**TUGAS**

**METODOLOGI PENGEMBANGAN**

**SISTEM INFORMASI**



Disusun oleh :

**FAZAL SAID WICAKSONO**

**145610146**

**JURUSAN SISTEM INFORMASI**

**SEKOLAH TINGGI MANAJEMEN DAN INFORMATIKA**

**AKAKOM YOGYAKARTA**

**2017**

* **Agile Software Development**

***Agile Software Development***  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](https://id.wikipedia.org/wiki/Metodologi_pengembangan_perangkat_lunak) yang digunakan dalam pengembangan perangkat lunak. Agile memiliki pengertian bersifat cepat, ringan, bebas bergerak, dan waspada.[[1]](https://id.wikipedia.org/wiki/Agile_Development_Methods#cite_note-1)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.

Saat bekerja dalam tim untuk mengerjakan suatu proyek sangatlah penting menentukan [Metodologi pengembangan perangkat lunak](https://id.wikipedia.org/wiki/Metodologi_pengembangan_perangkat_lunak) dan [Proses pengembangan perangkat lunak](https://id.wikipedia.org/wiki/Proses_pengembangan_perangkat_lunak)yang akan digunakan. [Metodologi pengembangan perangkat lunak](https://id.wikipedia.org/wiki/Metodologi_pengembangan_perangkat_lunak) sendiri adalah sebuah metodologi yang digunakan untuk membuat struktur, rencana, dan kontrol pengerjaan suatu proyek, sedangkan [Proses pengembangan perangkat lunak](https://id.wikipedia.org/wiki/Proses_pengembangan_perangkat_lunak) adalah model-model dan metodologi yang digunakan untuk mengembangkan suatu perangkat lunak. Ada beberapa model [Metodologi pengembangan perangkat lunak](https://id.wikipedia.org/wiki/Metodologi_pengembangan_perangkat_lunak) diantaranya : waterfall, fountain, spiral, rapid, prototyping, incremental, build & fix, dan synchronize & stabilizeTerdapat enam langkah yang digunakan dalam [Metodologi pengembangan perangkat lunak](https://id.wikipedia.org/wiki/Metodologi_pengembangan_perangkat_lunak) yaitu :

* 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 maintenanance kedepannya.

*Deployment*, yaitu proses yang dilakukan oleh penjamin kualitas untuk menguji kualitas sistem. Setelah sistem memenuhi syarat maka perangkat lunak siap di*deployment*.

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.

Agile manifesto

*Agile development methods* terdefinisi dalam empat nilai, biasa di sebut Agile Alliance’s Manifesto,[[4]](https://id.wikipedia.org/wiki/Agile_Development_Methods#cite_note-manifesto-4) diantaranya :

* **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.

Pengertian dari Agile Alliance's Manifesto[[4]](https://id.wikipedia.org/wiki/Agile_Development_Methods#cite_note-manifesto-4) dijelaskan di bawah ini:

* **Perangkat lunak yang berfungsi** lebih penting daripada dokumentasi yang lengkap, saat melakukan proses demonstrasi kepada klien, perangkat lunak yang berfungsi dengan baik akan lebih berguna daripada dokumentasi yang lengkap.
* **Kolaborasi dengan klien** lebih penting daripada negosiasi kontrak, salah satu ciri dari agile adalah klien menjadi bagian dari tim pengembangan perangkat lunak. Kolaborasi yang baik dengan klien saat proses pembuatan perangkat lunak sangatlah penting ketika menggunakan agile. Karena fungsi-fungsi dari perangkat lunak yang dikembangkan harus terus menerus dibicarakan dan diimprovisasi disesuaikan dengan keinginan klien.

Model proses agile

Beberapa model dari *agile development methods*,[[2]](https://id.wikipedia.org/wiki/Agile_Development_Methods#cite_note-agile-2) yaitu :

* Acceptance Test Driven Development (ATDD)
* Agile Modeling
* Adaptive Software Development (ASD)

Adaptive software development (ASD) diajukan oleh Jim Highsmith sebagai teknik untuk membangun software dan sistem yang kompleks. Filosofi yang mendasari *adaptive software development* adalah kolaborasi manusia dan tim yang mengatur diri sendiri. Sistem kerja *adaptive software development* : collaboration dan learning. '

Collaboration : orang-orang yang bermotivasi tinggi bekerja sama, saling melengkapi, rela membantu, kerja keras, terampil di bidangnya, dan komunikasikan masalah untuk menyelesikan masalah secara efektif.

Learning: tim developer sering merasa sudah tahu semua hal tentang proyek, padahal tidak selamanya begitu. Karena itu proses ini membuat mereka belajar lebih tentang proyek melalui tiga cara:

* Fokus grup, klien dan pengguna memberi masukan terhadap perangkat lunak.
* Formal Technique Reviews, tim ASD lengkap melakukan review.
* Postmortems, tim ASD melakukan instrospeksi pada kinerja dan proses.
* Tujuan agile

Secara garis besar tujuan dirumuskannya *agile development methods*,[[7]](https://id.wikipedia.org/wiki/Agile_Development_Methods#cite_note-7) yaitu :

* ***High-value & working App system***, diharapkan dengan memakai *agile development methods* dapat dihasilkan perangkat lunak yang mempunyai nilai jual yang tinggi, biaya pembuatan bisa di tekan dan perangkat lunak bisa berjalan dengan baik.
* ***Cost control & value-driven development***, salah satu tujuan dari *agile* yaitu pengembangan perangkat lunak disesuaikan dengan kebutuhan pengguna, tim bisa dengan cepat merespon kebutuhan yang diinginkan pengguna sehingga waktu dan biaya pembuatan perangkat lunak bisa dikontrol.
* ***High-quality production***, walaupun biaya pembuatan perangkat lunak bisa ditekan dan proses pembuatan bisa dipercepat , tetapi kualitas dari perangkat lunak yang dibuat harus tetap dijaga. Dengan melakukan tes setiap fungsionalitas perangkat lunak setelah selesei dibuat berarti *agile* juga mengakomodir kebutuhan ini.
* ***Flexible & risk management***, jika kita menggunakan metode pembuatan yang biasanya dipakai, jika ingin mengubah fungsionalitas dari *wireframe* yang telah dibuat di butuhkan proses yang rumit. Mulai dari pertemuan dengan sistem analis untuk mengubah sistem perangkat lunak, perubahan rencana rilis produk hingga perubahan biaya produksi. Pertemuan dengan klien untuk melakukan tes perangkat lunak juga sering dilakukan sehingga fungsionalitas perangkat lunak mudah diubah dan akhirnya kegagalan perangkat lunakpun bisa diminimalisir.
* ***Self-organizing, self-managing teams***, rekrut orang terbaik, beri dan dukung kebutuhan mereka lalu biarkan mereka bekerja. Itulah perbedaan *agile* dan SDM lainnya. Dengan *agile*, *developer* dapat memanajemen dirinya sendiri, sedangkan manajer tim hanya bertugas mengkolaborasikan *developer* perangkat lunak dengan klien. Sehingga terciptalah tim yang solid.
* **Bagaimana agile bekerja**

Topik selanjutnya yaitu tentang cara kerja *agile development.* Disini akan dijelaskan bagaimana *agile development methods* (model scrum) digunakan dalam manajemen proyek.

**Komposisi tim**

Secara umum komposisi dari sebuah tim pengembang perangkat lunak yaitu :

* *Owner* / Klien, bersama dengan developer sebagai bagian terpenting dalam proyek, tugas dari klien menentukan fungsi dari perangkat lunak yang akan di buat, melakukan testing dan memberikan feedback.
* Manajer / *Scrum Master*, bertugas mengkolaborasikan developer dengan klien, membuat dan mengevaluasi target pengerjaan perangkat lunak.
* Sistem Analis, membuat arsitektur sistem dari perangkat lunak yang akan dibuat.

Developer, merupakan titik vital dalam tim, tanpa developer perangkat lunak tidak akan bisa dibuat.

* Kelebihan dan Kekurangan

**Kelebihan** Beberapa kelebihan dari *agile* diantaranya[[8]](https://id.wikipedia.org/wiki/Agile_Development_Methods#cite_note-Silverburg.2CA._2012-8) :

* 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.