



Telkom
University

RIZKY PUTRA ANANDA

1201230029

PENGANTAR RPL

Metodologi pengembangan perangkat lunak

SDLC (Software Development Life Cycle) adalah suatu pendekatan sistematis yang digunakan dalam pengembangan perangkat lunak. Pendekatan ini membantu tim pengembangan perangkat lunak dalam mengatur proses pembuatan, pengembangan, pengujian, dan pemeliharaan perangkat lunak secara efisien dan terstruktur. Tujuan dari SDLC adalah untuk menghasilkan perangkat lunak berkualitas tinggi yang memenuhi kebutuhan pengguna, mengikuti tenggat waktu, dan berada dalam batas anggaran yang ditentukan. (tutorial point, 2019).

-WATERFALL MODEL

Model System Development Life Cycle (SDLC) yang berbeda sering digunakan untuk pengembangan perangkat lunak [Acharya et al, 2020]. Model SDLC menyediakan panduan teori untuk pengembangan software. Model-model SDLC sangat penting untuk evolusi perangkat lunak secara sistematis sehingga dapat dikerjakan dalam batas waktu tertentu dengan kualitas yang baik. Banyak perusahaan mengembangkan program perangkat lunak untuk menyediakan fasilitas bagi kantor. Beberapa masalah terjadi selama tahap awal pengembangan perangkat lunak [Rastogi et al, 2015]. Beberapa alur terorganisir dalam rekayasa perangkat lunak dan program perangkat lunak dibangun untuk menghindari masalah atau hambatan-hambatan yang ada pada tahap awal pengembangan software tersebut. SDLC adalah teknik di mana perangkat lunak dapat dikembangkan dengan sangat detail dan meningkatkan kemungkinan menyelesaikan proyek perangkat lunak dalam batas waktu serta menjaga konsistensi produk perangkat lunak sesuai dengan standar [Arora et al., 2016]. Kerangka kerja (SDLC) memberikan urutan kegiatan yang harus diikuti oleh perancang sistem dan pengembang untuk mengembangkan perangkat lunak. System Development Life Cycle (SDLC) merupakan sebuah metodologi untuk merancang, membangun dan memelihara software atau perangkat lunak.

-SPIRAL MODEL

Model Spiral Model Spiral diperkenalkan pertama kali oleh Barry William Boehm, seorang professor asal Amerika pada tahun 1986. Model tersebut diperkenalkan pada sebuah artikel berjudul “ A spiral model of software development and enhancement”. Model spiral diperkenalkan sebagai model yang menyediakan kerangka kerja (framework) terbaru untuk memandu proses pengembangan perangkat lunak, dimana model ini menggunakan pendekatan berbasis risiko (risk-driven) dan dapat digunakan dengan menggabungkan kelebihan dari model lainnya. Hal ini menjadikan model Spiral memiliki kelebihan dibandingkan dengan model ² model pengembangan perangkat lunak yang sudah ada sebelumnya, yang prosesnya spesifik atau mengikuti prototipe secara ketat. Pendekatan risk-driven pada model Spiral memungkinkan model ini dapat mengakomodasi model lainnya baik yang berorientasi spesifikasi, prototipe, simulasi, transformasi otomatis, atau pendekatan pengembangan perangkat lunak lainnya.

-V-MODEL

V-Model, juga dikenal sebagai V-model rekayasa sistem, menekankan pendekatan pemecahan masalah. Dimulai dari pembagian masalah pada tingkat awal menjadi bagian-bagian yang dapat dikelola. Dari tingkat detail, solusi akhir diintegrasikan kembali ke tingkat awal. Pada setiap tingkat, seseorang dapat membandingkan solusi atau bagian solusi dengan masalah atau bagian masalah yang sedang dipecahkan.

-ADAPTIVE SOFTWARE DEVELOPMENT

Perubahan yang tak terhindarkan sering memengaruhi pengembangan perangkat lunak, yang merupakan proses yang selalu berubah. Adaptive Software Development (ASD) metodologi dibuat pada awal 1990-an oleh John Highsmith dan Sam Bayer, dua manajer proyek. Ini adalah modifikasi dari kerangka kerja Rapid Application Development (RAD), yang terkenal dengan pendekatan ketat untuk proses pengembangan perangkat lunak (J. Highsmith, 2009). Di tengah lingkungan teknologi yang terus berkembang dan tuntutan pasar yang berubah dengan cepat, pendekatan yang kaku dan terlalu terstruktur mungkin tidak lagi cukup efektif. Inilah tempat di mana konsep "adaptive development" atau pengembangan yang adaptif muncul sebagai solusi yang sesuai. Pendekatan ini mengakui kompleksitas lingkungan pengembangan perangkat lunak dan memprioritaskan fleksibilitas serta responsibilitas terhadap perubahan. Adaptive development, juga dikenal sebagai adaptive methodologies, adalah paradigma dalam siklus hidup pengembangan perangkat lunak yang berfokus pada kemampuan untuk beradaptasi dengan perubahan yang tidak dapat dihindari.

-INCREMENTAL MODEL

Pengembangan perangkat lunak telah mengalami evolusi penting selama beberapa dekade terakhir, memunculkan banyak pendekatan dan metode berbeda untuk mencapai tujuan ini. Salah satu pendekatan yang mendapat perhatian luas dalam rekayasa perangkat lunak adalah model inkremental. Model ini memberikan pendekatan yang adaptif dan iteratif dalam mengembangkan sistem perangkat lunak, yang memungkinkan tahapan pembangunan yang bertahap dengan peningkatan fungsionalitas di setiap tahapnya. Dengan mengakui kompleksitas dan dinamika yang umumnya terlibat dalam proyek perangkat lunak, model Incremental menghadirkan solusi yang memungkinkan para pengembang untuk merespons perubahan dengan lebih efektif sambil memberikan hasil tanggung jawab secara berkelanjutan kepada pihak-pihak terkait. Dalam bab ini, kami akan membahas konsep dasar dari model Incremental, tahapantahapan yang terlibat, metode dan strategi yang dapat diterapkan, serta analisis keuntungan dan kendalanya dalam konteks pengembangan perangkat lunak modern

-RAPID APPLICATION DEVELOPMENT

Rapid Application Development (RAD) mengacu pada strategi pengembangan perangkat lunak adaptif yang berpusat di sekitar pembuatan prototipe dan umpan balik yang dipercepat, sementara kurang fokus pada desain yang teliti. Teknik RAD dicirikan oleh fokus pada pengembangan dan pembuatan prototipe, berlawanan dengan perencanaan ekstensif, dalam keseluruhan strateginya. Pengembangan aplikasi yang cepat memungkinkan pengembang untuk mengimplementasikan beberapa iterasi dan perubahan secara efisien pada produk tanpa perlu memulai proses pengembangan lagi. Praktik ini membantu menjamin bahwa hasil akhir berpusat pada kualitas dan sesuai dengan spesifikasi pelanggan akhir. Salah satu kelemahan signifikan yang terkait dengan model air terjun adalah ketidakmampuan pengujian untuk meninjau kembali dan memodifikasi fungsi dan fitur dasar produk setelah beralih ke fase pengujian. Hal ini pada akhirnya menghasilkan tim yang memiliki solusi perangkat lunak yang mungkin selaras atau tidak selaras dengan perubahan kebutuhan pengguna akhir. Konsep Rapid Application Development (RAD) berasal dari tahun 1980-an, menunjukkan kehadiran historisnya daripada kebaruan.

-SCRUM

Di era digital saat ini, pengembangan perangkat lunak telah menjadi komponen penting dalam kehidupan sehari-hari dan bisnis. Teknologi yang terus berkembang menuntut pendekatan yang fleksibel, adaptif, dan efisien dalam mengelola proyek perangkat lunak. Salah satu metode yang telah menonjol dalam dekade terakhir adalah Scrum. Scrum adalah kerangka kerja agile yang menawarkan pendekatan iteratif dan inkremental dalam pengembangan perangkat lunak. Keunikan dan efektivitas Scrum dalam memenuhi kebutuhan pasar yang cepat berubah menjadikannya pilihan populer di kalangan organisasi modern. Buku ini ditujukan untuk membantu para profesional, dosen, peneliti, dan

siapa saja yang tertarik dalam manajemen proyek perangkat lunak, khususnya dalam menggunakan Scrum. Dengan menyajikan konsep, praktik, dan analisis mendalam, buku ini bertujuan untuk menjadi panduan komprehensif yang menghubungkan teori dan praktik dalam konteks nyata.

-EXTREME PROGRAMMING (XP) MODEL

Metode pengembangan sistem Agile mungkin terdengar familiar ditelinga kita, namun pembahasan terkait model Extreme Programming (XP) masih cukup asing. Banyak dari kita yang merasa kurang yakin dengan penerapan model XP ini, namun, jangan melewati pemahaman terkait model XP. Bab 9 ini akan memberikan kita pemahaman melalui semua yang perlu kita ketahui tentang Model XP sehingga dapat kita manfaatkan secara optimal. Extreme Programming (XP) adalah salah satu dari beberapa Proses Agile yang populer. Model ini telah terbukti sangat sukses di banyak perusahaan dengan berbagai ukuran dan industri di seluruh dunia. Model XP banyak berhasil diterapkan karena lebih menekankan pada kepuasan pelanggan/pengguna. Keunikan pada model XP juga dapat dilihat pada siklus pengembangannya yang fleksibel. Model XP memungkinkan penggunaannya untuk memenuhi kebutuhan pada proses bisnis secara dinamis mengikuti perkembangan tanpa harus menunggu di waktu tertentu. Dalam Extreme Programming, pengembang perangkat lunak diberdayakan untuk menghadapi perubahan permintaan pelanggan secara lebih optimal, bahkan pada akhir siklus pengembangan.

-DYNAMIC SYSTEM DEVELOPMENT METHOD (DSDM)

Dynamic System Development Method (DSDM) adalah teknik pengembangan yang menekankan pada proses berulang dan peningkatan berkelanjutan, selaras dengan ide dasar metodologi Agile. Ini secara aktif melibatkan pengguna dan pelanggan secara terus menerus sepanjang keseluruhan proses pengembangan (Siau, 2007; Ibrahim and Awany, 2017; Pressman and Maxim, 2020). Asal-usul DSDM dapat ditelusuri kembali ke tahun 1990-an, periode di mana teknik Rapid Application Development (RAD) mendapatkan popularitas yang signifikan. Ini memfasilitasi penyelesaian masalah yang dipercepat melalui pemanfaatan teknik prototyping. DSDM muncul sebagai tanggapan terhadap keterbatasan yang diamati dalam metodologi sekuensial seperti air terjun dan RAD. Tujuannya adalah untuk mengintegrasikan kekuatan RAD sekaligus menggabungkan pemahaman yang komprehensif tentang pemangku kepentingan dan keterlibatan aktif mereka dalam proses pembangunan (Stapleton, 2003; Coyle and Conboy, 2009; Ibrahim and Awany, 2017). DSDM awalnya diperkenalkan pada tahun 1994, sehingga menawarkan kerangka kerja sistematis untuk RAD. Meskipun kadang-kadang disorganisasi akibat kekurangan metode standar.

-LEAN DEVELOPMENT

Pendekatan lean pada dasarnya berfokus pada pandangan manajemen yang mengedepankan eliminasi proses yang tidak perlu serta mendorong kerjasama yang efektif antara semua pihak yang terlibat dalam setiap tahapan pengembangan, mulai dari perencanaan hingga pengiriman produk akhir kepada pengguna (Janes, 2015). Awalnya, istilah lean dikenal sebagai lean production. Konsep ini muncul dari penelitian di Massachusetts Institute of Technology (MIT) dengan tujuan untuk memahami mengapa perusahaan otomotif Jepang seperti Toyota mampu mengungguli perusahaan-perusahaan Amerika dalam menguasai pangsa pasar (Jonsson et al., 2013; Abdullah and Ward, 2016). Prinsip-prinsip lean production sangat berkaitan dengan Toyota Production System (TPS), yang dikembangkan di Jepang pada tahun 1940-an (Womack and Jones, 1997). Sistem ini merubah cara kerja industri otomotif dan mendorong Jepang menjadi pemimpin global, meskipun konsep serupa pernah digunakan oleh Ford sebelumnya (Womack and Jones, 1997; Cawley et al., 2013). Kesuksesan penerapan lean production dalam TPS untuk industri otomotif mendorong perusahaan-perusahaan, baik di sektor yang sama maupun sektor lainnya, secara cepat mengadopsi lean production dalam kegiatan produksinya demi meraih competitive advantage (Womack and Jones, 1997; Cawley et al., 2013).