

Mengintip dasar pengembangan sistem informasi dengan metode Agile. *Why Agile Rocks?*

Hari Mantik
hari.mantik@yahoo.com

Abstrak

Tulisan ini memberikan ulasan dan wacana singkat mengenai Agile Methodology, termasuk definisi, keunggulan, dan perbedaannya dengan Waterfall Methodology. Menjamurnya perusahaan startup bukan hanya membuat agile semakin dinikmati dan semakin dipilih, tapi juga memberi pertanyaan, mengapa harus Agile? Agile merupakan jawaban akan respons cepat terhadap perubahan, dan mementingkan kolaborasi diatas yang lainnya. Beberapa kelemahan yang didapat ketika sistem dijalankan dengan metode waterfall, seperti kurang fleksibel, dan unpredictable, secara baik dapat ditutupi oleh Agile. Agile menggunakan framework yang disebut dengan "scrum", yaitu sebuah kerangka kerja berbasis manajemen proyek yang menekankan pada kerja tim, akuntabilitas, dan iteratif. Scrum dimulai dengan kick-off meeting, daily stand-up, project backlog, user story, dan sprint. Agile juga menggunakan metode "kanban" yang semakin membuat kerja user semakin cepat, terdedikasi, dan terarah.

Pendahuluan

Agile Goals

Tujuan dari agile, atau sering di sebut sebagai "*agile manifesto*", menyatakan beberapa nilai, diantaranya:

1. Menciptakan interaksi manusia melalui proses dan tools
2. Bekerjanya sistem/software melalui dokumen yang komprehensif
3. Kolaborasi pelanggan
4. Respons terhadap perubahan

Dalam Agile, kita sadar bahwa suatu project melekat suatu ketidakpastian. Keadaan ini memaksa suatu project harus berubah setiap saat. Baik itu perubahan pasar, perubahan regulasi atau perubahan fitur ketika produk mulai "*go-live*". Agile berhasil merangkul ketidakpastian ini dengan baik. Dengan memecah project kedalam potongan kecil, Agile membuatnya menjadi lebih mudah dalam melakukan prioritas dan menambah atau mengurangi fitur walaupun *project* sudah mencapai separuh jalan. Ini adalah

sesuatu yang sangat tidak mungkin kita lihat di project yang menggunakan metodologi tradisional *waterfall*.

12 Prinsip Agile

1. Prioritas utama Agile adalah memuaskan pelanggan melalui hasil yang lebih awal dan berkelanjutan dalam menciptakan perangkat lunak yang berkualitas
2. Menghasilkan (*deliver*) sistem secara berkala, baik dalam hitungan minggu atau bulan, dengan preferensi ke skala waktu yang lebih pendek
3. Terbuka terhadap perubahan *requirement*, bahkan ketika para programmer agile dalam fase kerja yang terlambat. Proses ini dimanfaatkan agile team sebagai keunggulan kompetitif yang mereka tawarkan kepada pelanggan.
4. Tim bisnis dan tim developer harus bekerjasama setiap hari selama project berlangsung demi menghasilkan

sistem dengan kualitas mutu yang baik.

5. Agile membangun project ditengah-tengah kelompok kerja yang sudah termotivasi. *Project Manager* Agile memberikan ruang dan lingkungan kerja yang kondusif, dengan kata kunci kepercayaan terhadap team dalam bekerja dan berimprovisasi.
6. Agile menggunakan metode yang efisien dan efektif penyampaian informasi kepada team development melalui percakapan tatap muka (*face-to-face conversation*).
7. Alat ukur utama dalam melihat kemajuan yang terjadi adalah dengan melihat atau membandingkan dengan sistem yang berjalan saat ini.
8. Proses dalam Agile menciptakan pengembangan yang berkelanjutan. Para sponsor, pengembang, dan pengguna harus saling bekerjasama mempertahankan program yang cepat dan berkelanjutan tanpa batas ini secara konstan.
9. Perhatian secara terus menerus pada keunggulan teknis dan desain yang mumpuni adalah bagian dari metodologi Agile.
10. Kesederhanaan seni memaksimalkan penggunaan sumber daya yang efisien (mengurangi jumlah pekerja yang *idle*) adalah penting.
11. Arsitektur terbaik, *requirement* yang optimal, serta desain timbul dari organisasi yang disebut dengan "*self-organizing team*".
12. Team Agile akan selalu melakukan pekerjaan secara efektif, selalu beradaptasi, menyesuaikan, dan "*fine-tuning*" menyesuaikan dengan perilaku mereka.

Mengapa Agile diminati? – Why Agile Rocks?

Kalimat "*why agile rocks*" menjadi fenomena tersendiri. Perusahaan startup yang biasanya berisikan anak muda sangat kondusif terhadap metodologi Agile. Mereka berusia dibawah 30, ambisius, semangat, bekerja dengan kebebasan yang bertanggungjawab, dan situasi kerja yang sangat nyaman. Berikut adalah beberapa karakteristik, mengapa "*Agile Rocks*"

1. ***Speed to Market***, Agile memungkinkan *Project Manager* mendapatkan konsep dari user secara cepat. Dalam setiap "*sprint*" project Agile akan memberikan satu hasil yang bernilai. Di titik manapun, kita dapat menentukan apakah kita akan meluncurkan/*deploy* apa yang sudah diselesaikan, mengujinya, atau memulai siklus baru pengembangan sistem.
2. ***Flexibility***, Agile tercipta untuk mengakomodir perubahan. Proyek sistem informasi secara konstan mengalami perubahan. Seiring dengan sebuah produk menjadi "*life*" atau pasar mulai mengalami pergeseran dan perubahan. Dari sini Agile sadar bahwa ide-ide hebat akan datang tiba-tiba disaat project sedang berjalan, dan kita harus memanfaatkannya, bukan mengunci ide tersebut.
3. ***Risk Management***, Agile menggunakan "*incremental releases*" atau sistem di keluarkan secara bertahap. Artinya produk dapat digunakan di awal-awal proses oleh user atau para stakeholders. Hal ini memungkinkan pengguna dalam mengidentifikasi isu-isu yang timbul lebih awal dan mengenal fitur atau fungsi yang mana yang perlu di tambah. Dengan beradaptasi untuk berubah, bukanlah sesuatu yang tidak mungkin jika ada perubahan *scope* dan fitur yang

- terjadi ditengah-tengah *project*. Sesuatu yang tidak mungkin dilakukan dengan metode *waterfall*.
4. **Cost Control**, Tidak seperti proyek dengan anggaran tetap (*fixed budget*), Agile sangat fleksibel dalam menata anggaran sesuai dengan scope yang diminati. Sering terjadi *client* dan user menyadari jika fitur yang mereka minta di awal ternyata saat ini tidak diperlukan. Hal ini membuat sistem di luncurkan lebih cepat dan *client* membayar dan membayar lebih sedikit. Agile bukanlah sesuatu yang harus dibayar mahal akan tetapi penuh dengan ketidakpastian. *Client* hanya membayar untuk sesuatu yang mereka perlukan. Agile mampu melakukan pengaturan ulang backlog produk pada sistem sesuai fitur yang akan di implementasikan, mengorbankan fitur yang tidak penting, sesuai dengan anggaran dari *client*.
 5. **Quality**, Agile mengintegrasikan pengujian di seluruh proses. Secara konsisten memberikan sistem yang sudah melalui tahap *Quality Assurance* berarti adalah jaminan suatu produk dengan kualitas yang baik
 6. **Right Product**, Perilisan produk secara inkremental memungkinkan team Agile melakukan pengujian lebih dini dan lebih sering. Bahkan, jika tidak di rilis ke publik pun, akan jauh lebih mudah untuk menemukan berbagai kekurangan dan hal-hal yang dapat di tingkatkan ketika kita mempunyai produk yang jadi yang siap menjadi ajang "*play around*" para *developer* dan *user*.
 7. **Transparency**, Agile memungkinkan kita melihat, merasakan, dan menggunakan proyek secara konsisten di seluruh proyek tanpa adanya pengkotak-kotakan (contoh transparansi menggunakan *kanban window* dari *Jira software*)
 8. **Responding to Change**, Selama proses kolaboratif dengan pelanggan, kebutuhan dan prioritas bisnis dapat bergeser dengan cepat. Agile lebih responsif dalam menerima perubahan dibandingkan *waterfall*, dikarenakan pengembangan software sejak awal terbagi dalam "*scrum*".

Kesalahpahaman dalam memahami AGILE

1. **It's defferent** – Perlu diakui, *waterfall* model bukanlah proses yang sempurna. Walaupun dengan metode *waterfall* semuanya terasa lebih terkendali karena tahapan dipetakan dari awal, akan tetapi proyek memakan waktu lebih lama dan biaya yang lebih besar. Model *waterfall* juga tidak secara fleksibel memberi ruang perubahan sebelum suatu project dianggap selesai dan *go-live*.
2. **Fixed budget** – Agile memberikan sumber daya yang terdedikasi. Artinya, *team* Agile dapat memperkirakan berapa lama harus menyelesaikan tujuan yang sudah di gariskan sesuai dengan dana yang dimiliki. Seiring dengan project yang bisa setiap saat berkembang, penambahan fitur baru, dengan metode Agile, kita dapat membuang fitur yang mungkin sudah usang atau tidak cocok, dan menggantinya dengan yang baru, yang lebih murah, dan sesuai dengan dana yang dialokasikan.
3. **It's unpredictable** – Agile tidak bisa di prediksi. Semua project pun sebenarnya tidak bisa diprediksi. Memang sangat mustahil untuk mengetahui apa yang diinginkan oleh *end-user*, akan tetapi Agile merangkul ketidakpastian ini dan meman-

faatkannya untuk menghasilkan yang lebih baik.

4. ***Developers make all the features*** – Kesalahpahaman yang sering timbul adalah bahwa para pengembang bisa memilih apa yang penting dan apa yang akan diimplementasikan. Hal ini adalah salah. Sebelum “*sprint*” dimulai, selalu ada rapat komprehensif yang kuncinya adalah: semua pemangku kepentingan harus hadir. Para pemangku itu mencakup pengembang, desainer, team business, serta siapa saja yang terlibat dalam produk. Jadi pengembang bukanlah penentu utama dari metodologi Agile ini.
5. ***It doesn't consider long-term*** – Banyak orang beranggapan karena Agile menerapkan *release* yang bersifat pendek dan berulang-ulang (*short-iterative releases*), Agile tidak memperhitungkan kebutuhan dan tujuan jangka panjang. Padahal, rilis yang bersifat pendek dan berulang-ulang adalah cara berbeda dari Agile untuk sampai ke ujung/tujuan. Dengan melakukan sesuatu yang bisa kita uji sejak awal, dengan cara membagi/memecah melalui *short-iterative releases*, akan lebih memungkinkan kita membuat keputusan dan hasil jangka panjang yang lebih baik.

Agile Lifecycle

Dalam Agile, siklus hidup yang singkat dalam pengembangan aplikasi berfokus pada “*scrum*”. Proyek *scrum* ini dipecah menjadi iterasi pendek yang disebut dengan “*sprint*”. Siklus setiap *sprint* meliputi seperti gambar di bawah ini:



Gambar 1: Siklus Sprint (Emerson Taymor, agilehandbook.com)

Kickoff / Sprint Planning

Setiap *scrum-project* dimulai dengan *kick-off meeting* (pertemuan inisialisasi pertama). Pertemuan pertama umumnya membahas kebutuhan *project backlog* yang harus dibuat, dan memperkenalkan team project. Tujuan awal dari *Sprint Planning* adalah:

1. Membuat *overview* / rangkuman dari project dan tujuan yang akan dicapai
2. Menyeleksi dan menetapkan siapa saja yang akan bekerja dalam project
3. Menetapkan PIC dari *client* yang akan melakukan *sign-off* (penandatanganan berita acara hasil *project*)
4. Membuat *project-backlog*
5. Menetapkan fitur atau fungsi mana yang akan dikerjakan dalam suatu sistem
6. “*Getting on the same page*” atau seluruh *team* dan *client* harus punya paradigma, pandangan dan arah yang sama.

Project Backlog

Project Backlog merupakan serangkaian daftar fitur produk yang secara umum di definisikan sebagai “*user stories*”. *User story* dalam Agile diartikan sebagai bahasa yang natural, informal yang

menjelaskan deskripsi dari satu fitur atau lebih dalam pengembangan sistem. Struktur penulisan user story adalah sebagai berikut:

<p style="text-align: center;">ASA [type of user],</p> <p style="text-align: center;">I WANT TO [do this thing],</p> <p style="text-align: center;">SO THAT I CAN [accomplish this goal].</p>
--

Gambar 2: Penulisan user story (Emerson Taymor, agilehandbook.com)

Penulisan *User Story* memang tidak perlu seperti format diatas. *User story* juga tidak mewakili *requirement* dari sistem, tapi dapat memaksa si penulis memberikan “artikulasi” dari 3 pertanyaan berikut:

1. Untuk siapa sistem kita bangun, siapa user kita? – *As a <type of user>*
2. Apa yang akan kita bangun, apa tujuan sistem ini? – *I want <some goal or objective>*
3. Kenapa kita membangunnya, value apa yang kita berikan ke user? – *So that <benefit, value>*

Contoh penulisan *User Story*:

“As a visitor I want to search by keyword, so that I can find and read relevant article”

“Sebagai nasabah Bank, saya ingin mengganti PIN secara online, sehingga saya tidak perlu datang ke cabang”

The Sprint

Project berbasis Agile dipecah-pecah menjadi kecil, dengan interval waktu yang konsisten. Interval ini disebut dengan “*sprint*”. *Sprint* bisa singkat dalam hitungan beberapa hari, dan umumnya tidak lebih dari 3 sampai 4 minggu. Sebelum setiap *Sprint*, akan ada “*sprint planning meeting*”, yaitu pertemuan untuk menentukan tujuan dan

mekanisme dari *sprint*. Selama *sprint*, tidak boleh ada fitur yang ditambahkan, dan sasaran dari *sprint* tidak boleh di rubah. Satu-satunya pengecualian adalah jika team menyelesaikan 1 *sprint* lebih awal. Komunikasi dengan klien atau team pengembang umumnya terbatas pada hasil *stand-up* harian, tetapi beberapa perusahaan memungkinkan dialog terbuka melalui fasilitas *chatroom*.

Daily Standup

Salah satu kebiasaan dan menjadi keunikan dari Agile adalah adanya “*daily standup*” yang dilakukan setiap hari selama project berjalan. *Daily Standup* wajib dihadiri oleh seluruh team yang terkait. Mulai dari *Project Manager, Product Owner, Developer, Tester, dan User*. Pertemuan dilaksanakan secara singkat (berkisar maksimum 15 menit), dan kadang disiapkan alarm atau pengatur waktu agar pertemuan tidak terlalu lama dan menghilangkan pembicaraan yang bertele-tele. Sesuai dengan namanya, *daily standup*, semua orang yang terlibat harus berdiri agar lebih terfokus dan penuh konsentrasi. Jika *Project* terbagi dalam beberapa *sprint* dengan jadwal yang sama, berarti juga *daily standup* dilakukan secara bersamaan dari tiap *sprint*. Dalam *daily standup*, tiap orang dalam team harus menjawab 3 pertanyaan sederhana:

1. *What did you do yesterday?* – Apa yang telah anda capai kemarin?
2. *What are you going to do today?* – Apa yang ingin anda capai hari ini?
3. *Do you need any help or there are any blockers in the way?* – Apakah anda perlu bantuan, atau ada sesuatu pekerjaan yang tidak bisa dilakukan?

Ketiga pertanyaan tersebut merupakan jembatan menuju transparansi. Semua orang di tim berada dalam keterkaitan (*in*

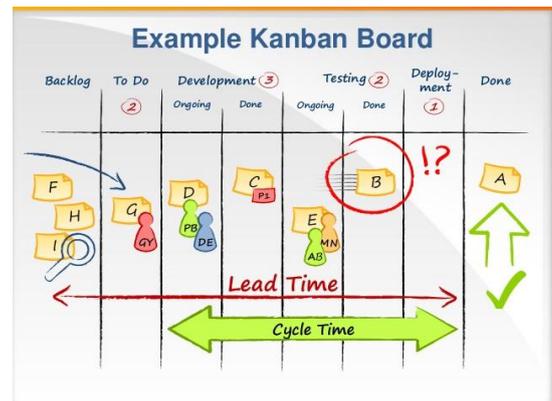
the loop), jawaban yang diberikan akan membuat orang tersebut bertanggung-jawab terhadap apa yang mereka katakan dan apa yang akan mereka hasilkan (*deliverable*). Hasil dari *daily standup* ini biasanya di bagikan ke client. Komunikasi harian ini memastikan bahwa jika ada yang menahan *team*, mereka bisa mendapat respon yang cepat.

Jalan Menuju Sukses

1. **Communication.** Setiap project mendapatkan manfaat tersendiri dari komunikasi yang baik. Tidak terkecuali project Agile. Jika anda tidak pernah menjalankan project berdasarkan metodologi Agile, komunikasi adalah bagian yang penting dalam Agile. Berada dalam lingkaran *project* dengan menatap jadwal (yang terlambat, *on progress*, ataupun apa yang di depan) pasti akan menimbulkan kekhawatiran dan ketidakpastian dalam benak seluruh pihak. Proses yang transparan dan komunikasi yang baik akan membuat team lebih merasa nyaman dan memungkinkan mereka fokus pada yang lebih penting: “memberikan produk terbaik kepada user”
2. **Dedicated Team.** Agile bekerja paling baik dalam team yang berdedikasi, berisikan orang-orang yang mau dan ingin berkolaborasi. Semakin baik kolaborasi, semakin baik produk.
3. **Good Planning.** Agar *project* Agile berhasil, sangat dibutuhkan perencanaan yang baik. Hal ini bukan berarti merencanakan segalanya hingga sedetail mungkin seperti dalam metode waterfall, tetapi dengan berpikir kedepan: “muncul dengan *project backlog* yang kuat dan mengestimasi fitur sebaik mungkin”

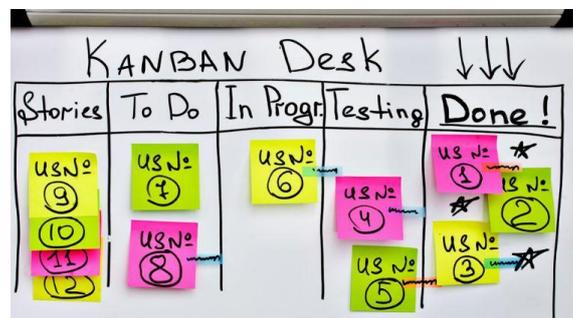
Metode Kanban

Metode kanban merupakan “*lean methodology*”, atau metode dengan pendekatan sistematis yang menggambarkan nilai bisnis melalui peningkatan secara terus menerus. Team Agile menggunakan “papan kanban” papan tulis, papan elektronik, bahkan jendela yang dicoret-coret atau ditempel dengan carikan kertas ataupun post-it-notes, untuk menampilkan yang kita sebut sebagai kanban (tahap dan status pekerjaan) yang disusun dalam kolom yang mewakili tahapan dalam proses, buffer, kondisi dan antrian. Papan kanban bisa di lihat seperti gambar berikut:



Gambar 2: Kanban Board

https://blog.pinkelephant.com/blog/pr_7_1_-_using_kanban_for_itsm_operations



Gambar 3: Kanban Desk

<https://bbvaopen4u.com/en/actualidad/kanban-method-why-does-it-work-software-programming>

Agile dan Start-Up Company

Beberapa tahun belakang ini terdapat istilah yang marak dibidang kewirausahaan teknologi informasi yaitu *Start-Up*. Definisi dari *Start-Up* adalah *fresh, new, innovative companies trying to do cool stuff with technology*. *Startup* merupakan perusahaan baru yang bergerak di bidang teknologi informasi yang mengusung ide baru sebagai model bisnis mereka, startup merupakan bentuk kewirausahaan yang memfokuskan untuk bergerak pada bidang teknologi informasi. Banyak perusahaan besar yang memulai bisnisnya dari perusahaan *startup*. Beberapa perusahaan startup bahkan saat ini sukses masuk ke lantai bursa di amerika, misalnya Evernote, Yelp, Boxvideo, dan Brightroll. Di jakarta kita mengenal Tokopedia yang sudah memasuki usia 10 tahun dan terus berkembang. Di Bali juga ada Bulseye Digital sebuah perusahaan layanan web yang mempunyai *client* besar seperti Coca Cola Company dan Ogilvy Australia.

Kesimpulan

Pilihan metodologi dalam perancangan sistem informasi kini tidak lagi terpaku pada konsep air terjun. Agile membawa perubahan yang cukup besar dalam perjalanan perancangan sistem informasi: lebih cepat, lebih dekat dengan user, lebih kolaboratif, dan lebih terbuka untuk perubahan. Akan tetapi konsep air terjun tidak serta-merta kita tinggalkan. Bagi perusahaan yang bergerak dalam institusi keuangan, perbankan, dan pemerintahan, konsep air terjun masih sangat signifikan dan penting untuk diterapkan. Regulasi pemerintah, security pada kerahasiaan pelanggan, sertifikasi keamanan, seperti PCI DSS (*payment card industri data security standard*), dan *2nd Authentication token*, masih membuat konsep *waterfall*

yang masih menerapkan cara tradisional dalam mendalami keinginan pelanggan (*business requirement documents*) hingga *security awareness* tetap menjadi favorit dalam dunia sistem informasi. Baik Agile ataupun konsep air terjun, keduanya adalah konsep terbaik saat ini, tergantung siapa anda, apa business anda, dan siapa *client* anda.

Daftar pustaka dan bacaan

- Pengembangan Sistem Informasi menggunakan Metodologi Agile, Raharjana, Indra Kharisma, Penerbit Deepublish, 2017
- Agile for dummies, Ambler, Scott W., Holitza, Matthew, Wiley, 2012
- Agile handbook, Taymor, Emerson, agilehandbook.com/agile-handbook.pdf, 2018
- User Story, unknown, https://en.wikipedia.org/wiki/User_story, 2018
- User Story, unknown, <http://www.yodiz.com/blog/writing-user-stories-examples-and-templates-in-agile-methodologies/>, 2018
- Metode Kanban, <https://bbvaopen4u.com/en/actualidad/kanban-method-why-does-it-work-software-programming>, 2018
- Metode Kanban, <https://blog.pinkelephant.com/blog/pr-71-using-kanban-for-itsm-operations>, 2018