**EKKA OKTAVIYANTI**

**A11.2019.11803**

**TUGAS PROYEK PERANGKAT LUNAK**

**MODEL PENGEMBANGAN PERANGKAT LUNAK**

1. **RAD (Rapid Application Development)**

* Pengertian RAD

RAD adalah metode pengembangan perangkat lunak yang pada pengembangan sistemnya menggunakan pendekatan berorientasi objek. RAD juga merupakan proses pengembangan yang menekankan siklus pengembangan dalam waktu yang singkat.

* Tahapan dalam metode RAD dibagi menjadi 4 Tahapan antara lain.

1. Perencanaan Kebutuhan

Perencanaan kebutuhan merupakan tahap awal dalam metode pengembangan RAD. PAda tahap ini dilakukan identifikasi permasalhan dan pengumpulan data yang diperoleh dari user atau stakeholder user yang bertujuan untuk mengidentifikasi tujuan dari sistem dan kebutuhan informasi yang di inginkan. Pada tahap awal ini keterlibatan pengembang dan user sangatlah penting dalam mengidentifikasi kebutuhan untuk pengembangan sistem.

1. Desain Sistem

Desain sistem merupakan tahap kedua dalam metode pengembangan RAD. Dalam tahap ini sangat dibutuhkan keaktifan pengguna, karena pada tahap ini dilakukan proses desain dan proses perbaikan desain secara berulang ulang apabila masih terdapat ketidaksesuaian desain terhadap kebutuhan pengguna yang telah di identifikasi pada tahap sebelumnya.

1. Pengembangan

Pada tahap pengembangan, desain sistem yang telh dibuat pada tahap sebelumnya dan telah disepakati, kemudian diubah kedalam aplikasi dari versi beta sampai dengan versi final. Pada tahapan ini juga programmer harus terus-menerus melakukan kegiatan pengembangan dan integerasi dengan bagian-bagian lainnya sambal terus mempertimbangkan feedback dari pengguna atau klien. Jika proses berjalan lancar maka dapat berlanjut ke tahapan berikutnya, sedangkan jika aplikasi yang dikembangkan belum menjawab kebutuhan, programmer akan kembali ke tahapan desain sistem.

1. Implementasi

Tahapan ini merupakan tahapan dimana programmer menerapkan desain dari suatu sistem yang telah disetujui pada tahapan sebelumnya. Sebelum sistem diterapkan, terlebih dahulu dilakukan proses pengujian terhadap program untuk mendeteksi kesalahan yang ada pada sistem yang dikembangkan. Pada tahap ini biasa memberikan tanggapan akan sistem yang sudah dibuat dan mendapat persetujuan mengenai sistem tersebut.

* Kelebihan dan Kekurangan RAD

Sebagai salah satu metode dalam mengembangkan perangkat lunak, RAD tentu memiliki beberapa kekurangan dan kelebihan.

Kelebihan dan kekurangan dalam pengembangan aplikasi dengan menggunakan metode RAD adalah sebagai berikut.

Kelebihan

* Dapat menggunakan kembali komponen yang ada (*reusable object*) sebelumnya sehingga tidak perlu membuat dari awal lagi.
* Integrasi proses yang lebih cepat dan efektif.
* Penyesuaian kebutuhan dan keinginan user menjadi lebih mudah.
* Memperkecil kemungkinan kesalahan atau error.

Kekurangan

* Memerlukan kolaborasi tim yang kuat dan memadai.
* Memerlukan komitmen yang kuat antara pengembang dan stakeholder.
* Hanya cocok diterapkan untuk proyek kecil dan memiliki waktu pengerjaan yang singkat.
* Hanya cocok digunakan untuk mengembangkan aplikasi yang memiliki fokus pada suatu fitur untuk dijadikan modular terpisah.
* Kapan Saat yang Tepat Menerapkan Metode RAD

Meskipun metode RAD merupakan pengembangan aplikasi yang popular, tidak dipungkiri juga jika metode RAD tidak selalu ideal dan optimal untuk diterapkan.

Berdasarkan proses yang telah dijelaskan, ada beberapa yang membuat penerapan RAD akan lebih maksimal.

1. Ketika Memiliki Tim yang Solid

Pada tahapan metode RAD, feedback dan testing ulang prototypes adalah poin penting dalam menerapkan metode ini. Oleh karena itu, tim yang solid sangatlah diperlukan ketika ingin memakai metode RAD.

Tim yang dimaksud bukan hanya tim internal dalam pengembangan sistem, akan tetapi pengguna yang terlibat dalam testing aplikasi, serta anggota lain dan juga stakeholder yang terlibat dalam pengembangan aplikasi/sistem.

Mereka semua perlu memiliki komitmen yang kuat untuk terus menerus mengikuti proses yang berulang-ulang pada pengembangan aplikasi/sistem. Dengan begitu mereka bisa memberikan feedback yang berkualitas serta patuh dengan timeline yang telah disepakati bersama.

1. Ketika Memilki Modal yang Besar

Metode RAD adalah metode yang hanya memerlukan jangka waktu yang singkat. Karena keefektifan dan keefisiensi dari metode RAD, diperlukan tim teknis ahli yang memilki bayaran yang tinggi. Oleh karena itu perlu memiliki modal yang besar untuk penerapan metode RAD yang maksimal.

1. Ketika hanya Memiliki Sedikit Waktu

Metode pengembangan aplikasi RAD sangatlah cocok diterapkan ketika, proyek  pengembangan aplikasi/sistem yang diambil memiliki batasan waktu yang pendek.

Metode ini tidak menghabiskan banyak waktu di tahap perencanaan. Sebaliknya, proses pengembangan aplikasi bisa berjalan fleksibel dan sesuai kebutuhan.

Ditambah lagi pada proses testing dan feedback yang intens membuat aplikasi yang dihasilkan lebih stabil.

1. **AGILE**

* Pengertian Agile

[Agile](https://hacktiv8.com/parttime/scrum) adalah sebuah metode *software development* yang mencakup website, *web application*, dan *mobile application* yang berfokus untuk menghasilkan software berkualitas tinggi secara konsisten dengan mengurangi biaya proyek dan meningkatkan nilai jual suatu bisnis. Secara definisi, Agile dapat diartikan sebagai sebuah pendekatan pada *project management* dengan menggunakan teknik iterasi dan bertahap secara dinamis (atau dikenal dengan *Sprint*) dalam proses pembuatan suatu produk.

* Metode Agile Development

Secara konsep ada banyak macam penerapan dari metode Agile. Beberapa tahapan Agile dalam pengerjaan *software development* antara lain:

1. Penemuan

Sebelum memulai pelaksanaan sebuah proyek baru, kita perlu memahami visi atau objektif dari client atau user yang memiliki kepentingan dan kepemilikan dalam proyek tersebut. *Software development* dengan metode Agile dimulai dengan sebuah riset untuk mencapai pengertian akan tujuan yang ingin dicapai oleh client, tantangan yang ada, iklim bisnis sekarang, serta customer dan user tersebut. Tahap pertama ini termasuk memastikan adanya pengertian yang sama di antara client, *Project Manager*, *Designer*, *Developer*, dan *Product Owner*.

1. Product Backlog

Setelah proses penemuan, tim akan mulai bekerja bersama untuk membuat sebuah *Product Backlog* tingkat tinggi yang berisi daftar fitur yang akan berguna bagi client dan user. *Product Owner* akan bekerja sama dengan client untuk memprioritaskan fitur dan menentukan urutan dari bagaimana fitur akan dirancang, *develop*, *test*, dan diterapkan. Hal ini akan membantu tim untuk tetap fokus dalam memberikan fitur dengan nilai tinggi sebelum bekerja pada tugas yang berprioritas lebih rendah.

1. Pengulangan

Setelah memastikan bahwa tim mengerti visi yang dimiliki oleh client dan membuat sebuah *backlog* yang tepat, tim akan mulai menerapkan fitur-fitur yang telah dibuat dalam sebuah pengulangan yang diukur oleh waktu yang dinamakan *Sprint*. *Sprint* sendiri berlangsung dalam durasi 1-4 minggu, tergantung dari besar proyek dan durasi waktu yang ada. Setiap *Sprint* akan memenuhi objektif atau target yang ada pada product *backlog*.

1. Penerusan Siklus

*Sprint* tambahan akan dilakukan sesuai dengan yang dibutuhkan untuk menghasilkan fitur tambahan dan memasukan *feedback* yang didapat dari *review* sebelumnya. Setiap *Sprint* baru harus menghasilkan perkembangan dari hasil *Sprint* sebelumnya dan juga menghasilkan fitur baru ke dalam sistem.

Pada akhirnya, metodologi Agile memiliki keuntungan jangka panjang dalam kinerja *software development* tim kamu dalam meraih kesuksesan maupun menciptakan etos kerja yang efektif dan efisien.

* Keuntungan Agile Software Development

Ada banyak keuntungan dalam mengimplementasikan metode Agile dalam sebuah pekerjaan. Berikut beberapa keuntungan yang bisa mereka dapatkan dengan metode pengembangan Agile.

* Keterlibatan Stakeholder

Agile memberikan banyak kesempatan dalam keterlibatan stakeholder dan tim yang mencakup sebelum, selama, dan setelah proses *Sprint*. Dengan melibatkan client di setiap langkah proyek, ada kolaborasi yang tinggi antara client dan tim proyek. Hal ini juga memberi lebih banyak kesempatan bagi tim untuk benar-benar memahami visi client. Tim yang mempresentasikan software lebih awal akan meningkatkan kepercayaan para stakeholder pada kemampuan tim untuk membuat software berkualitas tinggi dan mendorong mereka untuk lebih terlibat dalam proyek.

* 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 dan Mudah Diprediksi

Dengan menggunakan jadwal tetap *Sprint* dalam kurun waktu 1-4 minggu, fitur baru bisa dibuat dengan cepat dan lebih sering, juga dengan tingkat prediktabilitas yang tinggi. Ini juga memberi kesempatan untuk memasang atau menguji versi beta dari software lebih awal dari yang direncanakan.

* Biaya & Jadwal Bisa Ditentukan

Karena setiap *Sprint* memiliki durasi yang tetap, biaya suatu produk dapat diprediksi dan ditentukan dalam pekerjaan yang akan dilakukan dalam jangka waktu 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 tim harus berfokus dalam memberikan subset fitur produk yang disepakati di setiap *Sprint*, ada kesempatan untuk terus memperbaiki dan memprioritaskan isi product *backlog*. Item *backlog* baru atau yang telah diubah dapat direncanakan dalam pertemuan *Sprint* berikutnya untuk memberikan kesempatan dalam mengenalkan perubahan dalam beberapa minggu.

* Berfokus Pada Nilai Bisnis

Dengan melibatkan client dalam menentukan prioritas fitur, tim memahami hal yang paling penting bagi bisnis client dan dapat memberikan fitur yang memberikan nilai bisnis paling banyak.

* Fokus Pengguna

Agile biasanya menggunakan user *feedback* dengan kriteria yang berfokus pada bisnis untuk menentukan fitur produk. Dengan memfokuskan fitur pada kebutuhan nyata tiap pengguna, setiap fitur memberikan nilai (*value*) dan tidak hanya sekedar komponen IT. Ini juga memberikan kesempatan untuk menguji beta software setelah proses *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, tim proyek dapat berfokus pada pengembangan, pengujian, dan kolaborasi berkualitas tinggi. Selain itu, tim yang kerap melakukan ulasan dalam *Sprint* akan dapat meningkatkan kualitas dan memperbaiki permasalahan dengan cepat serta mengidentifikasi kesalahan dengan ekspektasi.

1. **SCRUM**

* Pengertian Scrum

[Scrum](https://hacktiv8.com/parttime/scrum) adalah sebuah metode iteratif yang termasuk dalam metode Agile tentang cara mengelola dan menjalankan sebuah proyek. Dengan menggunakan metode Scrum, kamu bisa mengelola segala macam proyek mulai dari pembuatan software, website, hardware, marketing, hingga *event planning*. Menggunakan Scrum merupakan *framework* terbaik untuk merevolusi cara kerja tim dalam menyelesaikan pekerjaan. Dalam metodologi Scrum setiap “*Sprint*” dimulai dengan meeting singkat untuk perencanaan dan diakhiri dengan review.

* Tahapan Metode Scrum

Ada beberapa tahap yang harus dilakukan dalam melakukan [*Scrum Development*](https://blog.hacktiv8.com/agile-software/). Berikut cara kerja Scrum.

1. Tentukan Tim Scrum

Dalam menentukan tim Scrum, tim harus beranggotakan 5-9 orang. Semua anggota harus memiliki skill masing-masing dan dapat mencakup untuk menjadi developer, penguji, pendukung, perancang, analisis bisnis, hingga quality check. Semua anggota akan bekerja sama dan bertanggung jawab untuk memastikan bahwa mereka akan memproduksi produk yang akan mereka presentasikan di akhir sesi *Sprint*.

1. Tentukan Waktu *Sprint*

*Sprint* adalah tenggang waktu yang berlangsung antara 7 dan 30 hari. Tentukan deadline-nya dalam sebuah meeting perencanaan sesuai dengan komitmen tim untuk menyelesaikan proyek ini. Pada akhir Sprint, biasakan untuk melakukan review atau ulasan dengan demonstrasi hasil kerja. Pada tahap ini, segala perbaikan akan dikerjakan agar dapat direncanakan pada sesi *Sprint* berikutnya.

1. Tunjuk Seorang Scrum Master

Scrum Master adalah seorang katalisator untuk sebuah kelompok Scrum. Mereka memastikan bahwa kelompok Scrum bekerja secara efektif dan efisien. Jika terjadi hambatan, Scrum Master akan menindaklanjuti dan menyelesaikan masalah tersebut. Scrum Master bisa disebut sebagai manajer proyek dalam tim, tapi Scrum Master tidak boleh mendikte apa yang tim kerjakan dan tidak boleh terlibat dalam pengelolaan mikro (micro-management). Scrum Master akan membantu tim dalam merencanakan pekerjaan *Sprint* yang akan datang.

1. Tunjuk Pemilik Produk (*Product Owner*)

*Product Owner* merupakan seseorang yang dapat bertanggung jawab untuk memastikan tim menghasilkan sebuah produk yang bisa dipresentasikan atau dipasarkan ke bisnis, klien, atau siapapun yang menginginkan hasil proyek tersebut. *Product Owner* biasanya menuliskan persyaratan sehubungan dengan apa yang diinginkan dari produk tersebut dalam bentuk sebuah cerita, kemudian memprioritaskan poin-poin tertentu dalam proses pembuatan, dan memasukkannya ke product *backlog*.

1. Buat *Product Backlog* Awal

Product *backlog* adalah daftar keinginan yang berupa semua cerita pengguna yang diharapkan dapat dibuat dan diselesaikan dalam proyek tersebut. Cerita yang paling penting harus berada di urutan teratas hingga keseluruhan cerita tersusun secara teratur berdasarkan urutan kepentingannya. Sebuah cerita biasanya berisi dua jenis pekerjaan, yaitu *Epics* dan *Stories*. *Epics* merupakan cerita tingkat tinggi yang sketsanya sangat kasar tanpa banyak detail. Sebuah *Epics* biasanya dapat dibagi menjadi beberapa cerita. Dalam suatu cerita biasanya akan dipecah menjadi beberapa tugas terpisah sehingga tim dapat bekerja dan melaporkan progres pengerjaannya. Di satu sisi, *Stories* mencakup persyaratan lebih rinci tentang hal yang harus atau mungkin dilakukan. Stories juga bisa memiliki beberapa tipe seperti *development*, *bug*, maupun tugas (*task*). *Stories* dapat ditulis dan ditambahkan ke backlog produk kapanpun dan oleh siapapun. Jika skala prioritas *Epics* meningkat, disarankan masukkan detail yang lebih rinci agar tim dapat mulai mengerjakannya. *Product Owner* dapat memprioritaskan kembali *backlog* sesuai dengan kebutuhannya.

1. Rencanakan dan Mulai *Sprint*

Berdasarkan prioritas *backlog*, sebuah tim dapat mulai mengambil item dari daftar yang paling atas. Setelah itu, tim biasanya melakukan *brainstorming* dan memutuskan daftar tugas (*task list*) dan berapa banyak yang dapat mereka selesaikan dalam *Sprint* mendatang. Apabila semua anggota tim setuju, *Sprint* dapat dimulai dan tim dapat mulai mengerjakan proyeknya.

1. Tutup dan Mulai *Sprint* Berikutnya

Jika batas tenggang waktu telah tiba dan semua pekerjaan yang direncanakan sudah selesai, tim berhak memutuskan pekerjaan selanjutnya akan dipindahkan ke *Sprint* berikutnya atau tetap berada dalam *backlog*. Dalam menyelesaikan tugas, sebuah tim disarankan untuk melakukan retrospektif yang bertujuan untuk mendiskusikan hal yang sedang berjalan dengan baik dan hal dapat diperbaiki untuk Sprint berikutnya. Setelah itu, pertemuan perencanaan *Sprint* untuk *Sprint* berikutnya dapat dimulai dengan proses yang sama. Tidak ada batasan untuk jumlah *Sprint* kecuali jika ditetapkan dengan *deadline* berdasarkan budget atau waktu atau seluruh *backlog* selesai.

* Kelebihan Scrum

Jadi dengan menggunakan model Scrum, kamu bisa mengelola proyek atau bisnis menjadi lebih efisien, serta membuat cara kerja tim menjadi lebih efektif dan terfokus. Scrum bisa menjadi solusi untuk proyek kamu yang kerap mangkrak atau tertunda-tunda.