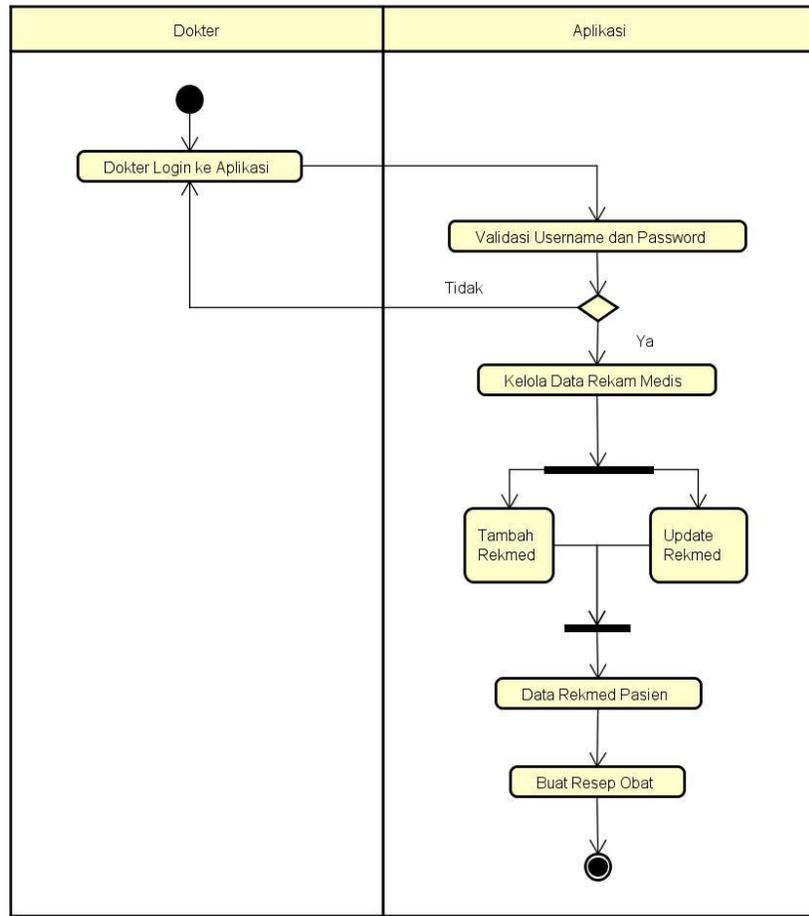
- Menambah produktivitas tim
- Menambah kualitas dari perangkat lunak
- Menambah kepuasan dari klien
- Menghemat biaya

2.

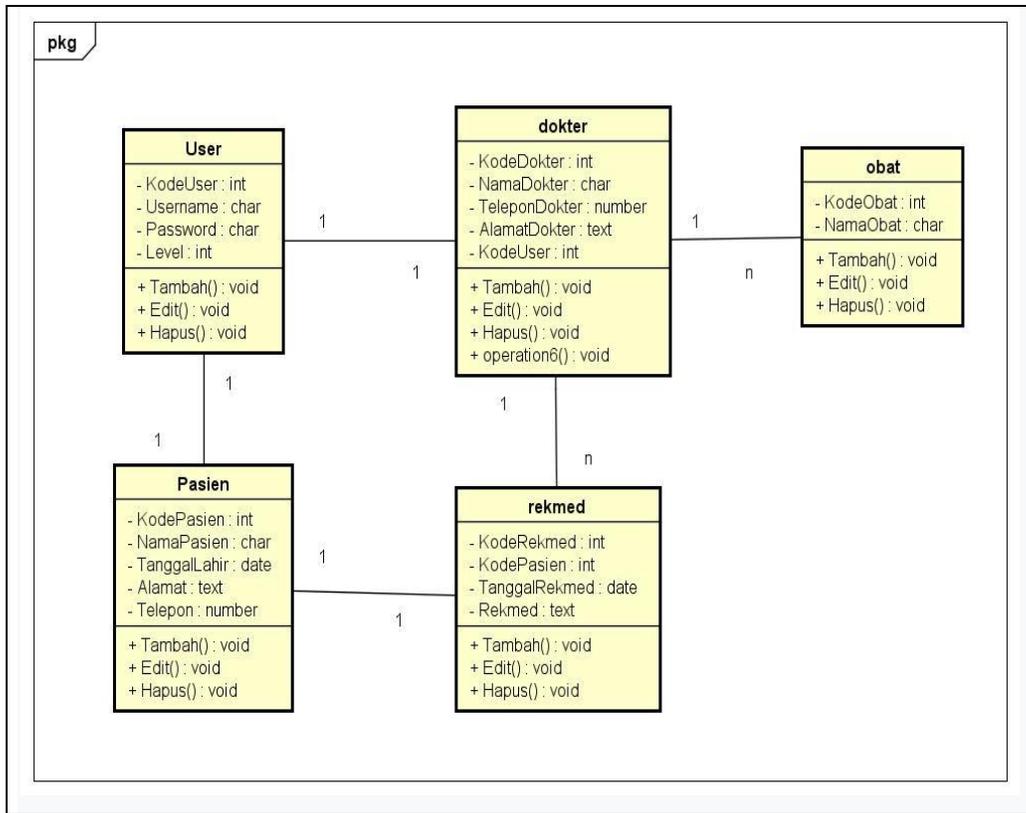




act Activity Diagram1



*Class Diagram*



- Untuk mengurangi biaya pada tahapan validasi dengan menggunakan software testing salah satunya yaitu pendekatan pengujian daur hidup.

Cara mengurangi biaya akibat kesalahan adalah menempatkan kesalahan-kesalahan tersebut 'lebih awal' di dalam daur hidup pengembangan. Artinya pengembang harus melakukan pengujian sedini mungkin untuk secepatnya menemukan kesalahan. Ilustrasi pengujian daur hidup pengembangan sistem dapat dilihat pada Buku William Perry hlm, 26.

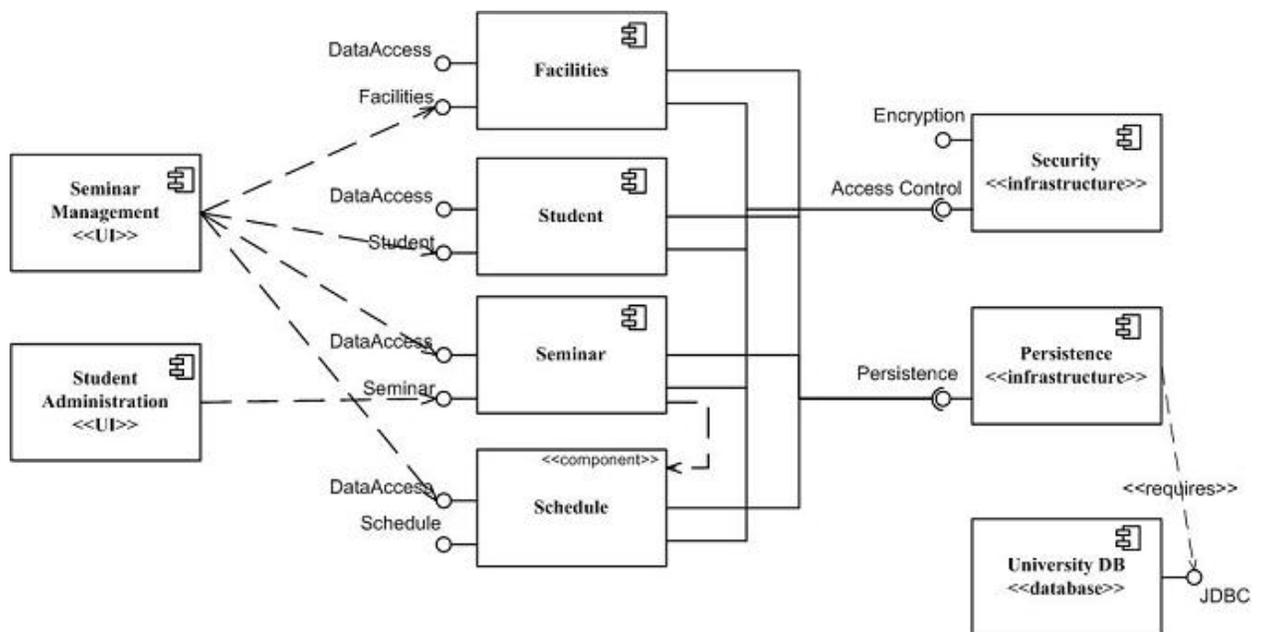
Nama : Suriani

NIM : 192420011

1. Saya lebih memilih plan drive development karena Metode plan driven cocok digunakan untuk berbagai hal, antara lain :

- Jika customer lepas tangan terhadap pembuatan perangkat lunak, customer tersebut tidak mau ikut campur dalam pembuatannya, baik itu karena alasan sibuk, malas dan sebagainya.
- Timnya besar (terdiri dari banyak orang).
- Analisisnya besar.
- Perangkat lunak tersebut dirancang agar bisa digunakan dalam tempo yang lama, tidak hanya 1 atau 2 tahun saja.
- Teknologi yang tersedia kurang mendukung.
- Tim pengembangnya tersebar dimana-mana, tidak harus kumpul satu lokasi agar bisa membuat suatu perangkat lunak

2. desain component



3. Saran saya untuk mengurangi biaya pada tahap validasi dan evaluasi sebuah pembangunan software, sebelum pembangunan software harus ada analisis kebutuhan yang bertujuan

1. Mengelola hasil elistasi kebutuhan untuk menghasilkan dokumen spesifikasi kebutuhan yang isi keseluruhannya sesuai dengan apa yang diinginkan pengguna.

2. Mengembangkan persyaratan kualitas yang memadai dan rinci, dimana para manajer dapat membuat pekerjaan proyek yang realistis dan staf teknis dapat melanjutkan dengan perancangan, implementasi dan pengujian.
3. Membangun pemahaman tentang karakteristik ranah permasalahan dan sekumpulan kebutuhan untuk menemukan solusi.

Ketiga tujuan tersebut dapat dicapai oleh perencana kebutuhan dengan melalui serangkaian tahapan-tahapan aktivitas. Tahapan aktivitas tersebut ialah **Tahap Analisis Kebutuhan dimana** pengumpulan kebutuhan-kebutuhan dari semua elemen sistem perangkat lunak yang akan di bangun. Pada tahap ini dibentuk spesifikasi kebutuhan perangkat lunak, fungsi perangkat lunak yang dibutuhkan, performansi (unjuk kerja) sistem perangkat lunak, penjadwalan proyek, identifikasi sumber daya (manusia , perangkat keras dan perangkat lunak yang dibutuhkan) dan taksiran biaya pengembangan perangkat lunak.

Apabila semua kebutuhan telah terperinci, maka proses pembangunan software dan study kelayakan berjalan sesuai dengan kebutuhan, maka tidak ada hambatan pada saat validasi software dimana semua pembuktian dari proses, prosedur, kegiatan, pelengkapan yang di inginkan oleh pengguna sudah sesuai dengan kebutuhan pengguna dengan cara ini dapat meminimalisir biaya validasi dan evaluasi terhadap suatu pembangunan software.

# The assignment of Adv. Is Analysis and Design From Mr. Tri Basuki Kurniawan, S.Kom., M.Eng., Ph.D.

**Yudy Pranata**  
**AR2 192420001**  
**Magister Teknik Informatika**  
**Universitas Bina Darma**

## **Quiz 1**

- 1. Tentukan pilihan anda sendiri, apakah lebih cenderung ke plan-driven development ataukah agile development? jelaskan alasan anda !**
- 2. buatlah satu design class object yang terlibat dalam sebuah component process menggunakan UML notation!**
- 3. Apa saran anda untuk mengurangi biaya pada tahapan validasi dan evaluasi sebuah pembangunan software!**

## **Jawaban**

1. Agile development, alasannya

### **KETERLIBATAN STAKEHOLDER**

Agile memberikan banyak kesempatan dalam keterlibatan stakeholder dan team - sebelum, selama dan setelah setiap Sprint. Dengan melibatkan client di setiap langkah proyek, ada kolaborasi yang tinggi antara client dan team proyek. Hal ini juga memberi lebih banyak kesempatan bagi team untuk benar-benar memahami visi client. Mempresentasikan software yang bekerja dengan lebih awal dan sering meningkatkan kepercayaan para stakeholder dalam kemampuan team untuk membuat software berkualitas tinggi dan mendorong mereka untuk lebih terlibat dalam proyek ini.

### **TRANSPARANSI**

Pendekatan Agile memberikan kesempatan unik bagi client untuk lebih terlibat dalam keseluruhan proyek. Ini beralih dari kemampuan memprioritaskan fitur untuk merencanakan dan meninjau pembaruan software yang berisi fitur baru.

### **PENGIRIMAN AWAL & MUDAH DIPREDIKSI**

Dengan menggunakan jadwal tetap Sprints 1-4 minggu, fitur baru bisa dibuat dengan cepat dan sering, juga dengan tingkat prediktabilitas yang tinggi. Ini juga memberi kesempatan untuk memasang atau menguji beta dari software lebih awal dari yang direncanakan jika ada nilai bisnis yang cukup.

### **BIAYA & JADWAL YANG BISA DITEBAK**

Karena setiap Sprint memiliki durasi yang tetap, biayanya dapat diprediksi dan sudah ditentukan di dalam pekerjaan yang akan dilakukan dalam jangka waktu yang ada membuat mudahnya perkiraan biaya yang ada. Dikombinasikan dengan perkiraan yang diberikan kepada client sebelum setiap Sprint, client dapat lebih mudah memahami biaya setiap fitur yang meningkatkan kemampuan pengambilan keputusan mengenai prioritas setiap fitur.

## MEMUNGKINKAN PERUBAHAN

Sementara team perlu tetap fokus untuk memberikan subset fitur produk yang disepakati di setiap Sprint, ada kesempatan untuk terus memperbaiki dan memprioritaskan isi backlog produk. Item backlog baru atau yang telah diubah dapat direncanakan dalam pertemuan Sprint berikutnya, memberikan kesempatan untuk mengenalkan perubahan dalam beberapa minggu.

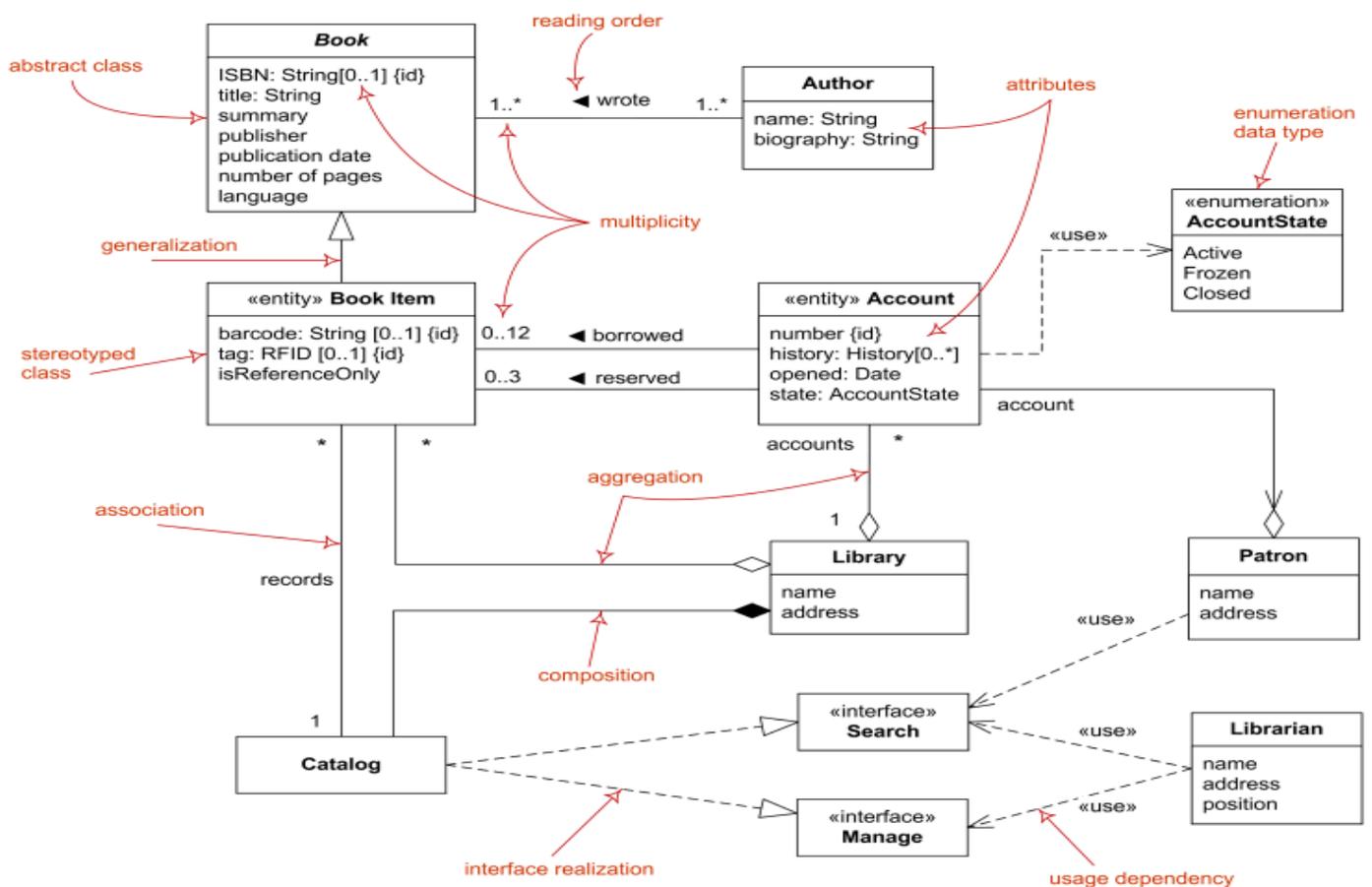
## FOKUS PENGGUNA

Agile biasanya menggunakan feedback user dengan kriteria yang berfokus pada bisnis untuk menentukan fitur produk. Dengan memfokuskan fitur pada kebutuhan pengguna nyata, setiap fitur memberikan nilai, tidak hanya sekedar komponen IT. Ini juga memberikan kesempatan untuk menguji beta software setelah setiap Sprint, mendapatkan umpan balik yang berharga di awal proyek dan memberikan kemampuan untuk melakukan perubahan seperlunya.

## MENINGKATKAN KUALITAS

Dengan membagi proyek menjadi unit-unit yang dapat dikelola, team proyek dapat berfokus pada pengembangan, pengujian dan kolaborasi berkualitas tinggi. Selain itu, dengan memproduksi sering membangun dan melakukan pengujian dan ulasan selama setiap Sprint, kualitas ditingkatkan dengan menemukan dan memperbaiki cacat dengan cepat dan mengidentifikasi kesalahan ekspektasi.

2.

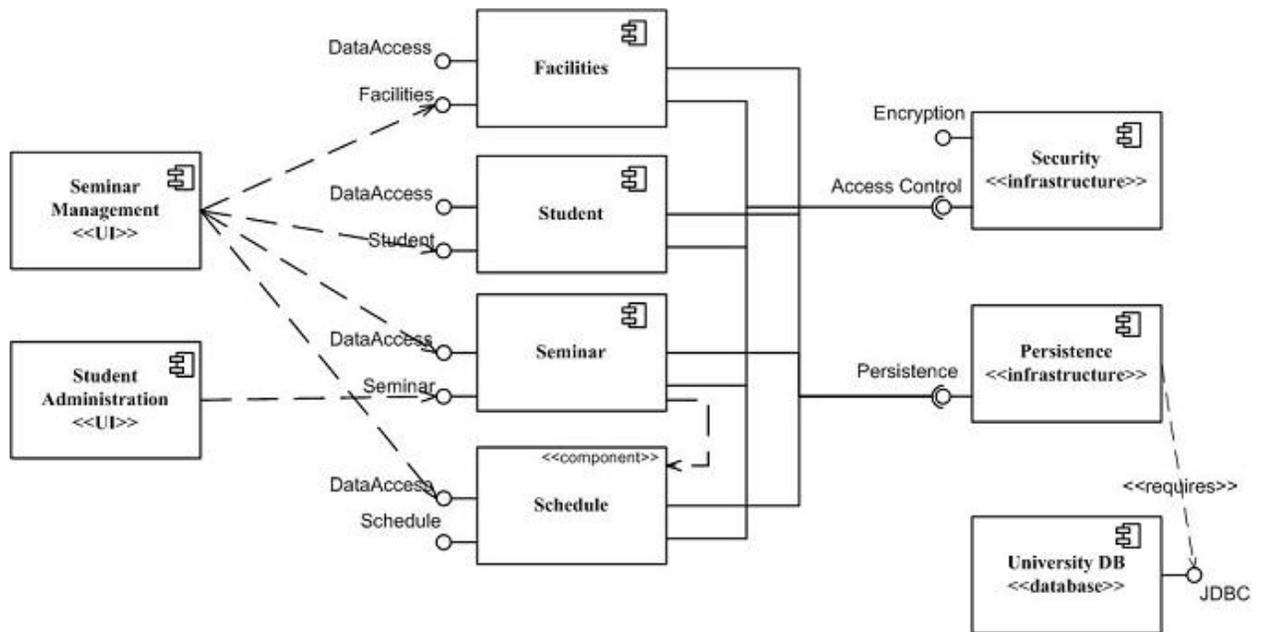


3. Dengan menggunakan pendekatan Agile development dapat mengurangi cost pada saat evaluasi dan validasi dikarenakan proses evaluasi dan validasi telah dilakukan pada setiap sprint sehingga tidak dibutuhkan biaya dan waktu yang lama untuk melakukan validasi dan evaluasi akhir.

1. Saya lebih memilih plan drive development karena Metode plan driven cocok digunakan untuk berbagai hal, antara lain :

- Jika customer lepas tangan terhadap pembuatan perangkat lunak, customer tersebut tidak mau ikut campur dalam pembuatannya, baik itu karena alasan sibuk, malas dan sebagainya.
- Timnya besar (terdiri dari banyak orang).
- Analisisnya besar.
- Perangkat lunak tersebut dirancang agar bisa digunakan dalam tempo yang lama, tidak hanya 1 atau 2 tahun saja.
- Teknologi yang tersedia kurang mendukung.
- Tim pengembangnya tersebar dimana-mana, tidak harus kumpul satu lokasi agar bisa membuat suatu perangkat lunak

2. desain component



3. Saran saya untuk mengurangi biaya pada tahap validasi dan evaluasi sebuah pembangunan software, sebelum pembangunan software harus ada analisis kebutuhan yang bertujuan

1. Mengelola hasil elistasi kebutuhan untuk menghasilkan dokumen spesifikasi kebutuhan yang isi keseluruhannya sesuai dengan apa yang diinginkan pengguna.
2. Mengembangkan persyaratan kualitas yang memadai dan rinci, dimana para manajer dapat membuat pekerjaan proyek yang realistis dan staf teknis dapat melanjutkan dengan perancangan, implementasi dan pengujian.

3. Membangun pemahaman tentang karakteristik ranah permasalahan dan sekumpulan kebutuhan untuk menemukan solusi.

Ketiga tujuan tersebut dapat dicapai oleh perekayasa kebutuhan dengan melalui serangkaian tahapan-tahapan aktivitas. Tahapan aktivitas tersebut ialah **Tahap Analisis Kebutuhan dimana** pengumpulan kebutuhan-kebutuhan dari semua elemen sistem perangkat lunak yang akan di bangun. Pada tahap ini dibentuk spesifikasi kebutuhan perangkat lunak, fungsi perangkat lunak yang dibutuhkan, performansi (unjuk kerja) sistem perangkat lunak, penjadwalan proyek, identifikasi sumber daya (manusia , perangkat keras dan perangkat lunak yang dibutuhkan) dan taksiran biaya pengembangan perangkat lunak.

Apabila semua kebutuhan telah terperinci, maka proses pembangunan software dan study kelayakan berjalan sesuai dengan kebutuhan, maka tidak ada hambatan pada saat validasi software dimana semua pembuktian dari proses, prosedur, kegiatan, pelengkapan yang di inginkan oleh pengguna sudah sesuai dengan kebutuhan pengguna dengan cara ini dapat menimalisir biaya validasi dan evaluasi terhadap suatu pembangunan software.

Nama : Yuni Astuti  
NIM : 192420004  
Mata Kuliah : Advanced Analysis and Design

Kerjakan soal berikut ini, kumpulkan jawaban anda dalam ms word!

1. Tentukan pilihan anda sendiri, apakah lebih cenderung ke *Plain-Drive Development* ataukah *Agile Development*? Jelaskan alasan anda!
2. Buatlah satu desain *class object* yang terlibat di dalam sebuah *component process* menggunakan UML notaion!
3. Apa saran anda untuk mengurangi biaya pada tahapan validasi dan evaluasi sebuah pembangunan *software* ?

**- Jawaban -**

1. *Plain Drive Development*

Dalam pengerjaan suatu proyek saya lebih memilih *Plain Drive Development* dikarenakan metodologi ini merupakan suatu metode yang lebih formal untuk membuat aplikasi. Dalam penggunaan Metodologi *Plain Drive Development* dalam suatu proyek dapat dilakukan pengulangan dan terprediksi, proses yang terorganisir, dokumen yang berkelanjutan, system muka arsitekturnya, mempunyai rencana yang terperinci, adanya proses pemantauan, pengendalian dan pengetahuan, manajemen resiko, verifikasi dan validasi.

Ada 3 hal yang paling menonjol dalam *Plain Drive Development* yaitu Proses Personal Software (PSP), Proses Perangkat Lunak dan Rational Unified Process (RUP).

Penggunaan metode *Plain Drive Development* cocok digunakan dalam berbagai hal, antara lain :

- a. Jika customer lepas tangan terhadap pembuatan perangkat lunak, customer tersebut tidak mau ikut campur dalam pembuatannya, baik itu karena sibuk atau sebagainya
- b. Anggota banyak (Timnya besar)
- c. Analisis yang terperinci

- d. Perangkat lunak tersebut dirancang agar bisa digunakan dalam waktu yang lama, tidak hanya 1 atau 2 tahun saja
- e. Teknologi yang kurang mendukung
- f. Tim pengembangannya bisa dilakukan dimana saja, tidak harus dalam satu lokasi

Untuk *Plain Drive Development* terdapat kelebihan dan kekurangannya yaitu :

Kelebihan :

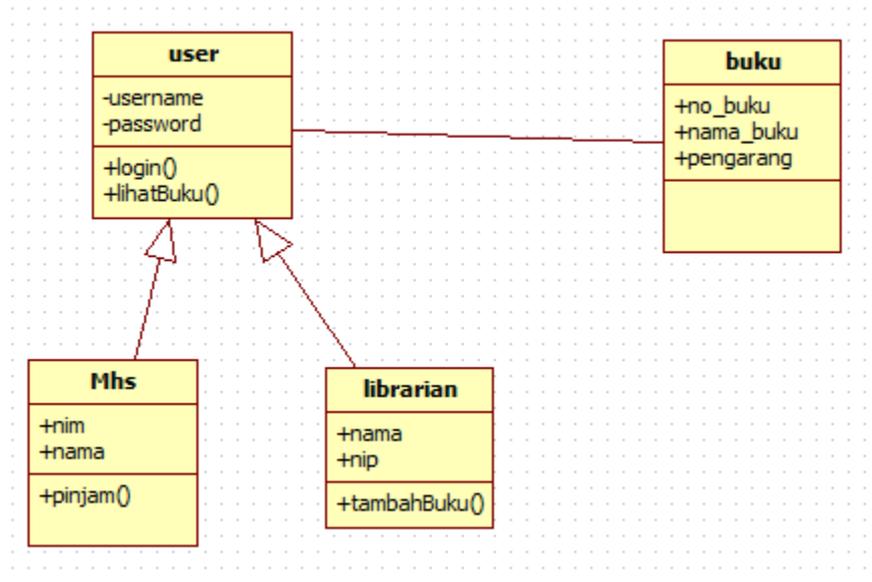
- a. Pembuatan program tersebut terencana, mulai dari spesifikasi, perancangan, validasi dan perubahan
- b. Customer tidak perlu repot-repot untuk memikirkan pembuatan perangkat lunak tersebut karena Cuma membuat kontrak kebutuhan diawal saja
- c. Perangkat lunak tersebut bisa digunakan dalam jangka waktu yang panjang sampai puluhan tahun
- d. Bisa dikerjakan oleh tim dimana mereka tidak harus berkumpul dalam satu lokasi
- e. Biaya relatif yang ringan karena pekerjaannya dilakukan berurutan
- f. Hasilnya sangat bagus karena pengolahannya sangat terperinci

Kekurangan :

- a. Tidak bisa dilakukan perubahan terhadap perangkat lunak ditengah-tengah pekerjaan sehingga tidak bisa menyesuaikan kemajuan dan keinginan masyarakat saat itu
- b. Hasilnya kurang sesuai dengan kebutuhan dan keinginan masyarakat pada saat perangkat lunak tersebut selesai

Menurut saya walaupun ada beberapa kekurangan untuk menggunakan metode *Plain Drive Development* tetapi masih bisa diatasi dengan customer nya.

2. Class object didalam component proses dalam menggunakan UML Notation :  
Contoh Class Object :



3. Saran saya untuk mengurangi biaya pada tahapan validasi dan evaluasi sebuah pembangunan *software* :

Dalam pembangunan *software* untuk mengurangi biaya dalam tahap validasi yaitu :

- a. Memastikan bahwa *software* yang akan digunakan sudah selesai sehingga tahap validasi nya tidak dilakukan berulang
- b. Membentuk komite validasi yang bertanggung jawab dalam pelaksanaan validasinya
- c. Membentuk *Validasi Master Plan* yaitu dokumen yang menguraikan (secara garis besar) pedoman pelaksanaan validasi dalam suatu *software* tersebut
- d. Membuat validasi prosedur tetap, protocol serta laporan validasi
- e. Pelaksanaan validasi yang jelas
- f. Melaksanakan peninjauan prodik pada *software* tersebut

Dalam pembangunan *software* untuk mengurangi biaya dalam tahap evaluasi yaitu :

1. Memastikan informasi yang valid tentang *software* yang akan digunakan
2. Memastikan strategi yang digunakan apakah telah benar
3. Ketepatan pemilihan cara operasi *software*
4. Pemilihan cara penggunaan yang lebih baik pada *software* tersebut
5. Pelaksanaan pengawasan terhadap kegiatan rutin yang sedang berjalan serta pengawasan untuk mengumpulkan informasi terhadap keluaran/hasil dan indicator yang digunakan untuk mengukur kinerja program
6. Pelaksanaan evaluasi dilaksanakan secara periodic dan berkala, dapat bersifat internal dan eksternal atau partisipatif sebagai umpan balik periodic.

# Tugas

## Rekayasa Perangkat Lunak

Nama :A.Firdaus

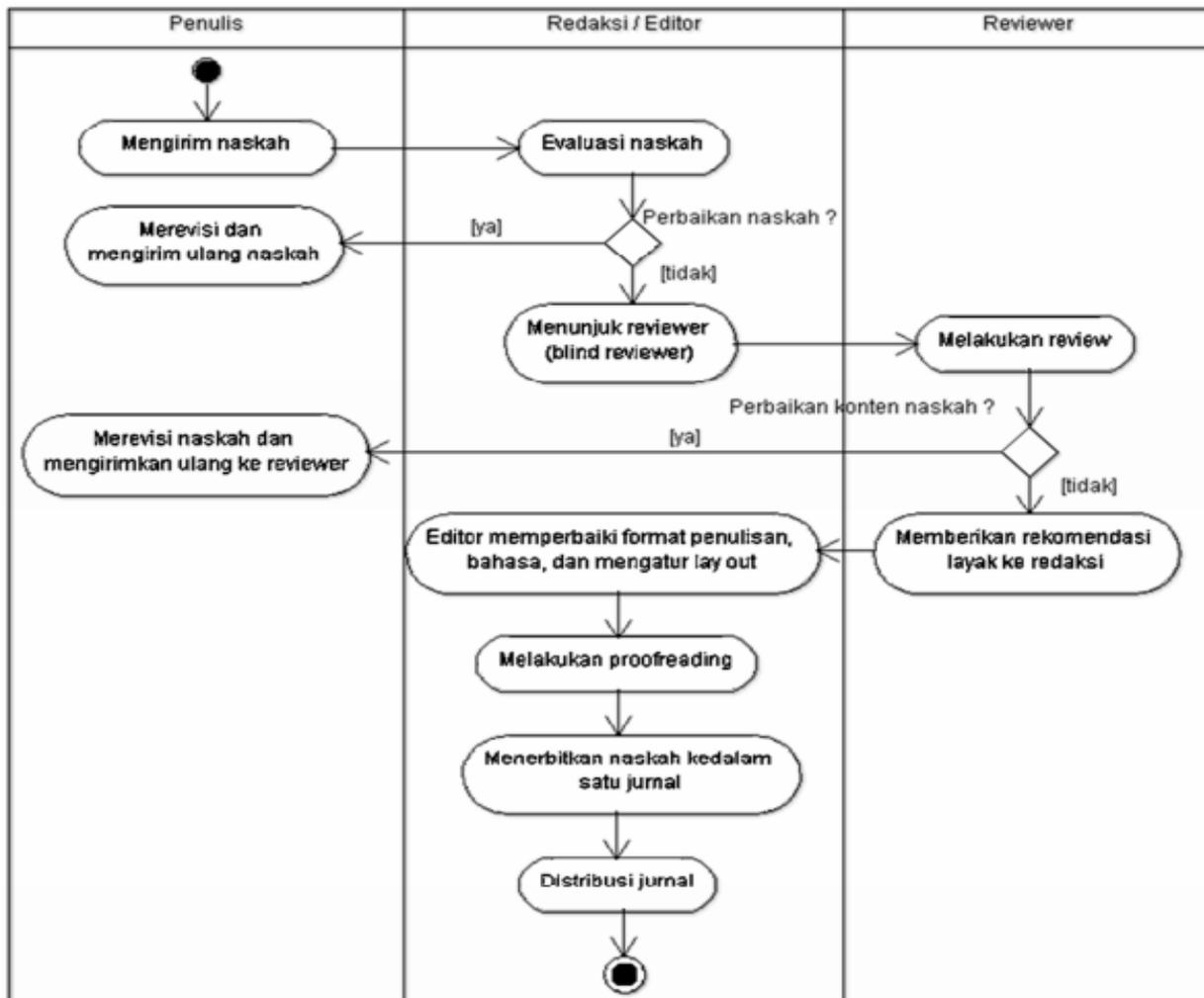
NIM : 192420043

Jawabansoal no. 1

Tidak ada alasan yang didorong oleh rencana dalam lingkungan bisnis ini. Rencana terus berubah. Jadi, satu-satunya hal yang cocok adalah pengembangan yang gesit. Agile adalah cara untuk memperbaiki produk secara berulang. Alasan utama Agile unggul dalam rencana yang didorong adalah:

- Model berulang Pekerjaan dilakukan dalam semburan singkat dua hingga tiga minggu pada akhir aplikasi yang berpotensi digunakan ditampilkan kepada pengguna akhir. Umpan balik yang diberikan oleh pengguna akhir digunakan untuk mengkalibrasi iterasi berikutnya dan seterusnya.
- Umpan balik di akhir iterasi, perubahan tambahan perlu diperagakan kepada pengguna akhir yang kemudian memberikan umpan balik yang berharga. Ini dapat diambil untuk meningkatkan iterasi di masa depan.
- Pekerjaan untuk iterasi didefinisikan dengan jelas dalam bentuk sprint backlog. Ini termasuk cerita dan bug yang perlu diselesaikan sebagai bagian dari sprint. Tidak ada pekerjaan baru yang bisa masuk ke dalam simpanan saat iterasi berjalan.

Jawaban soal no. 2 Diagram Proses Penerbitan Jurnal



Nama : ade saputra

Nim : 192420027

Kelas : MTI

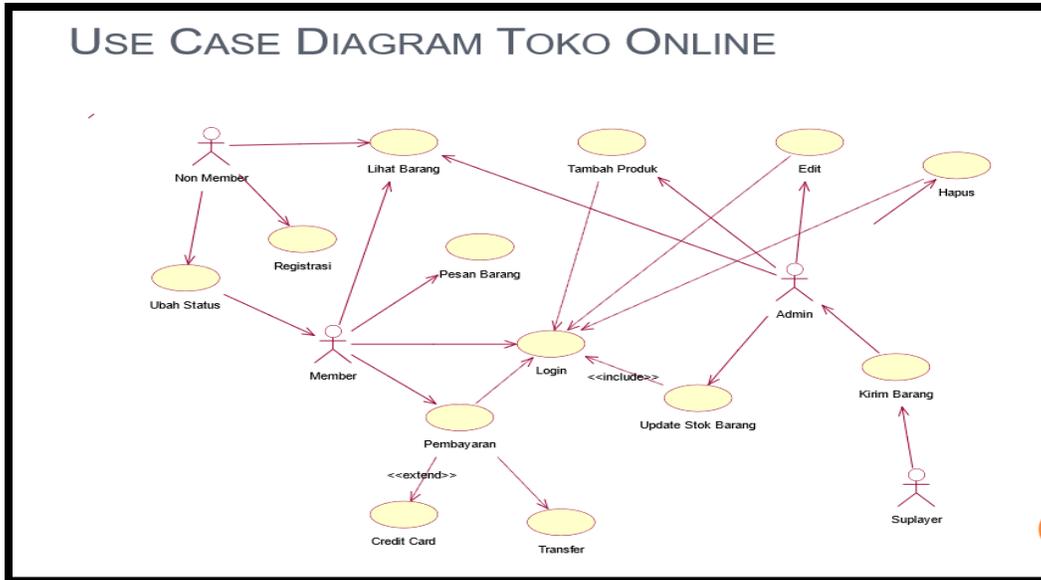
1. apakah lebih cenderung ke plain drive Development atau Agile Development ?

1. Saya lebih cenderung memakai Agile Development karena:

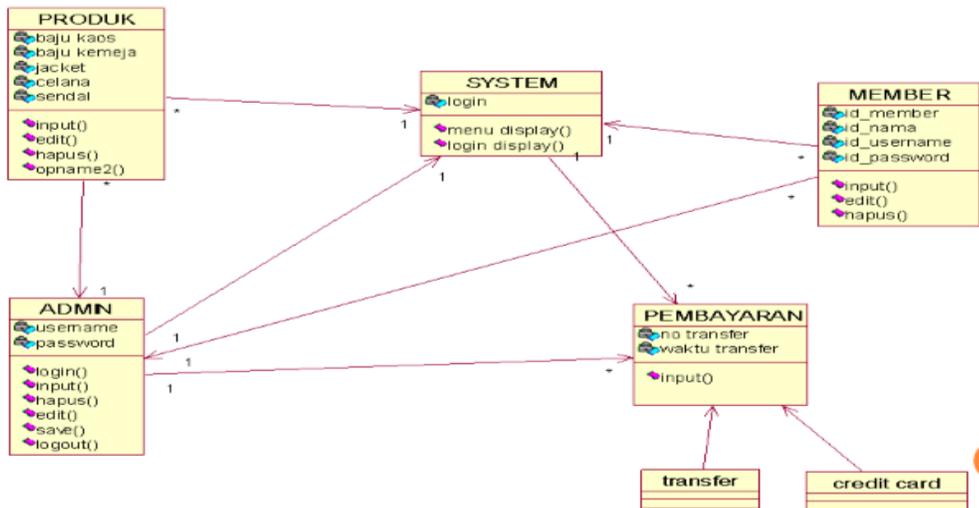
Menurut *Agile Alliance*, terdapat 12 prinsip yang harus dipenuhi bagi mereka yang ingin berhasil dalam penerapan *Agile Software Development*. Prinsip-prinsip tersebut antara lain :

- a) Kepuasan klien adalah prioritas utama dengan menghasilkan produk lebih awal dan terus menerus.
- b) Menerima perubahan kebutuhan, sekalipun diakhir pengembangan.
- c) Penyerahan hasil/software dalam hitungan waktu dua minggu sampai dua bulan.
- d) Bagian bisnis dan pembangun kerja sama tiap hari selama proyek berlangsung.
- e) Membangun proyek dilingkungan orang-orang yang bermotivasi tinggi yang bekerja dalam lingkungan yang mendukung dan yang dipercaya untuk dapat menyelesaikan proyek.
- f) Komunikasi dengan berhadapan langsung adalah komunikasi yang efektif dan efisien.
- g) Software yang berfungsi adalah ukuran utama dari kemajuan proyek.
- h) Dukungan yang stabil dari sponsor, pembangun, dan pengguna diperlukan untuk menjaga perkembangan yang berkesinambungan.
- i) Perhatian kepada kehebatan teknis dan desain yang bagus meningkatkan sifat agile.
- j) Kesederhanaan sangat penting.
- k) Arsitek. Kebutuhan dan desain yang bagus muncul dari tim yang mengatur dirinya sendiri.
- l) Secara periodik tim evaluasi diri dan mencari cara untuk lebih efektif dan segera melakukannya.

2. Buatlah desain class Objek yang terlibat dalam sebuah komponentproses menggunakan UML Notation?



### CLASS DIAGRAM TOKO ONLINE



3. Apa saran anda untuk mengurangi Biaya pada tahapan validasi dan evaluasi sebuah pembangunan Software?
- A. Pengurangan Bug pada system Software sehingga Bisa mengurangi Banyak nya tahap Testing
  - B. Dalam pengembangan system Software Harus Lebih Teliti Sehingga dapat menghindari banyak nya perubahan system Software
  - C. Sebelum ke tahap pengembangan harus lebih Teliti dalam membuat Kontark pembangunan Software ini berisikan Point –point dalam pengurangan biaya
  - D. Mengetahui kebutuhan hal terpenting dalam kebutuhan User

Nama : Andrian Perdana  
Kelas : AR1  
NIM : 192420017

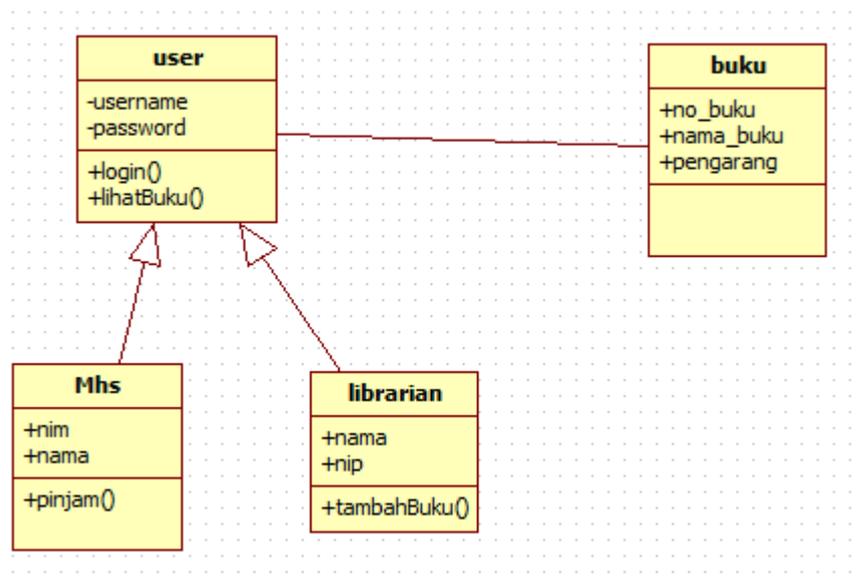
1. Tentukan pilihan anda sendiri, apakah lebih cenderung ke *Plain-Drive Development* ataukah *Agile Development*? Jelaskan alasan anda!
2. Buatlah satu desain *class object* yang terlibat di dalam sebuah *component process* menggunakan UML notaion!
3. Apa saran anda untuk mengurangi biaya pada tahapan validasi dan evaluasi sebuah pembangunan *software* ?

Jawaban :

1. Saya cenderung ke Plain-Drive Development

alasanya adalah :

- Perencanaan kebutuhan secara detail terhadap aplikasi tertuang sejak awal sehingga pengerjaan proyek akan mudah dikontrol dan terjadwal dengan baik;
- Biaya yang ditimbulkan juga bisa diprediksi dan sesuai dengan kontrak kerjasama karena rincian proses tidak sering berubah-ubah.



3. Menurut saya untuk mengurangi biaya validasi sebuah software adalah :

- Identifikasi kebutuhan terhadap software dari pengguna harus detail yang akan menyebabkan pada saat validasi benar benas seuai dengan kebutuhan yang diinginkan;

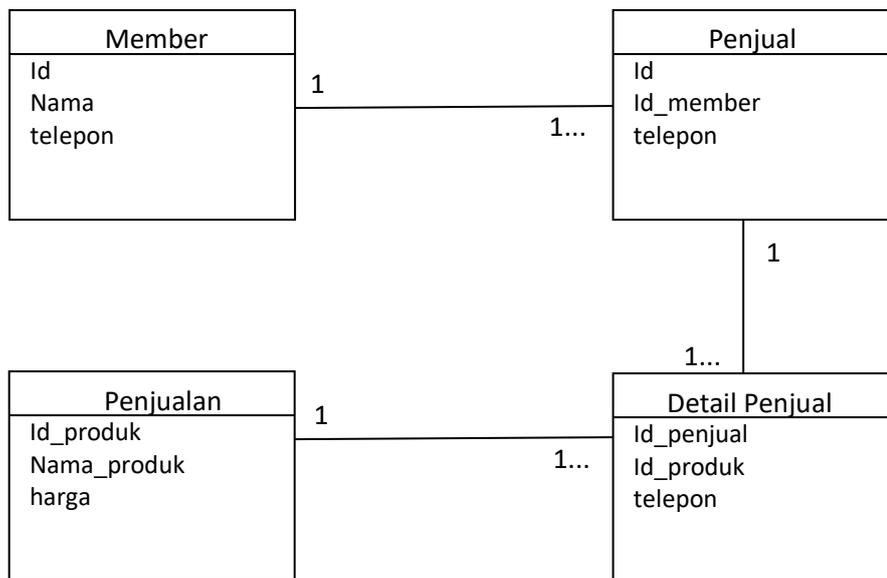
- Pemilihan software yang tepat dalam pengembangan aplikasi akan meningkatkan efektifitas dalam pengujian;
- semakin sedikit code yang dituliskan akan semakin sedikit yang anda pikirkan dalam memaintain aplikasi pada saat proses validasi.

Nama Mahasiswa : Andriansyah

NIM : 20192420020

1. Menurut saya **agile development** adalah pilihan yang tepat dikarenakan perkembangan teknologi yang begitu cepat yang dibutuhkan oleh masyarakat sekarang ini untuk memberikan kemudahan diberbagai aspek dalam waktu yang singkat. Metode Agile development cocok diterapkan untuk mengikuti keinginan masyarakat yang lebih suka simple, mudah, aman.

2.



3. Saran saya melakukan evaluasi proses berbasis kuisisioner sehingga dapat diketahui dan diprediksikan tingkat kelemahan sistem atau perangkat lunak yang sedang dikembangkan sehingga tim pengembang dapat mempredikasikan kedepan nya sistem tersebut sehingga tidak banyak mengeluarkan biaya.

Mata Kuliah : *Advanced Database*  
Dosen : Tri Basuki Kurniawan, S.Kom, M.Eng, Ph.D  
Kelas : Reguler **AR1**

Nama : Ardiansyah (192420013)

1. Tentukan pilihan anda sendiri, apakah lebih cenderung ke *Plain-Drive Development* ataukah *Agile Development*? Jelaskan alasan anda!
2. Buatlah satu desain *class object* yang terlibat di dalam sebuah *component process* menggunakan UML notaion!
3. Apa saran anda untuk mengurangi biaya pada tahapan validasi dan evaluasi sebuah pembangunan *software* ?
4. Apa yang mendorong anda untuk menggunakan metode *Waterfall* pada project yang akan ada bangun? Jelaskan alasan anda!
5. Mana yang akan anda prioritaskan pada project perusahaan anda, apakah *Generic Product* atau *Customize Product*? Jelaskan beserta alasannya!

### Jawab

1. Bersumber dari Wikipedia,  
**Agile Development Methods** adalah sekelompok metodologi pengembangan perangkat lunak yang didasarkan pada prinsip-prinsip yang sama atau pengembangan sistem jangka pendek yang memerlukan adaptasi cepat dari pengembang terhadap perubahan dalam bentuk apapun. *Agile development methods* merupakan salah satu dari [Metodologi pengembangan perangkat lunak](#) yang digunakan dalam pengembangan perangkat lunak. Agile memiliki pengertian bersifat cepat, ringan, bebas bergerak, dan waspada.<sup>[1]</sup> Sehingga saat membuat perangkat lunak dengan menggunakan *agile development methods* diperlukan inovasi dan responsibiliti yang baik antara tim pengembang dan klien agar kualitas dari perangkat lunak yang dihasilkan bagus dan kelincahan dari tim seimbang.

#### Kelebihan

Beberapa kelebihan dari *agile* diantaranya<sup>[8]</sup>:

- 82% Menambah produktivitas tim.
- 77% Menambah kualitas perangkat lunak.
- 78% Menambah kepuasan klien.
- 37% Menghemat biaya.

#### Kekurangan

Sedangkan kekurangan dari *agile* antara lain:

- Agile tidak akan berjalan dengan baik jika komitmen tim kurang.
- Tidak cocok dalam skala tim yang besar (>20 orang).
- Perkiraan waktu release dan harga perangkat lunak sulit ditentukan.

Bersumber dari wikipersity

*Plan-driven software development is a more formal specific approach to creating an application. Plan-driven methodologies all incorporate: repeatability and predictability, a defined incremental process, extensive documentation, up-front system architecture, detailed plans, process monitoring, controlling and education, risk management, verification and validation.* <sup>[1]</sup>

Berdasarkan data penjelasan diatas, maka saya yg masih pemula ini lebih cenderung dalam analisa menggunakan **plain – drive development**

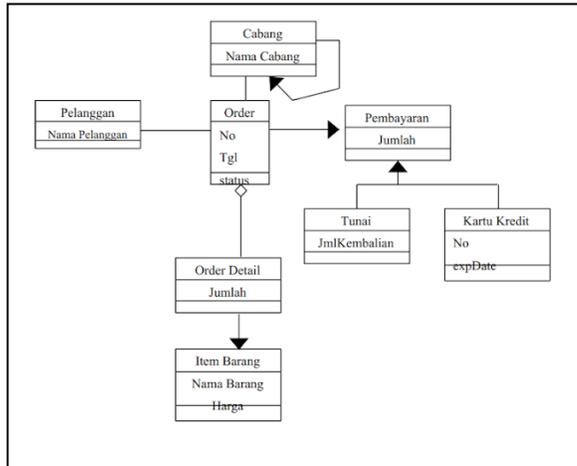
Alasan yang bisa saya jabarkan, kenapa tidak condong menggunakan **agile-development**:

## Kekurangan

- Agile tidak akan berjalan dengan baik jika komitmen tim kurang.
- Tidak cocok dalam skala tim yang besar (>20 orang).
- Perkiraan waktu release dan harga perangkat lunak sulit ditentukan.

Sedangkan **plain-drive development** dari struktur desain sampai ke perancangan membutuhkan standar yg baik dengan kelengkapan dokumentasi.

2.



3. Menurut saya untuk mengurangi biaya pada saat *validasi* dan *evaluasi* :  
Membuat rencana draft yg pasti sehingga pada waktu / saat pelaksanaan tidak menjadi sia sia dan berguna secara baik.
4. Bersumber dari Wikipedia:  
Saya menganalisa bahwa konsep waterfall dalam sebuah project sangat baik digunakan sebab menggunakan tahap demi tahap. Misal dalam sebuah pembuatan software, tahapan rancangan – tahapan awal pengerjaan – tahapan proses pengerjaan – tahapan akhir pengerjaan – finishing. Semua proses tsb mesti dilalui secara bertahap tidak melompat. Jadi lebih akurat dalam pengerjaannya dan menghasilkan sebuah software yang baik dan bermutu.
5. Generic (market-pull) products Tim memulai pengembangan produk dengan peluang pasar dan kemudian memilih semua teknologi yang tepat untuk memenuhi kebutuhan pasar. Contohnya sporting goods, furniture, tools.

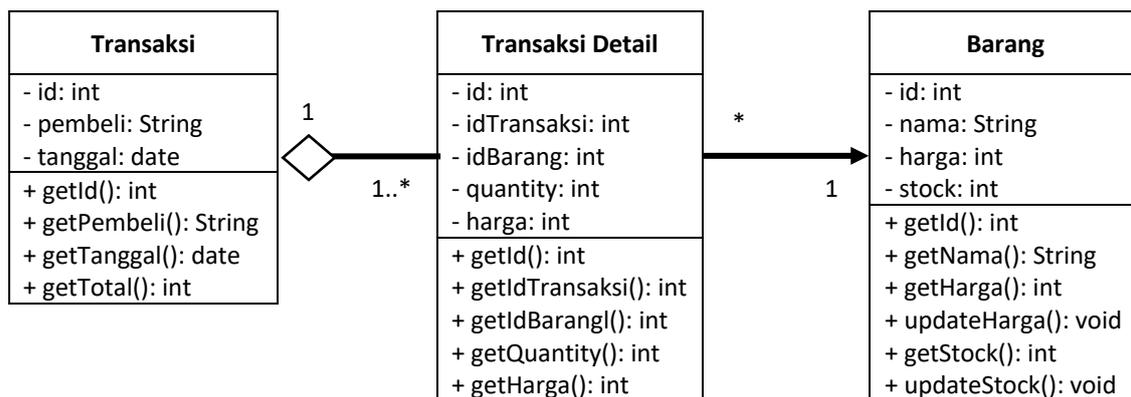
Customized products Produk baru yang akan dihasilkan hanya sedikit perbedaannya dari konfigurasi yang sudah ada. Contohnya motor, saklar, baterai, container.

Dari penjelasan data diatas, saya lebih cenderung ke generic product, sebab dgn metode ini sesuatu yang baru diciptakan dan lisensi ditangan kita, tapi kalau customize product, hanya melakukan pengembangan dari sesuatu yang sudah ada,

1) Saya lebih cenderung ke *Agile Development* dengan alasan sebagai berikut:

- Proses pengembangan lebih mengutamakan fungsional produk dibandingkan dengan dokumentasi atau desain produk.
- Proses pengembangan lebih cepat.
- Pelaporan progress lebih sering sehingga bisa mendapatkan feedback lebih cepat dari client.
- Menerima perubahan pada masa pengembangan.

2) Component Process Transaksi Penjualan



3) Untuk mengurangi biaya dapat dilakukan *Incremental Validation and Evaluation*. Hal ini dapat dilakukan dengan cara melakukan validasi dan evaluasi setiap kali suatu *component process* selesai dibuat. Metode ini dapat mengurangi waktu dan biaya dibandingkan dengan melakukan validasi dan evaluasi ketika tahap pengembangan telah selesai. Alasan metode ini dapat mengurangi waktu dan biaya karena jika validasi dan evaluasi dilakukan setelah tahap pengembangan maka validasi dan evaluasi harus dilakukan secara keseluruhan dan ketika terjadi kesalahan pada suatu tahap maka harus mengulang dari awal lagi.

1. Tentukan pilihan anda sendiri, apakah lebih cenderung ke plan-drive development ataukah agile development ? jelaskan alasan anda !
2. Buatlah satu desain class object yang terlihat didalam sebuah component process menggunakan uml notaion !
3. Apa saran anda untuk mengurangkan biaya pada tahapan validasi dan evaluasi sebuah pembangunan software?

Jawab

1. Menurut saya cenderung pada agile development, dikarenakan keunggulan agile development adalah :
  - Meningkatkan kepuasan kepada klien
  - Pembangunan system dibuat lebih cepat
  - Mengurangi resiko kegagalan implementasi software dari segi non-teknis
  - Jika pada saat pembangunan system terjadi kegagalan, kerugian dari segi materi relative kecil.
  - Bisa melakukan review pelanggan mengenai software yang di buat lebih awal.

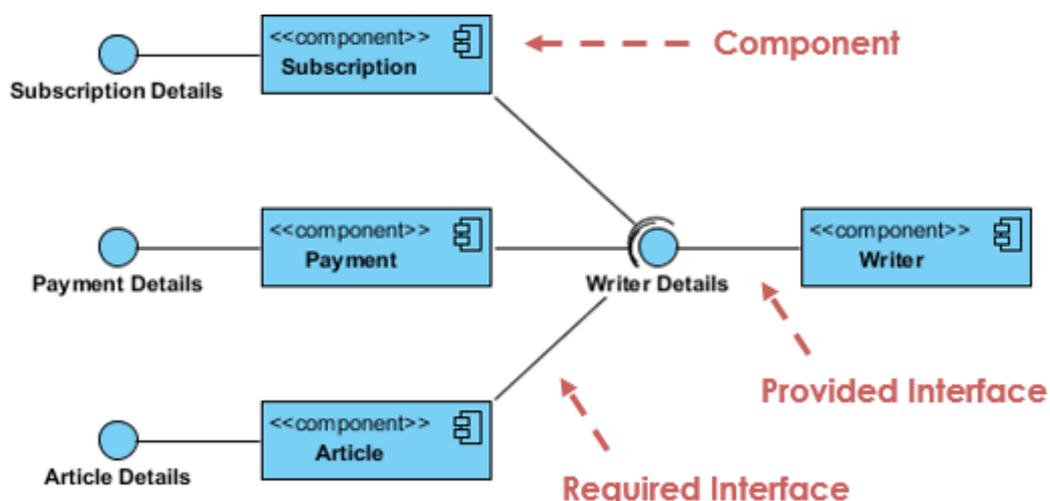
Dan kekurangannya :

- Total lama pengembangan menjadi lebih lama
- Meningkatkan resiko kesalahan teknis
- Proses pengembangan menjadi agak kurang terorganisir

Dibanding kekurangannya dengan plan-drive development, saya rasa agile development lebih baik untuk kepuasan pada klien, berikut kekurangan dari plan-drive development :

- Tidak bisa dilakukan perubahan terhadap perangkat lunak di tengah-tengah pekerjaan. Sehingga tidak bisa menyesuaikan kemajuan dan keinginan masyarakat saat itu.
- Hasilnya kurang sesuai dengan kebutuhan dan keinginan masyarakat pada saat perangkat lunak tersebut selesai.

2.



3. saran untuk mengurangi biaya pada tahapan validasi dan evaluasi sebuah pembangunan software
  - Dalam membudayakan dokumentasi dalam proses pengembangan perangkat lunak VTD, perlu adanya komitmen antara project manager dengan anggota tim.
  - Perlu adanya peningkatan kualitas proses pada area konfigurasi manajemen, integrasi produk, kebutuhan pengembangan, dan manajemen resiko.
  - Resiko dalam mengembangkan perangkat lunak pada VTD dapat diminimalisir bahkan dihilangkan apabila ada monitoring dan pengendalian selama proses pengembangan perangkat lunak.

Mata Kuliah : *Advanced Database*  
Dosen : Tri Basuki Kurniawan, S.Kom, M.Eng, Ph.D  
Kelas : Reguler **AR1**

Nama : EKO PURWANTO (192420019)

1. Tentukan pilihan anda sendiri, apakah lebih cenderung ke *Plain-Drive Development* ataukah *Agile Development*? Jelaskan alasan anda!
2. Buatlah satu desain *class object* yang terlibat di dalam sebuah *component process* menggunakan UML notaion!
3. Apa saran anda untuk mengurangi biaya pada tahapan validasi dan evaluasi sebuah pembangunan *software* ?
4. Apa yang mendorong anda untuk menggunakan metode *Waterfall* pada project yang akan ada bangun? Jelaskan alasan anda!
5. Mana yang akan anda prioritaskan pada project perusahaan anda, apakah *Generic Product* atau *Customize Product*? Jelaskan beserta alasannya!

### Jawab

1. Bersumber dari Wikipedia,  
**Agile Development Methods** adalah sekelompok metodologi pengembangan perangkat lunak yang didasarkan pada prinsip-prinsip yang sama atau pengembangan sistem jangka pendek yang memerlukan adaptasi cepat dari pengembang terhadap perubahan dalam bentuk apapun. *Agile development methods* merupakan salah satu dari [Metodologi pengembangan perangkat lunak](#) yang digunakan dalam pengembangan perangkat lunak. Agile memiliki pengertian bersifat cepat, ringan, bebas bergerak, dan waspada.<sup>[1]</sup> Sehingga saat membuat perangkat lunak dengan menggunakan *agile development methods* diperlukan inovasi dan responsibiliti yang baik antara tim pengembang dan klien agar kualitas dari perangkat lunak yang dihasilkan bagus dan kelincahan dari tim seimbang.

#### Kelebihan

Beberapa kelebihan dari *agile* diantaranya<sup>[8]</sup>:

- 82% Menambah produktivitas tim.
- 77% Menambah kualitas perangkat lunak.
- 78% Menambah kepuasan klien.
- 37% Menghemat biaya.

#### Kekurangan

Sedangkan kekurangan dari *agile* antara lain:

- Agile tidak akan berjalan dengan baik jika komitmen tim kurang.
- Tidak cocok dalam skala tim yang besar (>20 orang).
- Perkiraan waktu release dan harga perangkat lunak sulit ditentukan.

Bersumber dari wikipersity

*Plan-driven software development is a more formal specific approach to creating an application. Plan-driven methodologies all incorporate: repeatability and predictability, a defined incremental process, extensive documentation, up-front system architecture, detailed plans, process monitoring, controlling and education, risk management, verification and validation.* <sup>[1]</sup>

Berdasarkan data penjelasan diatas, maka saya yg masih pemula ini lebih cenderung dalam analisa menggunakan **plain – drive development**

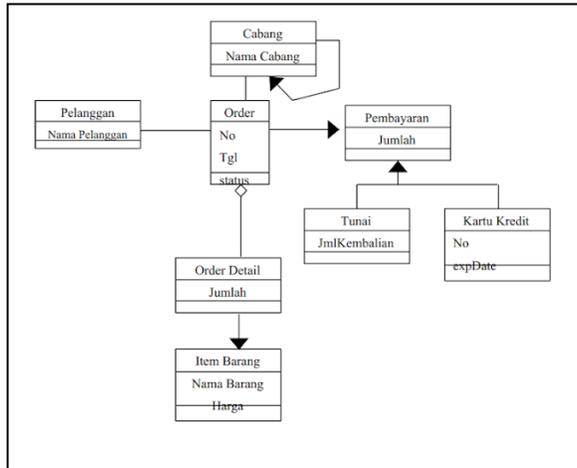
Alasan yang bisa saya jabarkan, kenapa tidak condong menggunakan **agile-development**:

## Kekurangan

- Agile tidak akan berjalan dengan baik jika komitmen tim kurang.
- Tidak cocok dalam skala tim yang besar (>20 orang).
- Perkiraan waktu release dan harga perangkat lunak sulit ditentukan.

Sedangkan **plain-drive development** dari struktur desain sampai ke perancangan membutuhkan standar yg baik dengan kelengkapan dokumentasi.

2.



3. Menurut saya untuk mengurangi biaya pada saat *validasi* dan *evaluasi* :  
Membuat rencana draft yg pasti sehingga pada waktu / saat pelaksanaan tidak menjadi sia sia dan berguna secara baik.
4. Bersumber dari Wikipedia:  
Saya menganalisa bahwa konsep waterfall dalam sebuah project sangat baik digunakan sebab menggunakan tahap demi tahap. Misal dalam sebuah pembuatan software, tahapan rancangan – tahapan awal pengerjaan – tahapan proses pengerjaan – tahapan akhir pengerjaan – finishing. Semua proses tsb mesti dilalui secara bertahap tidak melompat. Jadi lebih akurat dalam pengerjaannya dan menghasilkan sebuah software yang baik dan bermutu.
5. Generic (market-pull) products Tim memulai pengembangan produk dengan peluang pasar dan kemudian memilih semua teknologi yang tepat untuk memenuhi kebutuhan pasar. Contohnya sporting goods, furniture, tools.

Customized products Produk baru yang akan dihasilkan hanya sedikit perbedaannya dari konfigurasi yang sudah ada. Contohnya motor, saklar, baterai, container.

Dari penjelasan data diatas, saya lebih cenderung ke generic product, sebab dgn metode ini sesuatu yang baru diciptakan dan lisensi ditangan kita, tapi kalau customize product, hanya melakukan pengembangan dari sesuatu yang sudah ada,

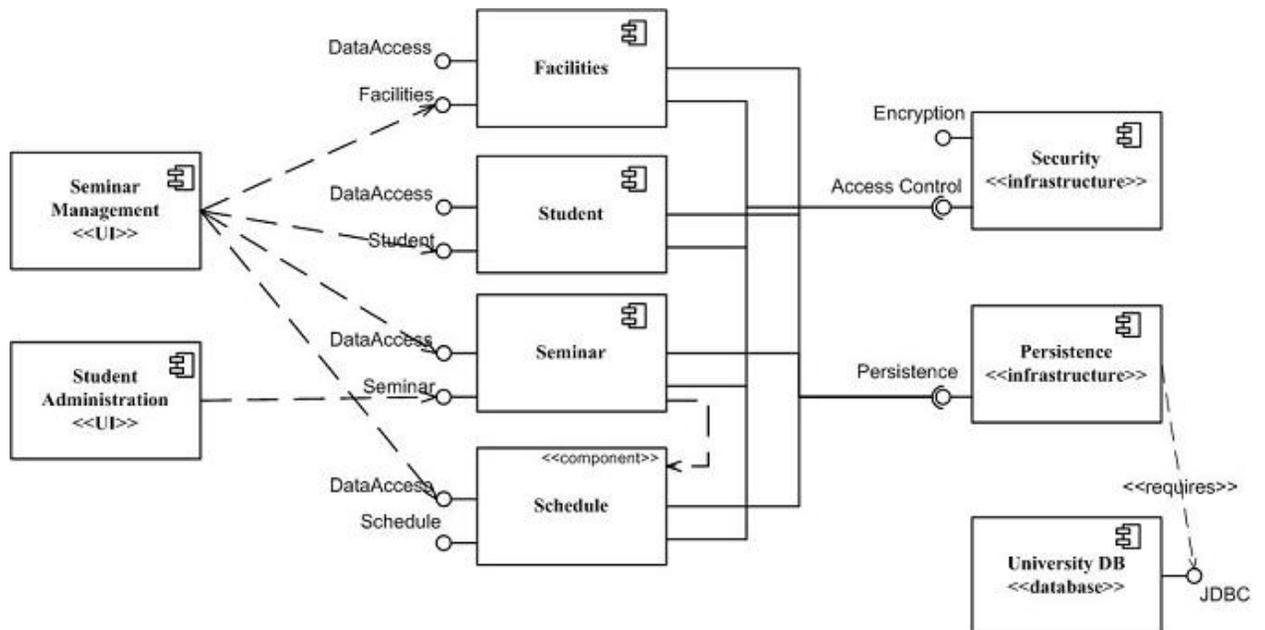
NAMA : Hasirul Qodar

KELAS : Reguler AR1

1. Saya lebih memilih plan drive development karena Metode plan driven cocok digunakan untuk berbagai hal, antara lain :

- Jika customer lepas tangan terhadap pembuatan perangkat lunak, customer tersebut tidak mau ikut campur dalam pembuatannya, baik itu karena alasan sibuk, malas dan sebagainya.
- Timnya besar (terdiri dari banyak orang).
- Analisisnya besar.
- Perangkat lunak tersebut dirancang agar bisa digunakan dalam tempo yang lama, tidak hanya 1 atau 2 tahun saja.
- Teknologi yang tersedia kurang mendukung.
- Tim pengembangnya tersebar dimana-mana, tidak harus kumpul satu lokasi agar bisa membuat suatu perangkat lunak

2. desain component



3. Saran saya untuk mengurangi biaya pada tahap validasi dan evaluasi sebuah pembangunan software, sebelum pembangunan software harus ada analisis kebutuhan yang bertujuan

1. Mengelola hasil elistasi kebutuhan untuk menghasilkan dokumen spesifikasi kebutuhan yang isi keseluruhannya sesuai dengan apa yang diinginkan pengguna.
2. Mengembangkan persyaratan kualitas yang memadai dan rinci, dimana para manajer dapat membuat pekerjaan proyek yang realistis dan staf teknis dapat melanjutkan dengan perancangan, implementasi dan pengujian.
3. Membangun pemahaman tentang karakteristik ranah permasalahan dan sekumpulan kebutuhan untuk menemukan solusi.

Ketiga tujuan tersebut dapat dicapai oleh perekayasa kebutuhan dengan melalui serangkaian tahapan-tahapan aktivitas. Tahapan aktivitas tersebut ialah **Tahap Analisis Kebutuhan dimana** pengumpulan kebutuhan-kebutuhan dari semua elemen sistem perangkat lunak yang akan di bangun. Pada tahap ini dibentuk spesifikasi kebutuhan perangkat lunak, fungsi perangkat lunak yang dibutuhkan, performansi (unjuk kerja) sistem perangkat lunak, penjadwalan proyek, identifikasi sumber daya (manusia , perangkat keras dan perangkat lunak yang dibutuhkan) dan taksiran biaya pengembangan perangkat lunak.

Apabila semua kebutuhan telah terperinci, maka proses pembangunan software dan study kelayakan berjalan sesuai dengan kebutuhan, maka tidak ada hambatan pada saat validasi software dimana semua pembuktian dari proses,prosedur,kegiatan, pelengkapan yang di inginkan oleh pengguna sudah sesuai dengan kebutuhan pengguna dengan cara ini dapat menimalisir biaya validasi dan evaluasi terhadap suatu pembangunan software.

**Tugas Elearning Advanced Analysis and Design**  
**Hendra Yada Putra**  
**MTI Regular B angkatan 21**

---

Berikut Jawaban atas Tugas Quiz Elearning tanggal 10 Desember 2019.

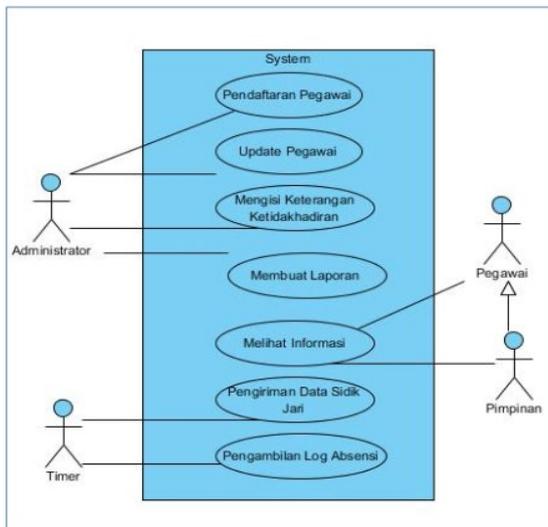
1. Terkait metode pengembangan yang mana akan saya pilih apakah lebih cenderung ke Plain Drive Development atau Agile Development, maka saya akan melihat dulu skala besera/kecil project yang akan dikerjakan, skala project dimaksud adalah seberapa besar risiko dalam project tersebut, baik risiko terhadap output yang dihasilkan oleh project itu sendiri (seperti jika terjadi kesalahan dari output sistem yang dibangun, sehingga dapat menimbulkan risiko finansial atau risiko reputasi yang merugikan customer, maka ini harus dikategorikan besar) maupun risiko dalam proses developingnya sendiri (seperti jika terjadi keterlambatan dalam project maka akan berdampak pada sisi compliance yang dapat menimbulkan denda maupun risiko reputasi).

Berdasarkan hal tersebut jika project masuk dalam skala besar, maka saya akan mengutamakan menggunakan Plain-Drive project, dimana segala sesuatunya harus didesain sematang mungkin untuk developmentnya, sebaliknya jika project masuk dalam skala kecil, maka saya akan cenderung menggunakan agile development.

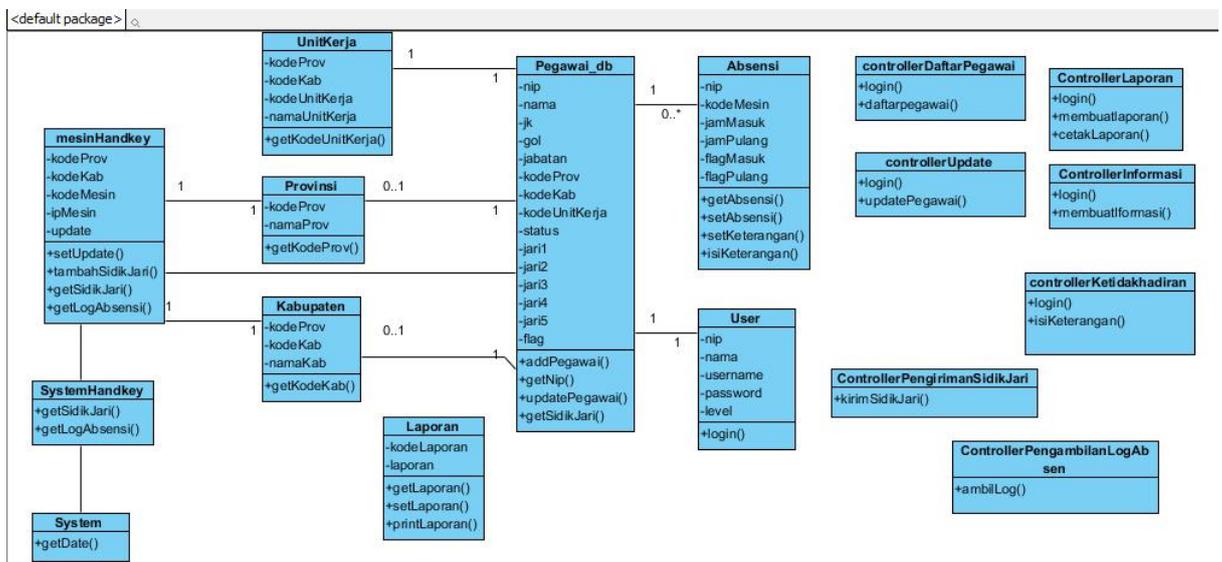
2. Saran saya dalam upaya menekan cost dalam proses validasi dan evaluasi adalah dengan menggabungkan beberapa tahapan testing pada proses validasi, khususnya dengan menggabungkan tahapan Development/componen testing dan system testing. dimana testing ini dimungkinkan dapat dilakukan bersamaan, selain itu dapat juga menggunakan tools bantu untuk testing komponen seperti selenium dan testcomplete terkait performance, system test, dan fungsi test.

1. Memilih Agile development, karena
  - a. Dapat melakukan review pelanggan mengenai software yang dibuat lebih awal.
  - b. Pembangunan system dibuat lebih cepat.
  - c. Perubahan dengan cepat ditangani
  - d. Jika pada saat pembangunan system terjadi kegagalan kerugian dari segi materi relatif kecil.

2. Desain clas object



Gambar use case diagram system Absensi



3. Saran untuk Mengurangi biaya Pembangunan Software

Yang pertama adalah pertimbangan apakah software tersebut sudah jadi atau masih dalam tahap pengembangan. Jika software tersebut sudah jadi maka vendor tidak lagi

melakukan pengembangan dan sudah tidak ada lagi biaya untuk pengembangan software tersebut. Lalu bandingkan dengan software yang masih dalam pengembangan, yang berarti software tersebut masih baru, sehingga pasti muncul biaya pengembangan. Dan biaya tersebut digunakan oleh vendor untuk menggaji programmer, operasional, transport, internet connection, diskusi jasa, dan yang lainnya. Jadi sudah tentu software yang sudah jadi lebih murah dibandingkan dengan software yang masih dalam pengembangan.

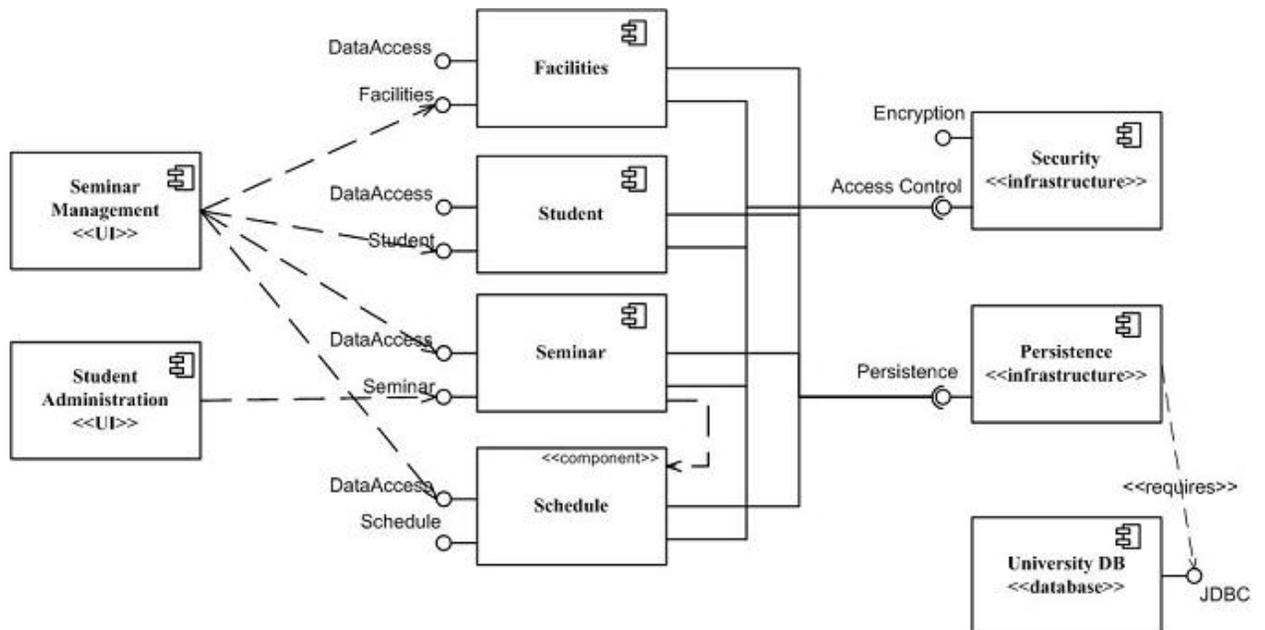
NAMA : Hasirul Qodar

KELAS : Reguler AR1

1. Saya lebih memilih plan drive development karena Metode plan driven cocok digunakan untuk berbagai hal, antara lain :

- Jika customer lepas tangan terhadap pembuatan perangkat lunak, customer tersebut tidak mau ikut campur dalam pembuatannya, baik itu karena alasan sibuk, malas dan sebagainya.
- Timnya besar (terdiri dari banyak orang).
- Analisisnya besar.
- Perangkat lunak tersebut dirancang agar bisa digunakan dalam tempo yang lama, tidak hanya 1 atau 2 tahun saja.
- Teknologi yang tersedia kurang mendukung.
- Tim pengembangnya tersebar dimana-mana, tidak harus kumpul satu lokasi agar bisa membuat suatu perangkat lunak

2. desain component



3. Saran saya untuk mengurangi biaya pada tahap validasi dan evaluasi sebuah pembangunan software, sebelum pembangunan software harus ada analisis kebutuhan yang bertujuan

1. Mengelola hasil elistasi kebutuhan untuk menghasilkan dokumen spesifikasi kebutuhan yang isi keseluruhannya sesuai dengan apa yang diinginkan pengguna.
2. Mengembangkan persyaratan kualitas yang memadai dan rinci, dimana para manajer dapat membuat pekerjaan proyek yang realistis dan staf teknis dapat melanjutkan dengan perancangan, implementasi dan pengujian.
3. Membangun pemahaman tentang karakteristik ranah permasalahan dan sekumpulan kebutuhan untuk menemukan solusi.

Ketiga tujuan tersebut dapat dicapai oleh perekayasa kebutuhan dengan melalui serangkaian tahapan-tahapan aktivitas. Tahapan aktivitas tersebut ialah **Tahap Analisis Kebutuhan dimana** pengumpulan kebutuhan-kebutuhan dari semua elemen sistem perangkat lunak yang akan di bangun. Pada tahap ini dibentuk spesifikasi kebutuhan perangkat lunak, fungsi perangkat lunak yang dibutuhkan, performansi (unjuk kerja) sistem perangkat lunak, penjadwalan proyek, identifikasi sumber daya (manusia , perangkat keras dan perangkat lunak yang dibutuhkan) dan taksiran biaya pengembangan perangkat lunak.

Apabila semua kebutuhan telah terperinci, maka proses pembangunan software dan study kelayakan berjalan sesuai dengan kebutuhan, maka tidak ada hambatan pada saat validasi software dimana semua pembuktian dari proses,prosedur,kegiatan, pelengkapan yang di inginkan oleh pengguna sudah sesuai dengan kebutuhan pengguna dengan cara ini dapat menimalisir biaya validasi dan evaluasi terhadap suatu pembangunan software.

Nama : Hermizahadiwidastra  
NIM : 192420035  
Jurusan/Kelas : MTI 21 / AR2  
Mata Kuliah : Advanced is Analysis and Design  
Dosen : Tri Basuki Kurniawan, S.Kom., M.Eng., Ph.D.

### Soal

1. Tentukan pilihan anda sendiri, apakah lebih cenderung ke plan-drive development ataukah agile development ? jelaskan alasan anda !
2. Buatlah satu desain class object yang terlihat didalam sebuah component process menggunakan uml notaion !
3. Apa saran anda untuk mengurangi biaya pada tahapan validasi dan evaluasi sebuah pembangunan software?

### Jawab :

1. Menurut saya cenderung pada agile development, dikarenakan keunggulan agile development adalah :
  - 1) Meningkatkan kepuasan kepada klien.
  - 2) Pembangunan system dibuat lebih cepat.
  - 3) Mengurangi resiko kegagalan implementasi software dari segi non-teknis.
  - 4) Jika pada saat pembangunan system terjadi kegagalan, kerugian dari segi materi relative kecil.
  - 5) Bisa melakukan review pelanggan mengenai software yang di buat lebih awal.

Dan kekurangannya :

- 1) Total lama pengembangan menjadi lebih lama.
- 2) Meningkatkan resiko kesalahan teknis.
- 3) Proses pengembangan menjadi agak kurang terorganisir.

Dibanding kekurangannya dengan plan-drive development, saya rasa agile developmen lebih baik untuk kepuasan pada klien, berikut kekurangan dari plan-drive development :

- 1) Tidak bisa dilakukan perubahan terhadap perangkat lunak di tengah-tengah pekerjaan. Sehingga tidak bisa menyesuaikan kemajuan dan keinginan masyarakat saat itu.
  - 2) Hasilnya kurang sesuai dengan kebutuhan dan keinginan masyarakat pada saat perangkat lunak tersebut selesai.
3. Saran untuk mengurangi biaya pada tahapan validasi dan evaluasi sebuah pembangunan software.
    - 1) Dalam membudayakan dokumentasi dalam proses pengembangan perangkat lunak VTD, perlu adanya komitmen antara project manager dengan anggota tim.
    - 2) Perlu adanya peningkatan kualitas proses pada area konfigurasi manajemen, integrasi produk, kebutuhan pengembangan, dan manajemen resiko.
    - 3) Resiko dalam mengembangkan perangkat lunak pada VTD dapat diminimalisir bahkan dihilangkan apabila ada monitoring dan pengendalian selama proses pengembangan perangkat lunak.

**Pertanyaan :**

**1. Tentukan pilihan anda sendiri apakah lebih cenderung ke Plain-Driven Development ataupun Agile Development? Jelaskan alasan anda!**

Jawab :

Saya memilih Agile Development Methods dengan penjelasan sebagai berikut :

**Agile Development Methods** adalah sekelompok metodologi pengembangan perangkat lunak yang didasarkan pada prinsip-prinsip yang sama atau pengembangan sistem jangka pendek yang memerlukan adaptasi cepat dari pengembang terhadap perubahan dalam bentuk apapun.

Fitur-fitur yang diberikan oleh Agile Software Development adalah sebagai berikut:

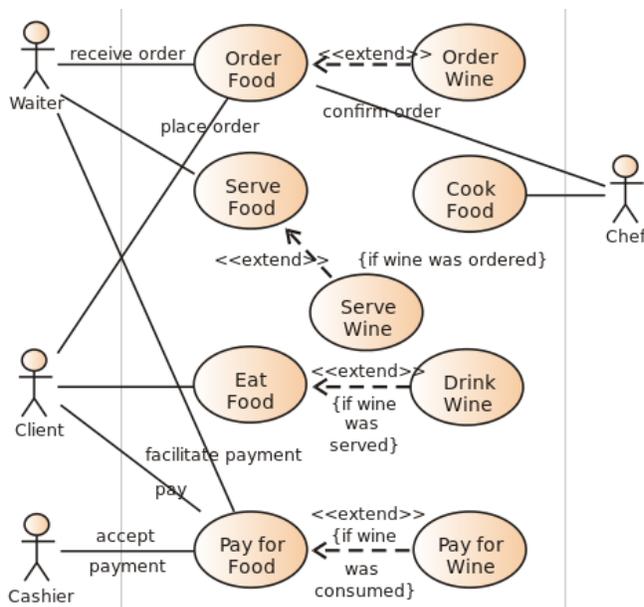
1. Literasi yang cepat dan pengiriman software yang berfungsi secara reguler memastikan kepuasan pelanggan
2. Perubahan yang telat dapat ditangani dengan mudah dan juga diterima secara terbuka
3. Perkembangan dinilai berdasarkan implementasi software
4. Komunikasi pelanggan dan pengguna ditekankan secara bertatap muka
5. Setiap pertemuan dengan anggota tim dilakukan secara bertatap muka
6. Setiap anggota tim pengembang adalah orang yang berkomitmen dan bermotivasi tinggi serta kompeten dan dapat dipercaya

**Kelebihan :**

1. Meningkatkan rasio kepuasan pelanggan
2. Bisa melakukan review pelanggan mengenai software yang dibuat lebih awal
3. Mengurangi resiko kegagalan implementasi software dari segi non-teknis
4. Besar kerugian baik secara material atau imaterial tidak terlalu besar jika terjadi kegagalan

**2. Buatlah satu design class object yang terlibat di dalam sebuah component process menggunakan UML notaion!**

Jawab :



**3. Apa saran anda untuk mengurangi biaya pada tahapan validasi dan evaluasi sebuah pembangunan *software*?**

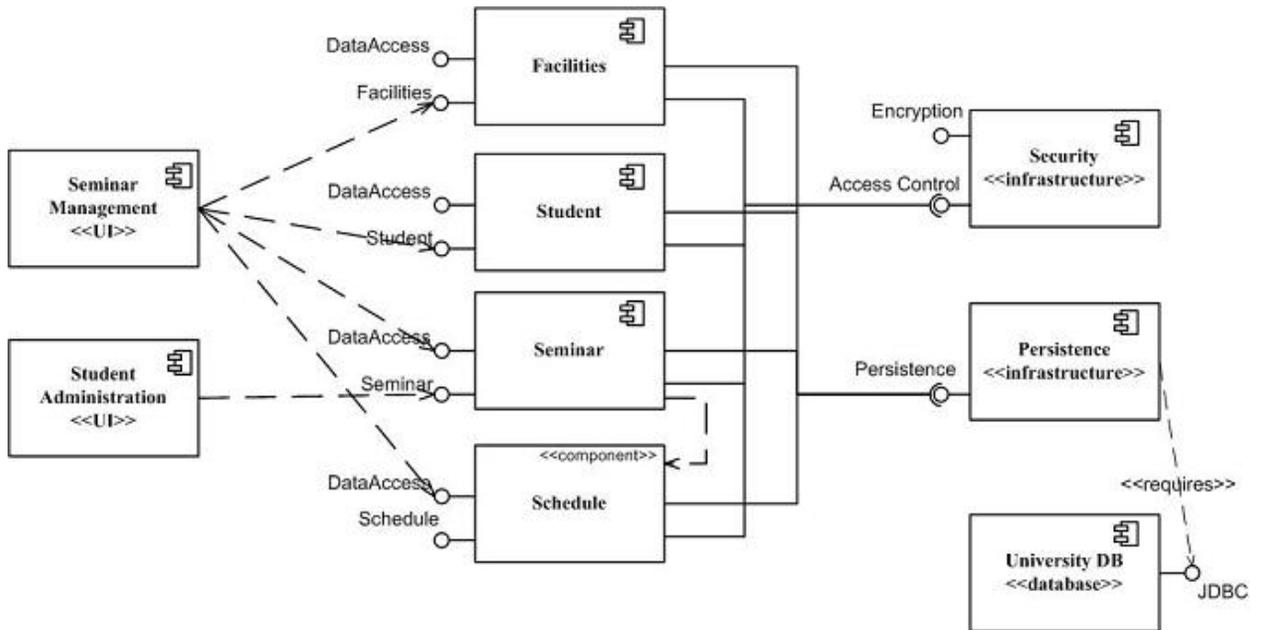
**Jawab :**

- Menguji coba dan mengevaluasi prototype bersama-sama dengan pemakai untuk mendapatkan umpan balik yang dapat membantu pengembang memperbaiki prototype yang sesuai dengan kebutuhan pemakai
- Mengimplemtasikan persyaratan user yang paling penting

1. Saya lebih memilih plan drive development karena Metode plan driven cocok digunakan untuk berbagai hal, antara lain :

- Jika customer lepas tangan terhadap pembuatan perangkat lunak, customer tersebut tidak mau ikut campur dalam pembuatannya, baik itu karena alasan sibuk, malas dan sebagainya.
- Timnya besar (terdiri dari banyak orang).
- Analisisnya besar.
- Perangkat lunak tersebut dirancang agar bisa digunakan dalam tempo yang lama, tidak hanya 1 atau 2 tahun saja.
- Teknologi yang tersedia kurang mendukung.
- Tim pengembangnya tersebar dimana-mana, tidak harus kumpul satu lokasi agar bisa membuat suatu perangkat lunak

2. desain component



3. Saran saya untuk mengurangi biaya pada tahap validasi dan evaluasi sebuah pembangunan software, sebelum pembangunan software harus ada analisis kebutuhan yang bertujuan

1. Mengelola hasil elistasi kebutuhan untuk menghasilkan dokumen spesifikasi kebutuhan yang isi keseluruhannya sesuai dengan apa yang diinginkan pengguna.
2. Mengembangkan persyaratan kualitas yang memadai dan rinci, dimana para manajer dapat membuat pekerjaan proyek yang realistis dan staf teknis dapat melanjutkan dengan perancangan, implementasi dan pengujian.

3. Membangun pemahaman tentang karakteristik ranah permasalahan dan sekumpulan kebutuhan untuk menemukan solusi.

Ketiga tujuan tersebut dapat dicapai oleh perekayasa kebutuhan dengan melalui serangkaian tahapan-tahapan aktivitas. Tahapan aktivitas tersebut ialah **Tahap Analisis Kebutuhan dimana** pengumpulan kebutuhan-kebutuhan dari semua elemen sistem perangkat lunak yang akan di bangun. Pada tahap ini dibentuk spesifikasi kebutuhan perangkat lunak, fungsi perangkat lunak yang dibutuhkan, performansi (unjuk kerja) sistem perangkat lunak, penjadwalan proyek, identifikasi sumber daya (manusia , perangkat keras dan perangkat lunak yang dibutuhkan) dan taksiran biaya pengembangan perangkat lunak.

Apabila semua kebutuhan telah terperinci, maka proses pembangunan software dan study kelayakan berjalan sesuai dengan kebutuhan, maka tidak ada hambatan pada saat validasi software dimana semua pembuktian dari proses, prosedur, kegiatan, pelengkapan yang di inginkan oleh pengguna sudah sesuai dengan kebutuhan pengguna dengan cara ini dapat menimalisir biaya validasi dan evaluasi terhadap suatu pembangunan software.

Nama : Istiqomah Febrianty

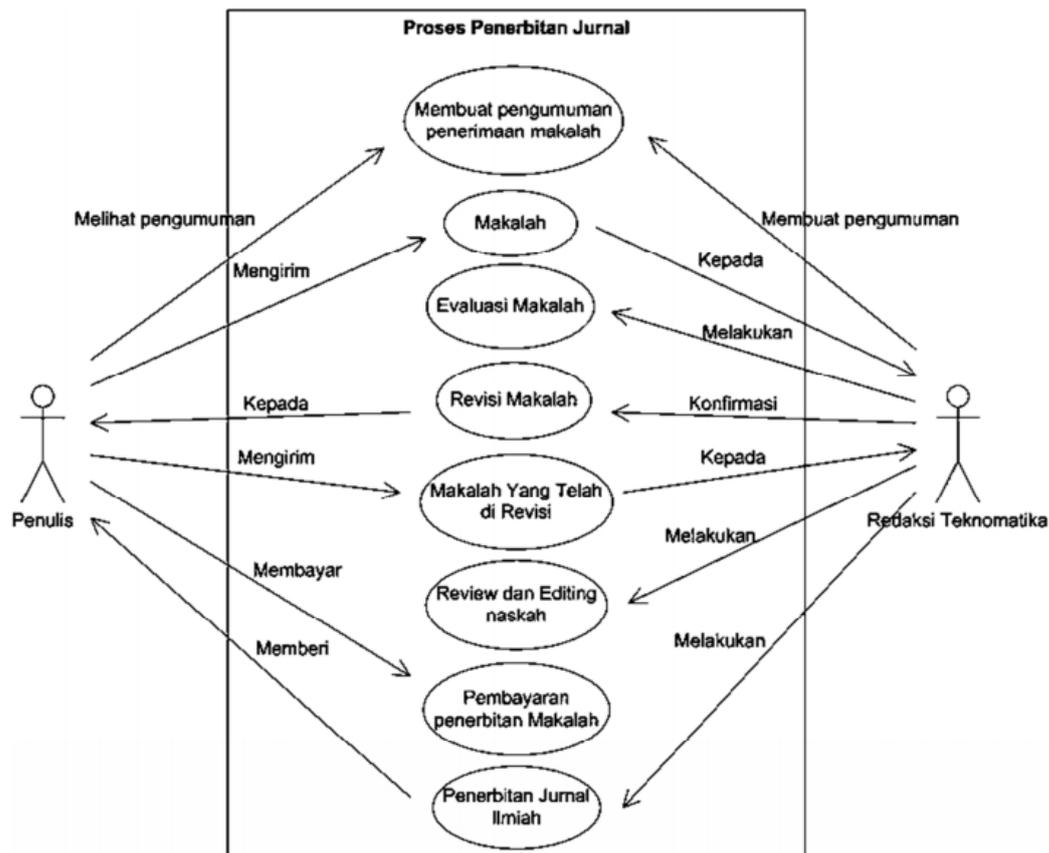
NIM : 192420042

Jawaban soal no. 1

Tidak ada alasan yang didorong oleh rencana dalam lingkungan bisnis ini. Rencana terus berubah. Jadi, satu-satunya hal yang cocok adalah pengembangan yang gesit. Agile adalah cara untuk memperbaiki produk secara berulang. Alasan utama Agile unggul dalam rencana yang didorong adalah:

- Model berulang - Pekerjaan dilakukan dalam semburan singkat dua hingga tiga minggu pada akhir aplikasi yang berpotensi digunakan ditampilkan kepada pengguna akhir. Umpan balik yang diberikan oleh pengguna akhir digunakan untuk mengkalibrasi iterasi berikutnya dan seterusnya.
- Umpan balik di akhir iterasi, perubahan tambahan perlu diperagakan kepada pengguna akhir yang kemudian memberikan umpan balik yang berharga. Ini dapat diambil untuk meningkatkan iterasi di masa depan.
- Pekerjaan untuk iterasi didefinisikan dengan jelas dalam bentuk sprint backlog. ini termasuk cerita dan bug yang perlu diselesaikan sebagai bagian dari sprint. Tidak ada pekerjaan baru yang bisa masuk ke dalam simpanan saat iterasi berjalan.

Jawaban soal no. 2 Diagram Proses Penerbitan Jurnal



Activity Diagram Proses Penerbitan Jurnal

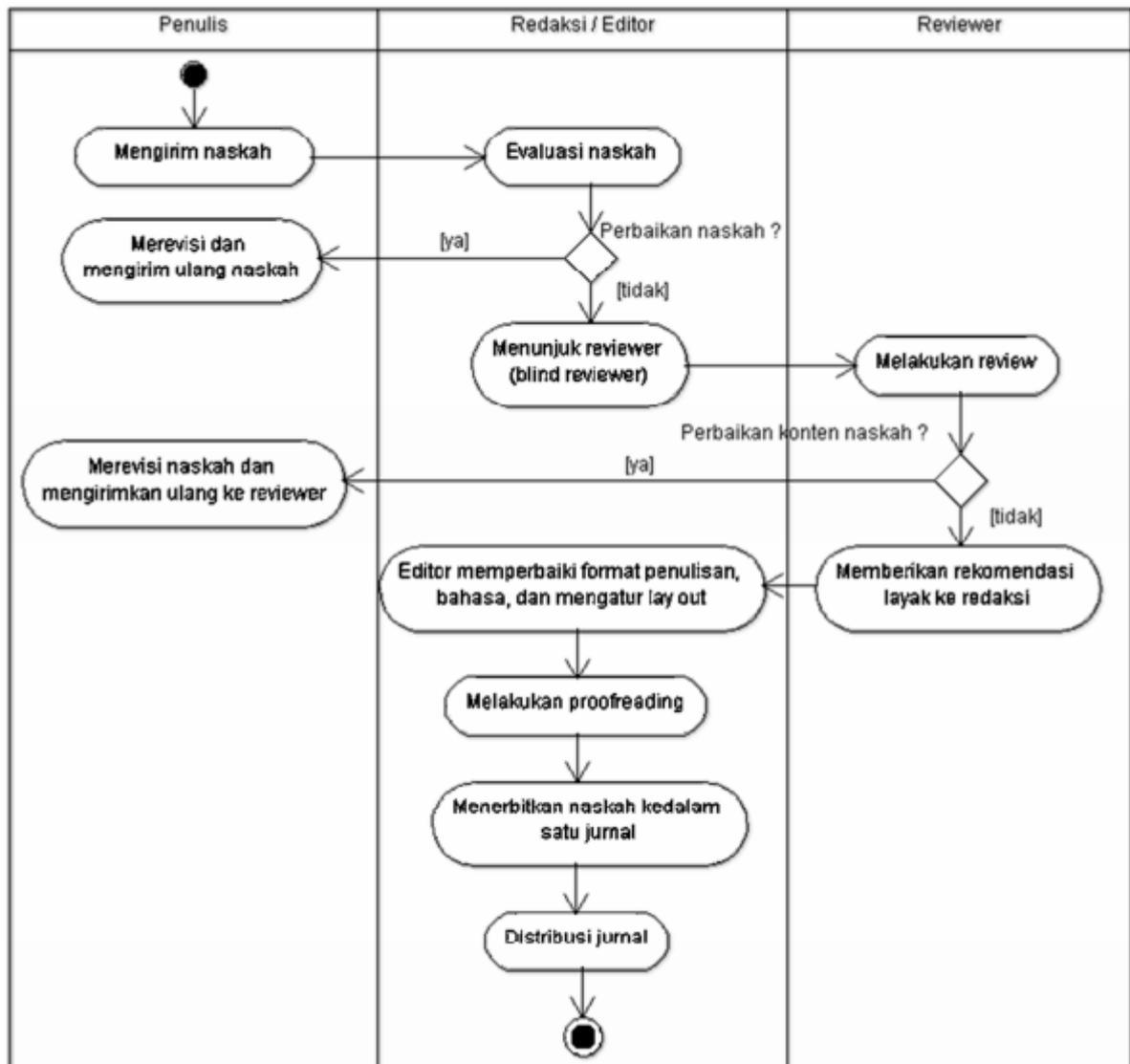
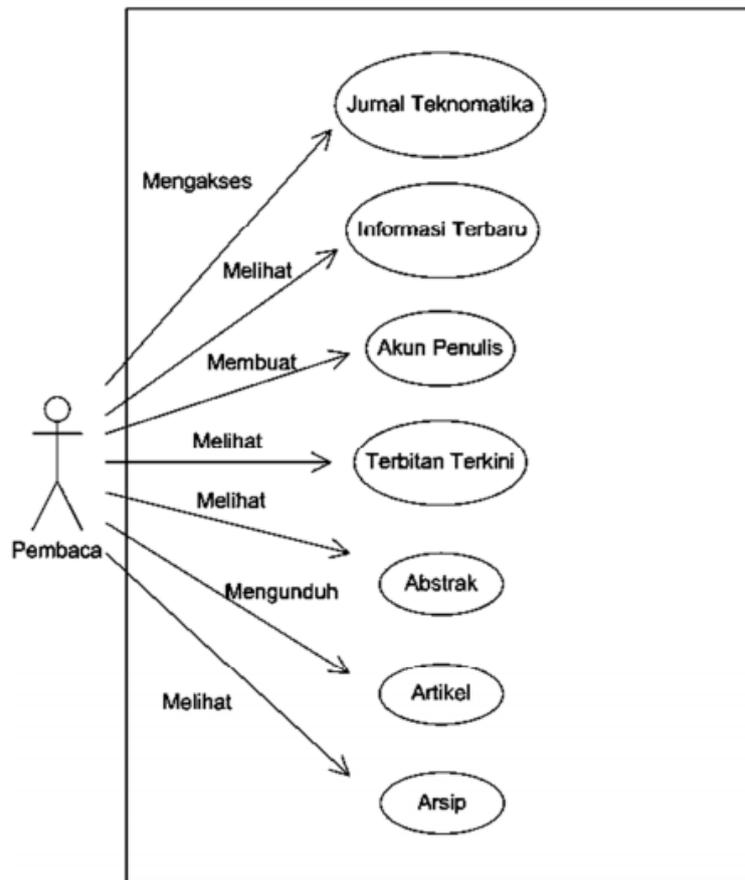


Diagram Pembaca



Jawaban soal no. 3

Untuk mengurangi biaya prototype dan mempercepat jadwal penyerahan, anda dapat tidak memasukkan beberapa fungsionalitas dari prototype. Anda bisa memutuskan untuk melonggarkan persyaratan non-fungsional seperti waktu tanggap dan pemakaian memori. Penanganan dan manajemen error dapat diabaikan atau bersifat mendasar, kecuali jika tujuan prototype adalah menentukan interface user. Standar keandalan dan kualitas program dapat dikurangi.

1. Tentukan pilihan anda sendiri, apakah lebih cenderung ke plan-driven development ataukah agile development ? jelaskan alasan anda !
2. buatlah satu design class object yang terlibat dalam sebuah component process menggunakan UML notation!
3. Apa saran anda untuk mengurangi biaya pada tahapan validasi dan evaluasi sebuah pembangunan software

## **JAWAB :**

1. **Agile development, alasannya**

### KETERLIBATAN STAKEHOLDER

Agile memberikan banyak kesempatan dalam keterlibatan stakeholder dan team - sebelum, selama dan setelah setiap Sprint. Dengan melibatkan client di setiap langkah proyek, ada kolaborasi yang tinggi antara client dan team proyek. Hal ini juga memberi lebih banyak kesempatan bagi team untuk benar-benar memahami visi client. Mempresentasikan software yang bekerja dengan lebih awal dan sering meningkatkan kepercayaan para stakeholder dalam kemampuan team untuk membuat software berkualitas tinggi dan mendorong mereka untuk lebih terlibat dalam proyek ini.

### TRANSPARANSI

Pendekatan Agile memberikan kesempatan unik bagi client untuk lebih terlibat dalam keseluruhan proyek. Ini beralih dari kemampuan memprioritaskan fitur untuk merencanakan dan meninjau pembaruan software yang berisi fitur baru.

### PENGIRIMAN AWAL & MUDAH DIPREDIKSI

Dengan menggunakan jadwal tetap Sprints 1-4 minggu, fitur baru bisa dibuat dengan cepat dan sering, juga dengan tingkat prediktabilitas yang tinggi. Ini juga memberi kesempatan untuk memasang atau menguji beta dari software lebih awal dari yang direncanakan jika ada nilai bisnis yang cukup.

### BIAYA & JADWAL YANG BISA DITEBAK

Karena setiap Sprint memiliki durasi yang tetap, biayanya dapat diprediksi dan sudah ditentukan di dalam pekerjaan yang akan dilakukan dalam jangka waktu yang ada membuat mudahnya perkiraan biaya yang ada. Dikombinasikan dengan perkiraan yang diberikan kepada client sebelum setiap Sprint, client dapat lebih mudah memahami biaya setiap fitur yang meningkatkan kemampuan pengambilan keputusan mengenai prioritas setiap fitur.

### MEMUNGKINKAN PERUBAHAN

Sementara team perlu tetap fokus untuk memberikan subset fitur produk yang disepakati di setiap Sprint, ada kesempatan untuk terus memperbaiki dan memprioritaskan isi backlog produk. Item backlog baru atau yang telah diubah dapat direncanakan dalam pertemuan Sprint berikutnya, memberikan kesempatan untuk mengenalkan perubahan dalam beberapa minggu.

## FOKUS PENGGUNA

Agile biasanya menggunakan feedback user dengan kriteria yang berfokus pada bisnis untuk menentukan fitur produk. Dengan memfokuskan fitur pada kebutuhan pengguna nyata, setiap fitur memberikan nilai, tidak hanya sekedar komponen IT. Ini juga memberikan kesempatan untuk menguji beta software setelah setiap Sprint, mendapatkan umpan balik yang berharga di awal proyek dan memberikan kemampuan untuk melakukan perubahan seperlunya.

## MENINGKATKAN KUALITAS

Dengan membagi proyek menjadi unit-unit yang dapat dikelola, team proyek dapat berfokus pada pengembangan, pengujian dan kolaborasi berkualitas tinggi. Selain itu, dengan memproduksi sering membangun dan melakukan pengujian dan ulasan selama setiap Sprint, kualitas ditingkatkan dengan menemukan dan memperbaiki cacat dengan cepat dan mengidentifikasi kesalahan ekspektasi.

2.

3. Dengan menggunakan pendekatan Agile development dapat mengurangi cost pada saat evaluasi dan validasi dikarenakan proses evaluasi dan validasi telah dilakukan pada setiap sprint sehingga tidak dibutuhkan biaya dan waktu yang lama untuk melakukan validasi dan evaluasi akhir.

Nama : M Danial Sentosa

Nim : 192420040

MTI kelas A R2

1. Tentukan pilihan anda sendiri, apakah lebih cenderung ke plan-drive development ataukah agile development ? jelaskan alasan anda !
2. Buatlah satu desain class object yang terlihat didalam sebuah component process menggunakan uml notaion !
3. Apa saran anda untuk mengurangkan biaya pada tahapan validasi dan evaluasi sebuah pembangunan software?

Jawab

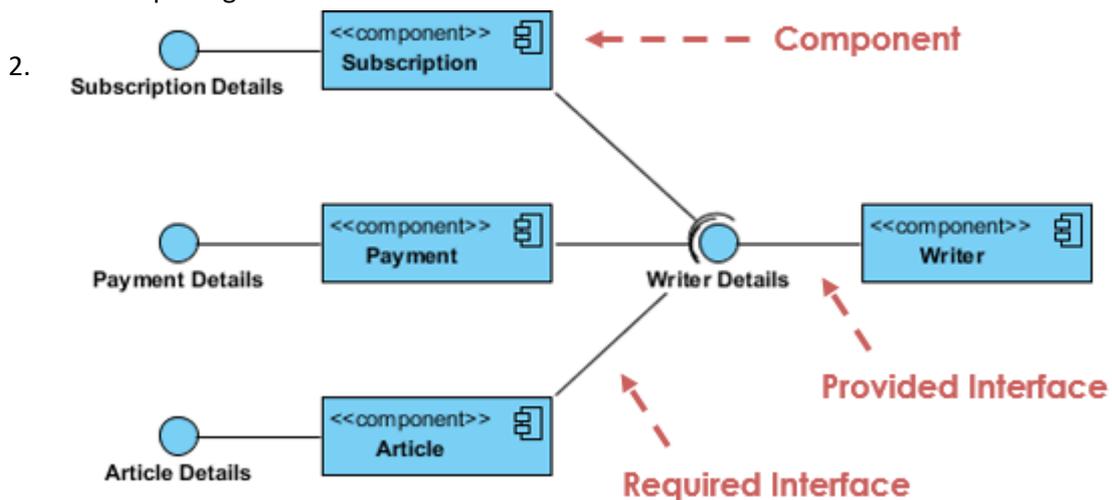
1. Menurut saya cenderung pada agile development, dikarenakan keunggulan agile development adalah :
  - Meningkatkan kepuasan kepada klien
  - Pembangunan system dibuat lebih cepat
  - Mengurangi resiko kegagalan implementasi software dari segi non-teknis
  - Jika pada saat pembangunan system terjadi kegagalan, kerugian dari segi materi relative kecil.
  - Bisa melakukan review pelanggan mengenai software yang di buat lebih awal.

Dan kekurangannya :

- Total lama pengembangan menjadi lebih lama
- Meningkatkan resiko kesalahan teknis
- Proses pengembangan menjadi agak kurang terorganisir

Dibanding kekurangannya dengan plan-drive development, saya rasa agile development lebih baik untuk kepuasan pada klien, berikut kekurangan dari plan-drive development :

- Tidak bisa dilakukan perubahan terhadap perangkat lunak di tengah-tengah pekerjaan. Sehingga tidak bisa menyesuaikan kemajuan dan keinginan masyarakat saat itu.
- Hasilnya kurang sesuai dengan kebutuhan dan keinginan masyarakat pada saat perangkat lunak tersebut selesai.



3. saran untuk mengurangi biaya pada tahapan validasi dan evaluasi sebuah pembangunan software
  - Dalam membudayakan dokumentasi dalam proses pengembangan perangkat lunak VTD, perlu adanya komitmen antara project manager dengan anggota tim.
  - Perlu adanya peningkatan kualitas proses pada area konfigurasi manajemen, integrasi produk, kebutuhan pengembangan, dan manajemen resiko.
  - Resiko dalam mengembangkan perangkat lunak pada VTD dapat diminimalisir bahkan dihilangkan apabila ada monitoring dan pengendalian selama proses pengembangan perangkat lunak.

Nama : Muhammad Iqbal Rizky Tanjung  
Nim : 192420045  
Kelas : MTI Reg B1

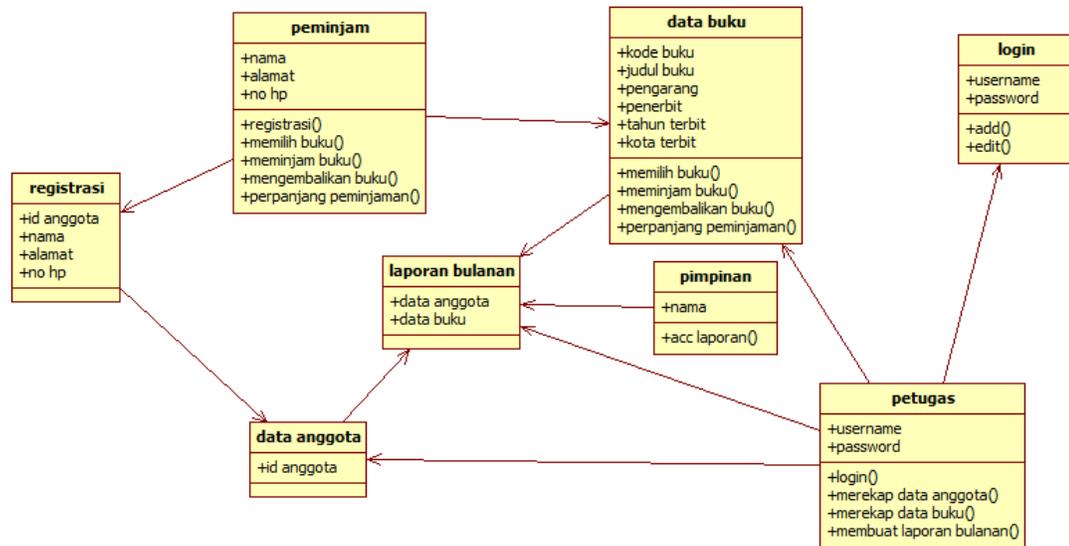
**Pertanyaan :**

1. apakah lebih cenderung ke plain drive Development atau Agile Development ?
2. Buatlah desain class Objek yang terlibat dalam sebuah komponentproses menggunakan UML Notation?
3. Apa saran anda untuk mengurangi Biaya pada tahapan validasi dan evaluasi sebuah pembangunan Software?

**Jawaban :**

1. Saya lebih cenderung memakai Agile Development karna, Menurut *Agile Alliance*, terdapat 12 prinsip yang harus dipenuhi bagi mereka yang ingin berhasil dalam penerapan *Agile Software Development*. Prinsip-prinsip tersebut antara lain :
  - a) Kepuasan klien adalah prioritas utama dengan menghasilkan produk lebih awal dan terus menerus.
  - b) Menerima perubahan kebutuhan, sekalipun diakhir pengembangan.
  - c) Penyerahan hasil/software dalam hitungan waktu dua minggu sampai dua bulan.
  - d) Bagian bisnis dan pembangun kerja sama tiap hari selama proyek berlangsung.
  - e) Membangun proyek dilingkungan orang-orang yang bermotivasi tinggi yang bekerja dalam lingkungan yang mendukung dan yang dipercaya untuk dapat menyelesaikan proyek.
  - f) Komunikasi dengan berhadapan langsung adalah komunikasi yang efektif dan efisien.
  - g) Software yang berfungsi adalah ukuran utama dari kemajuan proyek.
  - h) Dukungan yang stabil dari sponsor, pembangun, dan pengguna diperlukan untuk menjaga perkembangan yang berkesinambungan.
  - i) Perhatian kepada kehebatan teknis dan desain yang bagus meningkatkan sifat agile.
  - j) Kesederhanaan sangat penting.
  - k) Arsitek. Kebutuhan dan desain yang bagus muncul dari tim yang mengatur dirinya sendiri.
  - l) Secara periodik tim evaluasi diri dan mencari cara untuk lebih efektif dan segera melakukannya.

## 2. Class Object:



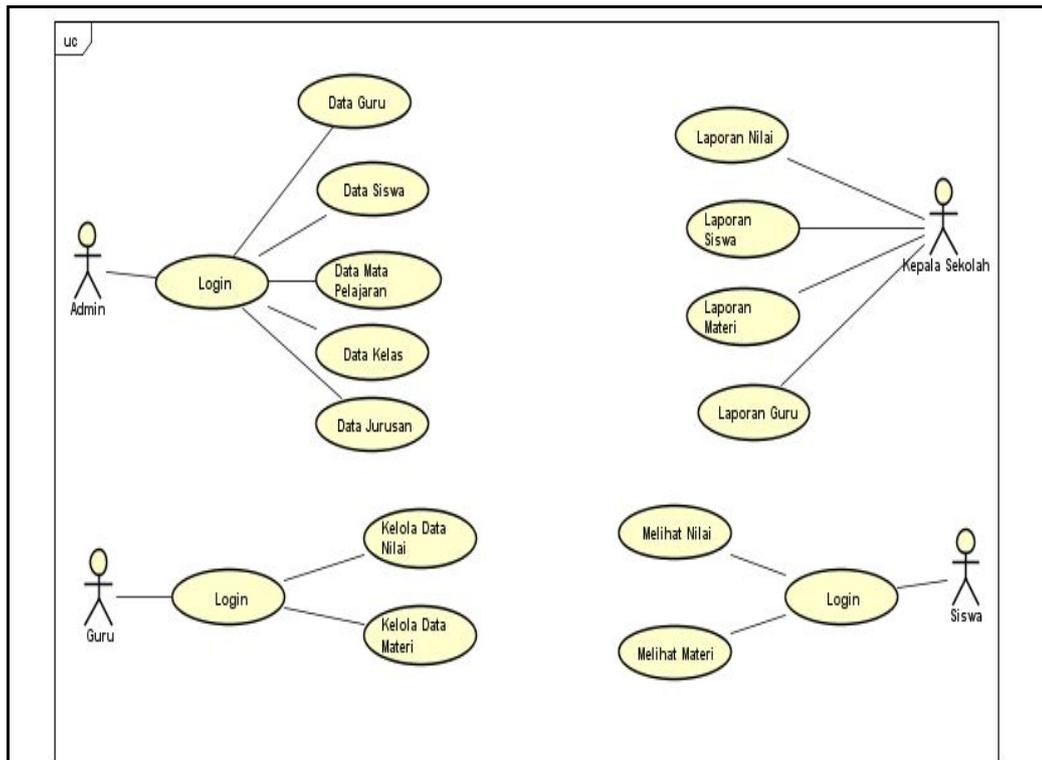
## 3. Saran Saya untuk mengurangi biaya pada tahapan validasi dan evaluasi adalah:

- Pengurangan Bug pada system Software sehingga Bisa mengurangi Banyak nya tahap Testing
- Dalam pengembangan system Software Harus Lebih Teliti Sehingga dapat menghindari banyak nya perubahan system Software
- Sebelum ke tahap pengembangan harus lebih Teliti dalam membuat Kontark pembangunan Software ini berisikan Point –point dalam pengurangan biaya
- Mengetahui kebutuhan hal terpenting dalam kebutuhan User

Judul Tugas	Quiz 1 (Rekayasa Perangkat Lunak)
MK	Advanced Is Analysis and Design
Mahasiswa	M. Riski Qisthiano (182420040)
Tanggal	11-12-2019
Dosen	Tri Basuki Kurniawan, Ph.D.

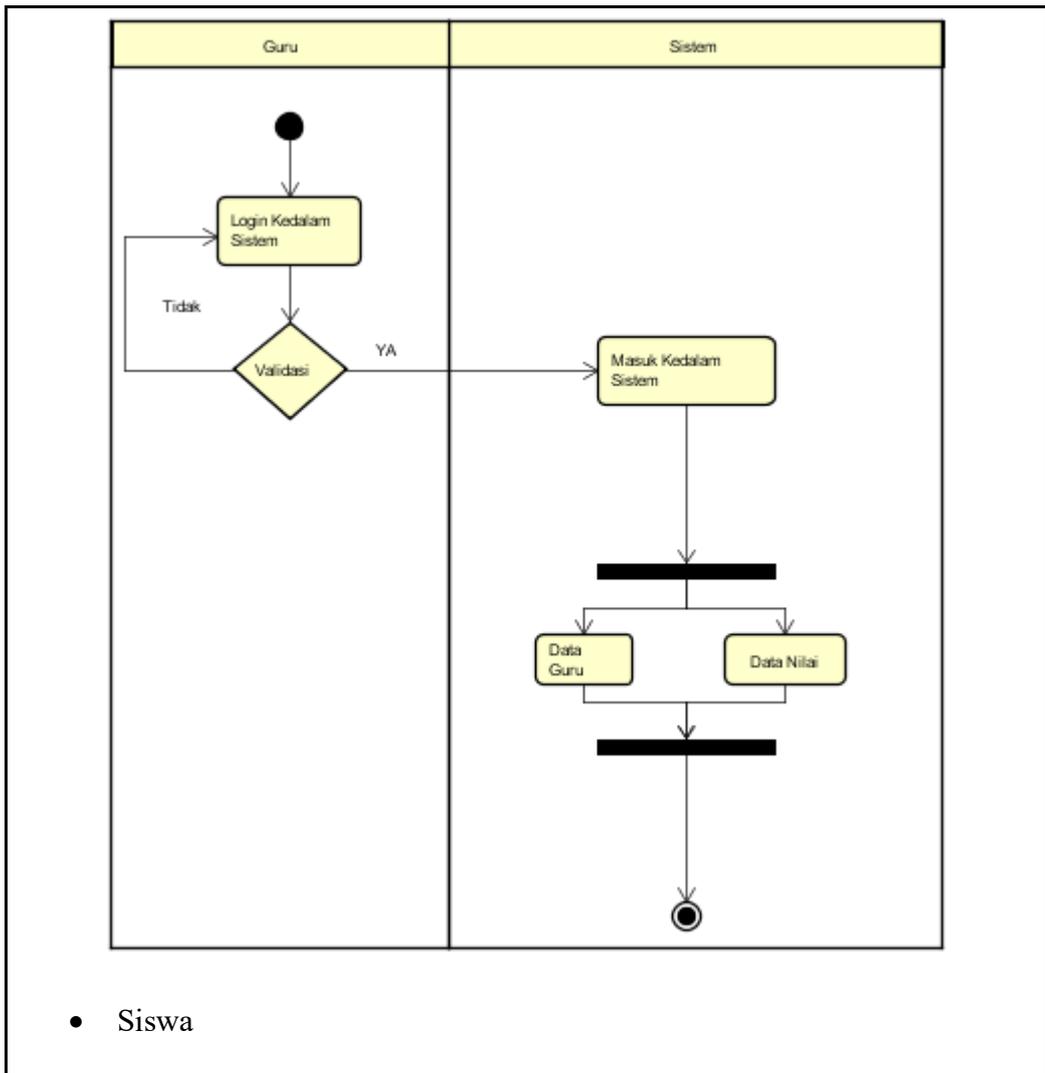
1	Tentukan pilihan anda sendiri, apakah lebih cenderung ke <i>plain-drive development</i> ataukah <i>agile development</i> ? jelaskan alasan anda !
<p>Jawab :</p> <p>Saya lebih cenderung ke <i>agile development</i> karena proses yang cepat selain itu juga semua spesifikasi, desain, implementasi, dan pengujian saling terkait dan output dari proses pengembangan diputuskan melalui proses negosiasi selama proses pengembangan perangkat lunak. Hal ini dapat menguntungkan pihak pengembang serta membuat semua keinginan pelanggan menjadi lebih terakomodir.</p>	

2	Buatlah satu desain <i>class object</i> yang terlibat di suatu <i>component process</i> menggunakan UML notaion !
<p>Jawab :</p> <p><i>Component process</i> berikut ini mengenai proses yang terjadi pada Sistem Informasi Akademik Sekolah.</p> <ol style="list-style-type: none"> <li>1. Use Case Diagram</li> </ol>	

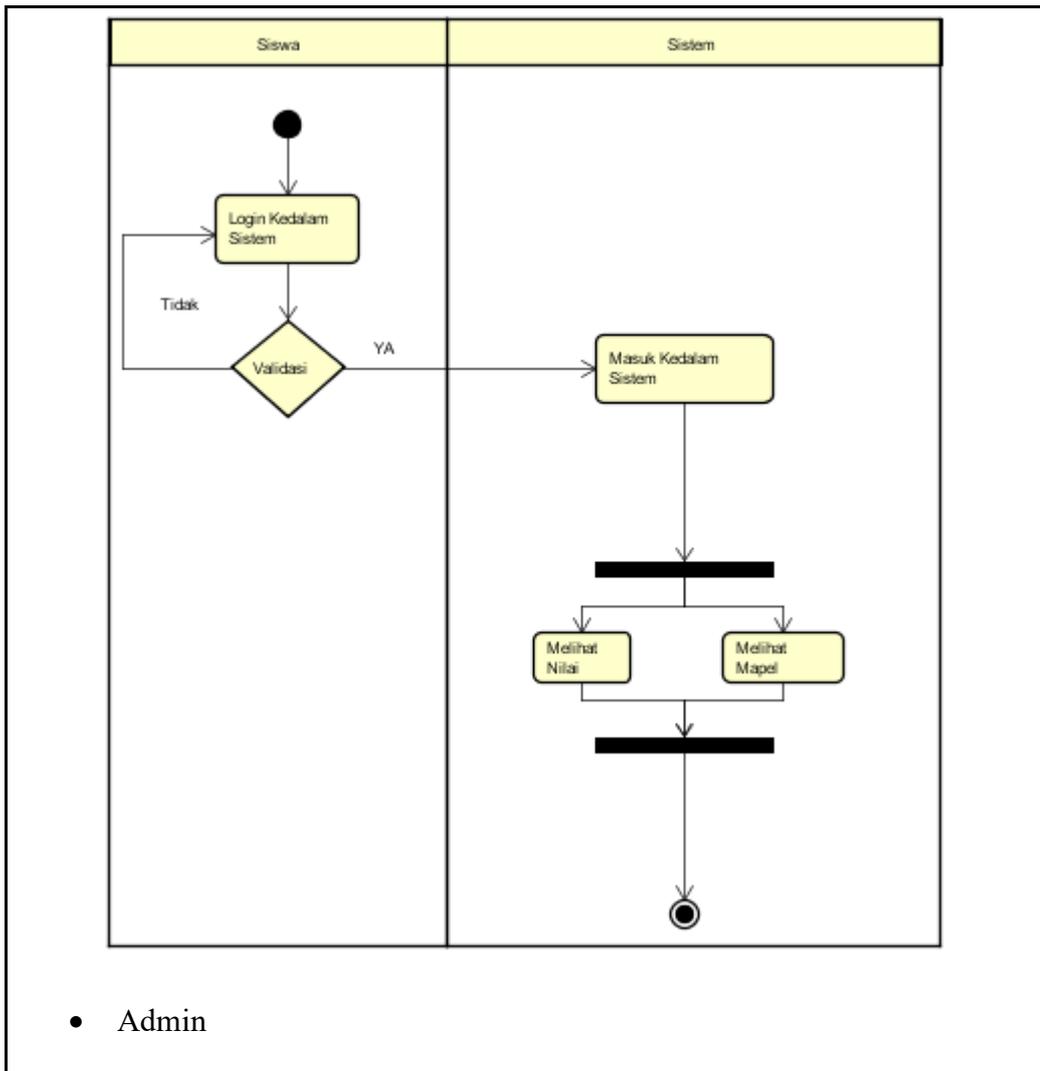


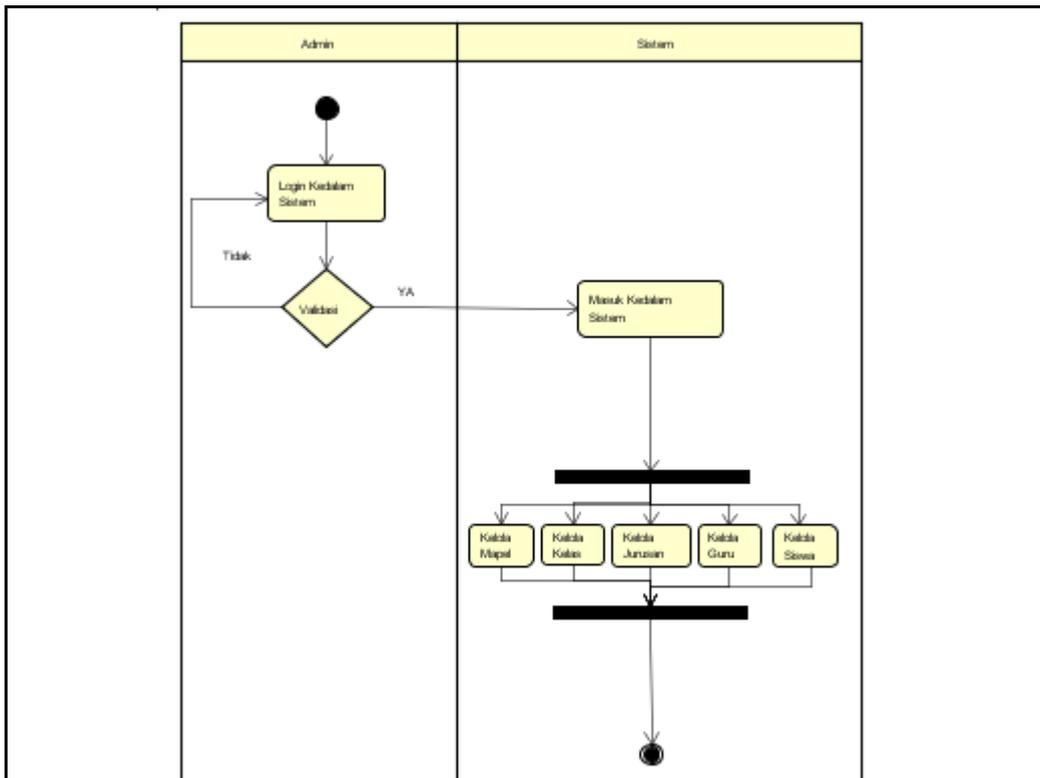
## 2. Activity Diagram

- Guru

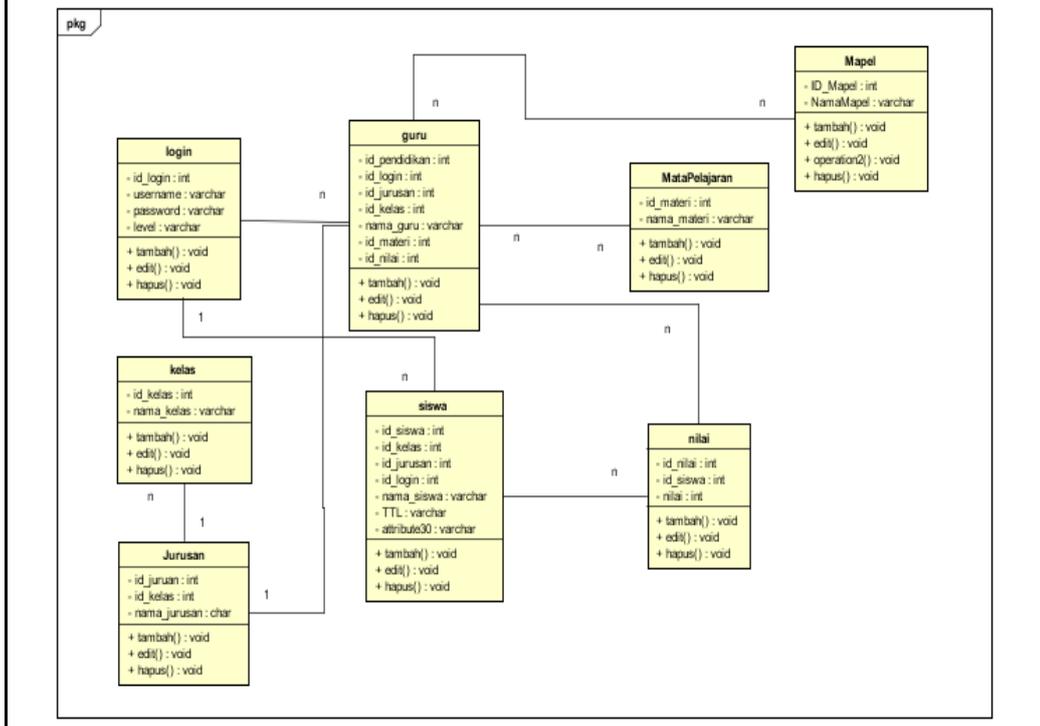


- Siswa





### 3. Class Diagram



3	Apa saran anda untuk mengurangi biaya pada tahap validasi dan evaluasi pada pembangunan sebuah <i>software</i> ?
<p>Jawab :</p> <p>Untuk mengurangi banyaknya pengeluaran disarankan jangan terlalu banyak perubahan, buatlah perancangan yang tepat sehingga mengurangi perubahan, ketiga kurangi banyak bug yang dapat membuat testing berulang-ulang</p>	

Nama : M.Afdhaluddin (192420012)  
Kels : Reguler A R1  
Mata kuliah : Advanced Analysis AndDesign  
Dosen : Tri Basuki Kurniawan, S.Kom., M.Eng

1. Tentukan Pilihan sendiri , apakah lebih cenderung ke *Plain- Drive Development* ataukah *Agile Depeloment*? Jelaskan alasan anda
2. Buatlah satu desain *class object* yang terlibat didalam sebuah *component process* menggunakan UML notaion!
3. Apa saran anda untuk mengurangkan biaya pada tahapan validasi dan evaluasi sebuah pembangunan *software*?

Jawaban

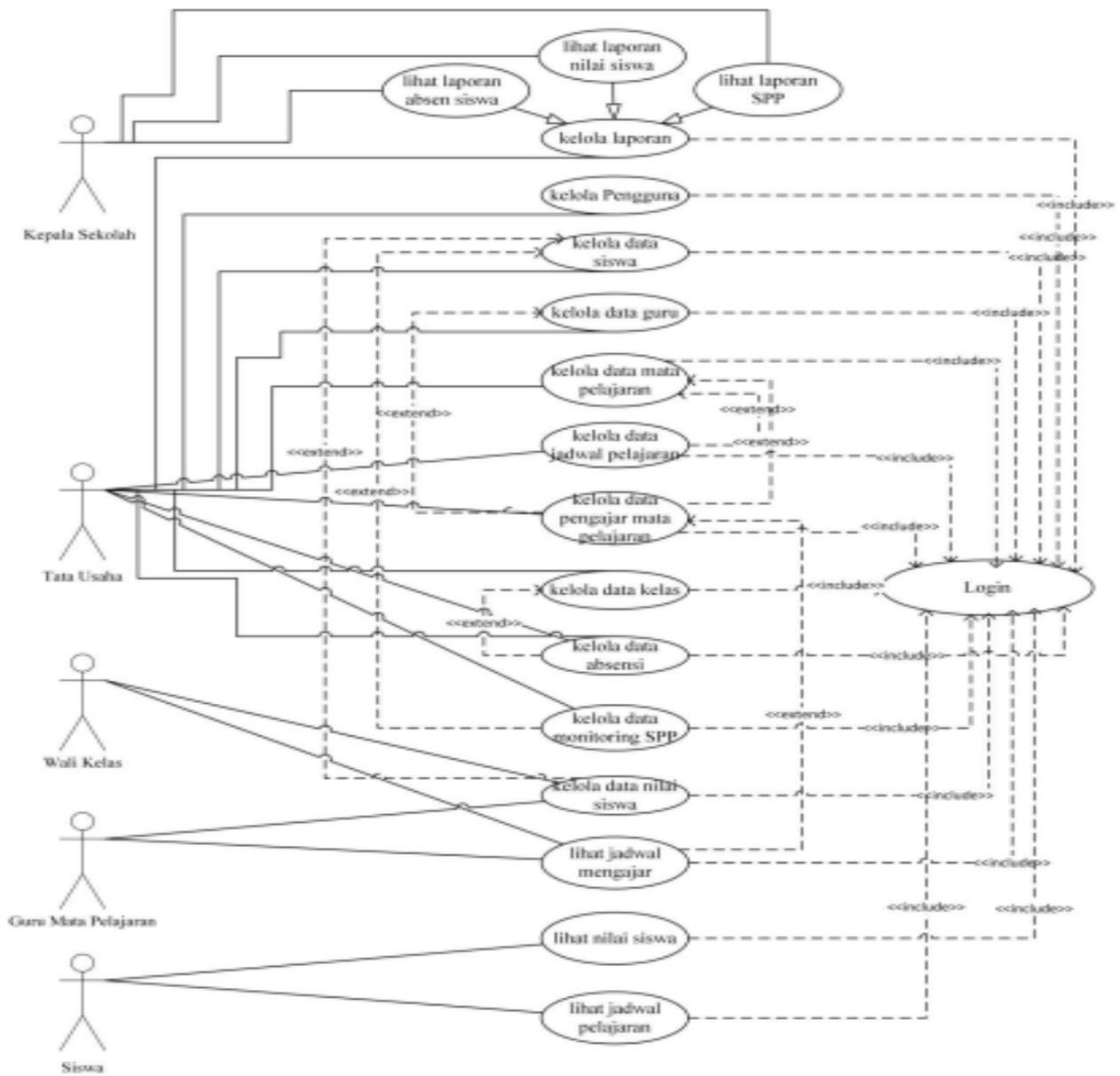
1. Saya memilih *Agile Depeloment* dikarenakan

Terstruktur tahap perancangan dari perencanaan hingga pemeliharaan

- Perencanaan, pada langkah ini pengembang dan klien membuat rencana tentang kebutuhan dari perangkat lunak yang akan dibuat.
- Implementasi, bagian dari proses dimana programmer melakukan pengkodean perangkat lunak.
- Tes perangkat lunak, disini perangkat lunak yang telah dibuat di tes oleh bagian kontrol kualitas agar bug yang ditemukan bisa segera diperbaiki dan kualitas perangkat lunak terjaga.
- Dokumentasi, setelah dilakukan tes perangkat lunak langkah selanjutnya yaitu proses dokumentasi perangkat lunak untuk mempermudah proses maintenance kedepannya.
- Deployment, yaitu proses yang dilakukan oleh penjamin kualitas untuk menguji kualitas sistem. Setelah sistem memenuhi syarat maka perangkat lunak siap dideployment.
- Pemeliharaan, langkah terakhir yaitu pemeliharaan. Tidak ada perangkat lunak yang 100% bebas dari bug, oleh karena itu sangatlah penting agar perangkat lunak dipelihara secara berkala.

Memprioritaskan kepuasan klien dan meningkatkan kerja sama tim dalam suatu pembangunan perangkat lunak.

2. Class komponen proses menggunakan UML notaion



3. mengurangi biaya validasi dan evaluasi software

- Mengimplemtasikan persyaratan user yang paling penting
- Melonggarkan persyaratan non fungsional seperti waktu dan tanggap dan pemakaian memori
- Disisikan waktu untuk pelatihan user dan tujuan protipe, agar user dapat menemukan error dalam menggunakan software

## QUIZ 1

1. Tentukan pilihan anda sendiri, apakah lebih cenderung ke plan-driven development ataukah agile development ? jelaskan alasan anda !
2. buatlah satu design class object yang terlibat dalam sebuah component process menggunakan UML notation!
3. Apa saran anda untuk mengurangi biaya pada tahapan validasi dan evaluasi sebuah pembangunan software

JAWAB :

### 1. Agile development, alasannya

- Meningkatkan rasio kepuasan pelanggan.
- Bisa melakukan rewiw pelanggan mengenai software yang dibuat lebih awal.
- Mengurangi resiko kegagalan implementasi software dari non-teknis.
- Besar kerugian baik secara material atau imaterial tidak terlalu besar jika terjadi kegagalan

Pada Agile development juga memiliki :

#### KETERLIBATAN STAKEHOLDER

Agile memberikan banyak kesempatan dalam keterlibatan stakeholder dan team - sebelum, selama dan setelah setiap Sprint. Dengan melibatkan client di setiap langkah proyek, ada kolaborasi yang tinggi antara client dan team proyek. Hal ini juga memberi lebih banyak kesempatan bagi team untuk benar-benar memahami visi client. Mempresentasikan software yang bekerja dengan lebih awal dan sering meningkatkan kepercayaan para stakeholder dalam kemampuan team untuk membuat software berkualitas tinggi dan mendorong mereka untuk lebih terlibat dalam proyek ini.

#### TRANSPARANSI

Pendekatan Agile memberikan kesempatan unik bagi client untuk lebih terlibat dalam keseluruhan proyek. Ini beralih dari kemampuan memprioritaskan fitur untuk merencanakan dan meninjau pembaruan software yang berisi fitur baru.

#### PENGIRIMAN AWAL & MUDAH DIPREDIKSI

Dengan menggunakan jadwal tetap Sprints 1-4 minggu, fitur baru bisa dibuat dengan cepat dan sering, juga dengan tingkat prediktabilitas yang tinggi. Ini juga memberi kesempatan untuk memasang atau menguji beta dari software lebih awal dari yang direncanakan jika ada nilai bisnis yang cukup.

#### BIAYA & JADWAL YANG BISA DITEBAK

Nama : Muhammad Fajar  
NIM : 192420037

Karena setiap Sprint memiliki durasi yang tetap, biayanya dapat diprediksi dan sudah ditentukan di dalam pekerjaan yang akan dilakukan dalam jangka waktu yang ada membuat mudahnya perkiraan biaya yang ada. Dikombinasikan dengan perkiraan yang diberikan kepada client sebelum setiap Sprint, client dapat lebih mudah memahami biaya setiap fitur yang meningkatkan kemampuan pengambilan keputusan mengenai prioritas setiap fitur.

#### MEMUNGKINKAN PERUBAHAN

Sementara team perlu tetap fokus untuk memberikan subset fitur produk yang disepakati di setiap Sprint, ada kesempatan untuk terus memperbaiki dan memprioritaskan isi backlog produk. Item backlog baru atau yang telah diubah dapat direncanakan dalam pertemuan Sprint berikutnya, memberikan kesempatan untuk mengenalkan perubahan dalam beberapa minggu.

#### FOKUS PENGGUNA

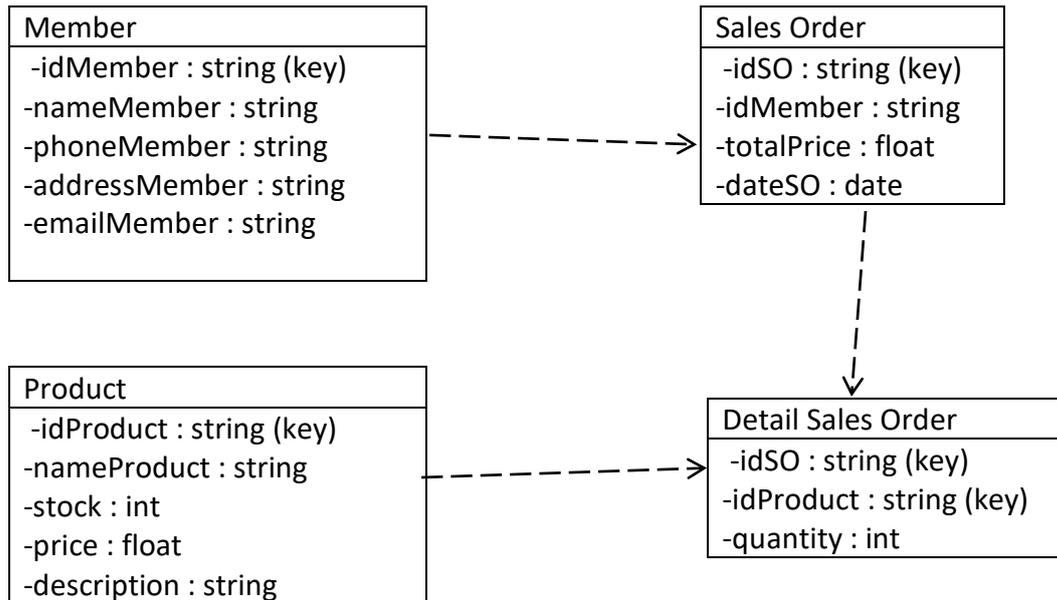
Agile biasanya menggunakan feedback user dengan kriteria yang berfokus pada bisnis untuk menentukan fitur produk. Dengan memfokuskan fitur pada kebutuhan pengguna nyata, setiap fitur memberikan nilai, tidak hanya sekedar komponen IT. Ini juga memberikan kesempatan untuk menguji beta software setelah setiap Sprint, mendapatkan umpan balik yang berharga di awal proyek dan memberikan kemampuan untuk melakukan perubahan seperlunya.

#### MENINGKATKAN KUALITAS

Dengan membagi proyek menjadi unit-unit yang dapat dikelola, team proyek dapat berfokus pada pengembangan, pengujian dan kolaborasi berkualitas tinggi. Selain itu, dengan memproduksi sering membangun dan melakukan pengujian dan ulasan selama setiap Sprint, kualitas ditingkatkan dengan menemukan dan memperbaiki cacat dengan cepat dan mengidentifikasi kesalahan ekspektasi.

Nama : Muhammad Fajar  
NIM : 192420037

2.



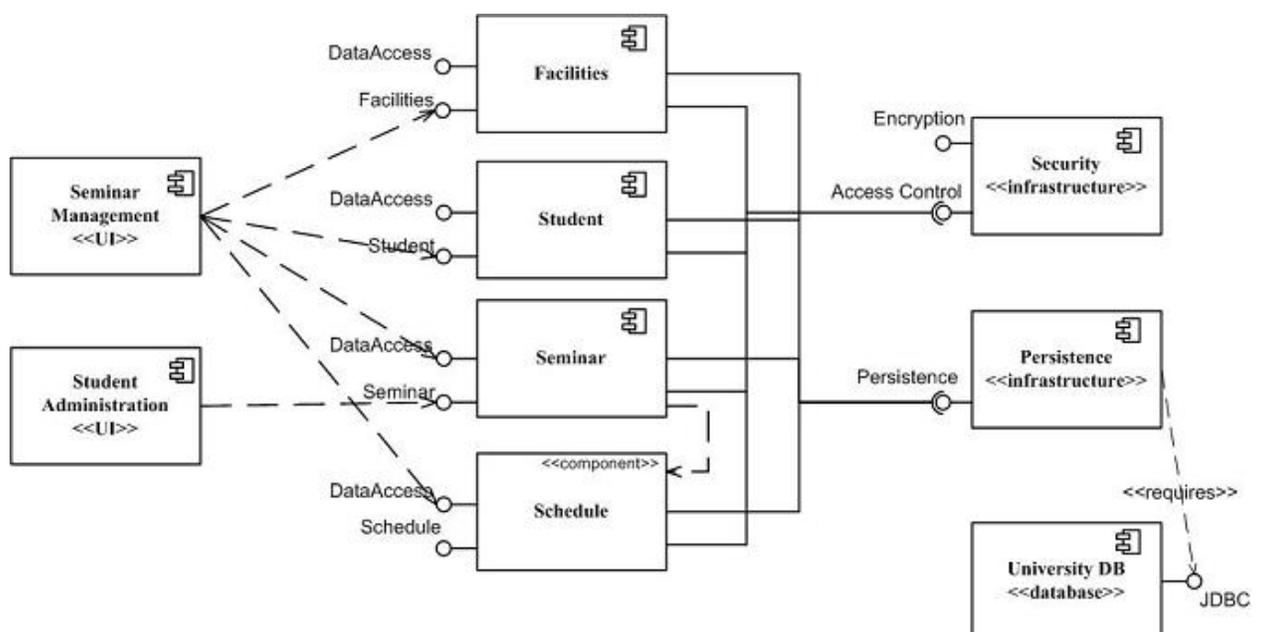
3. Dengan menggunakan pendekatan Agile development dapat mengurangi cost pada saat evaluasi dan validasi dikarenakan proses evaluasi dan validasi telah dilakukan pada setiap sprint sehingga tidak dibutuhkan biaya dan waktu yang lama untuk melakukan validasi dan evaluasi akhir.

1. Tentukan pilihan anda sendiri, apakah lebih cenderung ke *Plain-Drive Development* ataukah *Agile Development*? Jelaskan alasan anda!
2. Buatlah satu desain *class object* yang terlibat di dalam sebuah *component process* menggunakan UML notaion!
3. Apa saran anda untuk mengurangi biaya pada tahapan validasi dan evaluasi sebuah pembangunan *software* ?

Jawaban :

1. Saya lebih memilih plan drive development karena Metode plan driven cocok digunakan untuk berbagai hal, antara lain :
  - Jika customer lepas tangan terhadap pembuatan perangkat lunak, customer tersebut tidak mau ikut campur dalam pembuatannya, baik itu karena alasan sibuk, malas dan sebagainya.
  - Tim nyabesar (terdiri dari banyak orang).
  - Analisisnya besar.
  - Perangkat lunak tersebut dirancang agar bias digunakan dalam tempo yang lama, tidak hanya 1 atau 2 tahun saja.
  - Teknologi yang tersedia kurang mendukung.
  - Tim pengembangnya tersebar dimana-mana, tidak harus kumpul satu lokasi agar bias membuat suatu perangkat lunak

2. desain component



3. Saran saya untuk mengurangi biaya pada tahap validasi dan evaluasi sebuah pembangunan software, sebelum pembangunan software harus ada analisis kebutuhan yang bertujuan

1. Mengelola hasil elistasi kebutuhan untuk menghasilkan dokumen spesifikasi kebutuhan yang isi keseluruhannya sesuai dengan apa yang di inginkan pengguna.
2. Mengembangkan persyaratan kualitas yang memadai dan rinci, dimana paramanager dapat membuat pekerjaan proyek yang realistis dan staf teknis dapat melanjutkan dengan perancangan, implementasi dan pengujian.
3. Membangun pemahaman tentang karakter istikranah permasalahan dan sekumpulan kebutuhan untuk menemukansolusi.

Ketiga tujuan tersebut dapat dicapai oleh perekayasa kebutuhan dengan melalui serangkaian tahapan-tahapan aktivitas.Tahapan aktivitas tersebut ialah **Tahap Analisis Kebutuhan dimana** pengumpulan kebutuhan-kebutuhan dari semua elemen system perangkat lunak yang akan di bangun. Pada tahap ini dibentuk spesifikasi kebutuhan perangkat lunak, fungsi perangkat lunak yang dibutuhkan, performansi (untuk kerja) system perangkat lunak, penjadwalan proyek, identifikasi sumberdaya (manusia ,perangkat keras dan perangkat lunak yang dibutuhkan) dan taksiran biaya pengembangan perangkat lunak.

Apabila semua kebutuhan telah terperinci, maka proses pembangunan software dan study kelayakan berjalan sesuai dengan kebutuhan, maka tidak ada hambatan pada saat validasi software dimana semua pembuktian dari proses, prosedur, kegiatan, pelengkapan yang di inginkan oleh pengguna sudah sesuai dengan kebutuhan pengguna dengan cara ini dapat menimalisir biaya validasi dan evaluasi terhadap suatu pembangunan software.