Pengembangan Aplikasi Pembayaran Online *Mobile Money System* Dengan Menggunakan *Agile* Metodologi dan Teknologi *Software Oriented Architecture*

CANDRA GUNAWAN[1]

[1] *Magister Manajement Sistem Informasi, Universitas Gunadarma, 2016, Jakarta*
[*] *Corresponding author: sherard@outlook.com*

Compiled using LaTeX May 30, 2016

Mobile Money System **adalah suatu aplikasi bisa digunakan untuk berbagai macam pembayaran, mulai dari pembelian pulsa maupun belanja online. Dalam pengembangan aplikasi tersebut, dibutuhkan suatu metode pengembangan sistem yang bisa menjawab kebutuhan pasar yang sangat cepat. Dan juga, dibutuhkan suatu teknologi yang bisa beradaptasi dengan metode pengembangan sistem tersebut.** ***Agile Development Methods*** **adalah sekelompok metodologi pengembangan perangkat lunak yang di dasarkan pada prinsip - prinsip jangka pendek yang memerlukan adaptasi cepat dari pengembang terhadap perubahan dalam bentuk apapun.** ***service oriented architecture*** **(*SOA* adalah suatu gaya arsitektur sistem yang membuat dan menggunakan proses bisnis dalam bentuk paket layanan sepanjang siklus hidup nya. Dalam hal ini, akan dibahas bagaimana implementasi perancangan sistem pembayaran online dengan menggunakan metodologi *agile* dan teknologi *SOA***

submision codes:

http://Gunadarma.ac.id

1. PENDAHULUAN

A. Latar Belakang Masalah

Mobile Money System adalah salah satu alat pembayaran dengan menggunakan telepon genggam *(handphone)*, atau disebut juga *mobile money*. *Mobile Money System* ini memungkinkan seseorang melakukan pembayaran suatu layanan dengan menggunakan *handphone*, sehingga lebih mudah dan bisa dimana saja. untuk mengatasi kebutuhan pasar terhadap layanan pembayaran yang sangat cepat, maka penyediaan layanan tersebut harus di dukung dengan kemampuan dari sistem yang cepat ber-adaptasi dan cepat menyediakan kebutuhan pasar.

Menyadari akan hal tersebut, maka diperlukan suatu metodologi yang bisa cepat ber-adaptasi terhadap kebutuhan pasar, dan bisa mendefinisikan pengembangan suatu layanan dengan cepat. Terkait dengan kebutuhan tersebut, maka pilihan nya adalah menggunakan ***Agile Development Methodology***, dibandingkan dengan menggunakan metodologi tradisional, seperti misalnya adalah ***waterfall***.

Selain ***Development Methodology***, diperlukan juga suatu teknologi yang mendukung hal tersebut. Dimana teknologi tersebut membuat sistem bisa dikembangkan secara modular / *service*. Sehingga modul - modul tersebut bisa di kembangkan secara terpisah dan bisa di *re-use*. Karena itu maka di putuskan teknologi yang tepat adalah menggunakan ***Service Oriented Architecture (SOA)***.

B. Rumusan Masalah

Rumusan masalah pada jurnal **"Pengembangan Aplikasi Pembayaran Online *Mobile Money System* Dengan Menggunakan *Agile* Metodologi"** adalah untuk menjelasakan bagaimana pengembangan sistem tersebut menggunakan ***Agile Development Methodology*** dan dengan menggunakan ***Service Oriented Architecture*** teknologi.

C. Tujuan Penulisan

Tujuan penulisan ini adalah untuk menjelaskan peran ***Agile Methodology*** dan tenologi ***Service Oriented Architecture*** dalam menjawab kebutuhan permintaan pengembangan sistem yang cepat beradaptasi.

2. LANDASAN TEORI

A. Agile Development Method

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. 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.

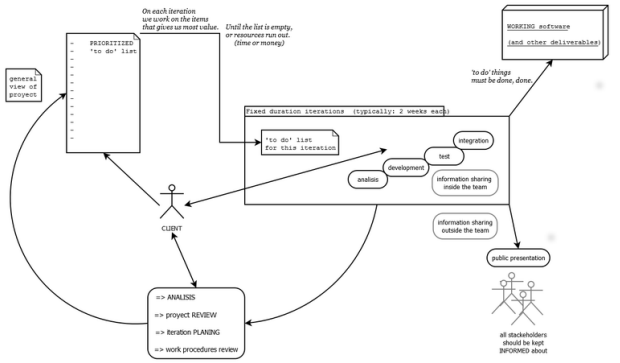

Fig. 1. Agile Methodology

Agile development methods terdefinisi dalam empat nilai, biasa di sebut ***Agile Alliances Manifesto***. Berikut ini isi dari *Agile Alliances Manifesto* :

Manifesto for Agile Software Development

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more.

Dari *Agile Manifesto* terdefinisi dalam empat nilai :

- **Interaksi dan personel** lebih penting daripada proses dan alat.
- **Perangkat lunak yang berfungsi** lebih penting daripada dokumentasi yang lengkap.
- **Kolaborasi dengan klien** lebih penting daripada negosiasi kontrak.
- **Respon terhadap perubahan** lebih penting daripada mengikuti rencana.

B. Service Oriented Architecture

SOA (service oriented architecture, arsitektur berorientasi layanan) adalah suatu gaya arsitektur sistem yang membuat dan menggunakan proses bisnis dalam bentuk paket layanan sepanjang siklus hidupnya. SOA membagi fungsi - fungsi menjadi unit - unit yang berbeda (layanan), yang dapat di distribusikan melalui suatu jaringan dan dikombinasikan serta digunakan ulang untuk membentuk aplikasi bisnis.

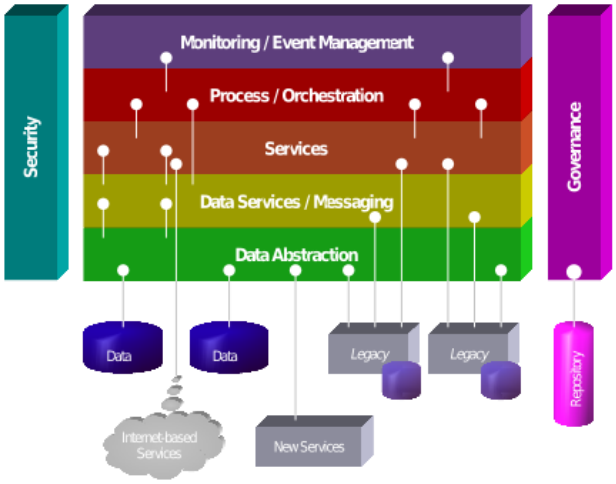

Fig. 2. SOA Model

Dalam meng-implementasikan *service-oriented architecture*, bisa menggunakan web service *(Web Service Approach)*. Yaitu dengan membuat fungsi - fungsi yang independent dan bisa di akses melalui jaringan, dan tidak tergantung kepada *platform* dan bahasa pemrograman. Service ini bisa me-representasikan suatu aplikasi atau sebagai *wrapper* suatu *legacy system* agar bisa di akses melalui jaringan.

Setiap aplikasi SOA bisa mempunyai role seperti berikut ini :

1. *Service provider*. Merupakan aplikasi yang menyediakan web service. Setiap service provider harus menyediakan service - service yang bisa di gunakan dalam bentuk *interface*.
2. *Service consumer*. Yaitu aplikasi yang akan menggunakan service tersebut (*consume*).

Dalam implementasi nya, dalam membangun SOA menggunakan standar komunikasi *web service* seperti misalnya menggunakan SOAP (*Simple Object Access Protocol*). Tetapi bisa juga menggunakan metode komunikasi lain yang di sepakati.

3. PEMBAHASAN

A. Service Pembayaran Online

Dalam Bab ini, penulis akan menggambarkan service yang dibuat. Dengan asumsi beberapa service sudah tersedia sebelum nya, maka disini hanya akan di batasi kepada service utama nya. Juga pembahasan akan dibatasi hanya pada spesifikasi service tersebut, tapi tidak ke implementasi nya. Berikut ini design service yang terlibat / yang akan dibuat dalam penulisan berikut ini :

service yang akan dibuat adalah :

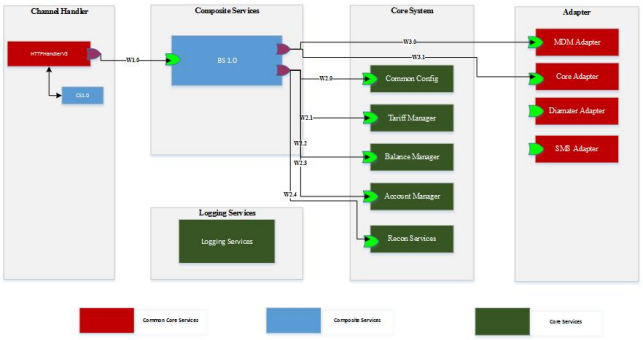

Fig. 3. Service Model

Table 1. Daftar Service

No	Service Code	Service Category	Service Name
1	BS1.0	Business services	svgBilXLTunaiPartnerServicesV1.0

untuk spesifikasi interface service adalah seperti berikut ini :

Table 2. Daftar Interface

Wire Code	Call Method	Service Consumer	Service Provider	Operation Name
W2.0	SOAP	BS1.0	Common Config	OpGetConfig
W2.1	SOAP	BS1.0	Tariff Manager	opGetTariffAmount
W2.2	JMS	BS1.0	Balance Manager	opAdjustBalance
W2.2	JMS	BS1.0	Account Manager	opGetCustomerData
W2.2	JMS	BS1.0	Account Manager	opGetMerchantData

B. Business Process Model

Berikut ini Business Process Model untuk menggambarkan design dari aplikasi.

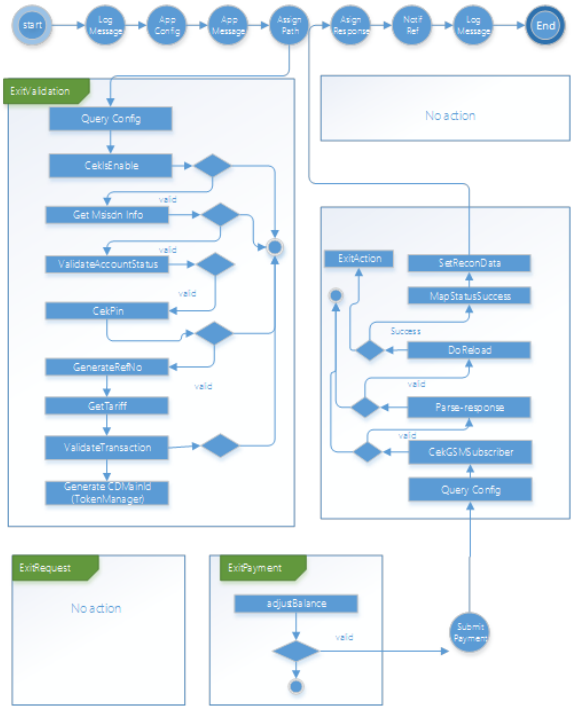

Fig. 4. Business Flow

C. Skema XML

Berikut ini skema XML yang digunakan dalam service ini :

Table 3. Skema *Payment*

Payload Name	Sub element Name	Data Type	Mandatory
PaymentRq	PaymentRq	Complex Element	Optional
PaymentRq	MSISDN	String	N
	service-id	String	N
	order-id	String	N
	Reference-no	String	N
	amount	String	N
	RequestPayload	Complex Element, Any	N

D. Agile

Pada tahap ini, penulis mencoba untuk membuat implementasi dari *agile methodology*. Dalam melaksanan proses *agile* ini, ada beberapa *term* yang harus di mengerti :

- *Iteration*. Adalah periode dimana *agile* tim harus menyelasikan tugas yang sudah di sepakati di awal. Dalam hal ini adalah mengembangkan aplikasi yang sudah di definisikan bersama. Setelah proses ini selesai, maka tim harus siap untuk me-respon perubahan dan adaptasi terhadap pengembangan aplikasi.
- *Task*. Adalah list pekerjaan yang sudah di detailkan dalam mengembangkan aplikasi. Biasanya *task* ini lebih kepada pembagian dari fungsi - fungsi dari aplikasi.
- *Backlog*. Adalah sekumpulan dari *task* yang harus di kerjakan berikut nya oleh tim. *Task* yang harus dikerjakan ini, bisa jadi adalah task yang belum selesai sebelum nya.

Dalam pengembangan aplikasi ini, dilakukan *project tracking* dengan menggunakan agile methodology. Pertama kali dalam agile methodology adalah dengan membuat *story* dari aplikasi tersebut.

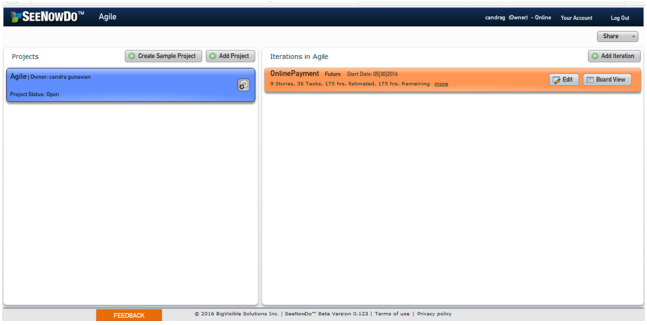

Fig. 5. Agile Story

Setelah dibuat *story* utama, maka bisa dibuat *sub story* dalam pengembangan aplikasi tersebut. *Sub-story* tersebut dalam agile, dibuat menjadi task - task. Lalu di tracking tahapan nya. Dalam setiap *task*, bisa secara *independent* di lakukan, bisa *on-progress*, *backlog*, atau memang sudah ada yang *done*.

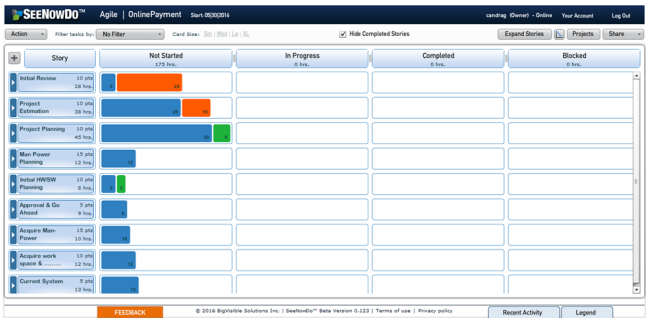

Fig. 6. Agile Sub-Story

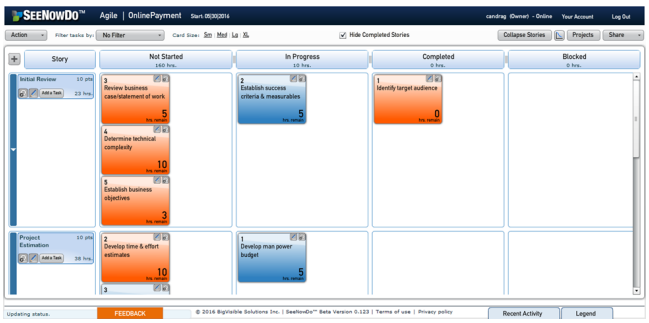

Fig. 7. Agile Progress

Dengan melakukan pengembangan aplikasi dengan metode *Agile*, setiap story dapat berjalan *independent*, tetapi dengan tujuan yang sama. Dan juga, untuk suatu *task*, bisa di *tracking* secara independent. Proses penyelesaian aplikasi pun bisa lebih cepat, karena setiap task, harus bisa di *deliver*.

4. KESIMPULAN

A. Rangkuman

Agile Methodology memungkinkan proses pengembangan sistem dan aplikasi menjadi lebih cepat. Karena aplikasi akan di buat menjadi bentuk *story - story* dan *task* yang detail. Untuk proses *delivery* setiap *task* detail. Karena agar proses setiap *task* bisa segera di evaluasi. Dan prinsip dari *Agile* adalah merespon perubahan, maka jika ada perubahan, bisa dikerjakan di *task* atau *story* berikut nya.

Service Oriented Architecture (SOA) di *implementasikan* disini untuk mendukung metodologi *agile* ini. Sehingga apliksai bisa dibuat berdasarkan layanan, dan *lousy coupled*. sehingga di harapkan tidak ada saling ketergantungan, dan setiap perubahan / adaptasi dilakukan tidak saling impact.

B. Saran

Adapun dari penulisan berikut ini, ada beberapa saran yang penulis rangkum di dalam penulisan ini :

- Penulis baru menggambarkan perancangan sistem dengan menggunakan agile pada proses bisnis utama, dengan asumsi komponent lain sudah tersedia
- Penulis menulisakan spesifikasi SOA dengan anotasi *SCA (Service Composite Architecture)* dan baru di leve interface nya saja. item Penulis belum melakukan menampilkan data komparasi antara metode biasa (contoh nya SDLC) dan dengan agile. Manakah yang lebih efisien.