

Metodologi Manajemen Proyek



Standardisasi proyek tidak mungkin dilakukan karena setiap proyek memiliki persyaratan dan proses yang berbeda. Namun, proses manajemen proyek dapat distandarisasi dengan menggunakan beberapa metodologi proyek yang telah ditentukan sebelumnya.

Terdapat beberapa metodologi manajemen proyek yang umum digunakan untuk mengelola proyek seperti...

- **Model Air Terjun:**

Model Waterfall didasarkan pada model inkremental. Dalam metode ini, proyek direncanakan dari atas ke bawah sekaligus. Seluruh proyek akan dibagi menjadi tujuh fase berurutan. Dalam metode ini, semua fitur dikirimkan sekaligus di akhir siklus.

- **Lincah:**

Dalam metode ini, desain proses dipecah menjadi model-model individual. Proses pengembangan bersifat iteratif, dan di akhir setiap iterasi, fitur-fitur produk yang siap dikirimkan diserahkan kepada pelanggan.

- **PANGERAN 2:**

Ini adalah metodologi manajemen proyek umum dan dapat digunakan untuk proyek dalam skala apa pun. Ini adalah pendekatan berorientasi proses di mana setiap proses memiliki masukan dan keluaran, tugas dan aktivitas untuk diselesaikan.

- **RAD:**

Pengembangan aplikasi cepat (Rapid Application Development/RAD) adalah metodologi manajemen proyek yang digunakan untuk pengembangan perangkat lunak, di mana tujuan utamanya adalah mengembangkan aplikasi lebih cepat dengan kualitas tinggi. Seluruh proses pengembangan didefinisikan menjadi empat fase.

- **Kanban:**

Dalam metode Kanban, papan digunakan tempat catatan tempel ditempatkan di kolom sesuai dengan status tugas. Status umum yang digunakan untuk tugas proyek meliputi "dalam antrian", "sedang dikerjakan", dan "baru saja selesai".

- **Six Sigma:**

Metode Six Sigma lebih disukai untuk proyek yang membutuhkan pengukuran yang tepat. Metode ini menggunakan statistik untuk meningkatkan proses dan meminimalkan cacat pada produk atau layanan. Six Sigma menggunakan dua set metodologi yang berbeda, yaitu DMAIC dan DMADV.

- **SDLC:**

SDLC (System Development Life Cycle) adalah pendekatan umum untuk manajemen proyek yang mencakup konsep-konsep dasar manajemen proyek seperti analisis kebutuhan, desain, pelatihan, pengiriman, dan dukungan.

- **Pemrograman Ekstrem (XP):**

XP sering digunakan untuk manajemen proyek TI. Seperti Scrum, XP juga bekerja secara iteratif, tetapi durasi iterasinya biasanya singkat. Lingkup dalam setiap iterasi bersifat fleksibel selama pekerjaan belum dimulai.

- **Metodologi Kristal:**

Metode ini cukup mendukung untuk kontrak harga tetap. Iterasi dalam metode kristal dirilis secara berkala. Metode ini dapat mendeteksi kekurangan apa pun pada tahap awal proyek, dan kesalahan cenderung lebih kecil kemungkinannya terjadi.

- **CCPM (Manajemen Proyek Rantai Kritis):**

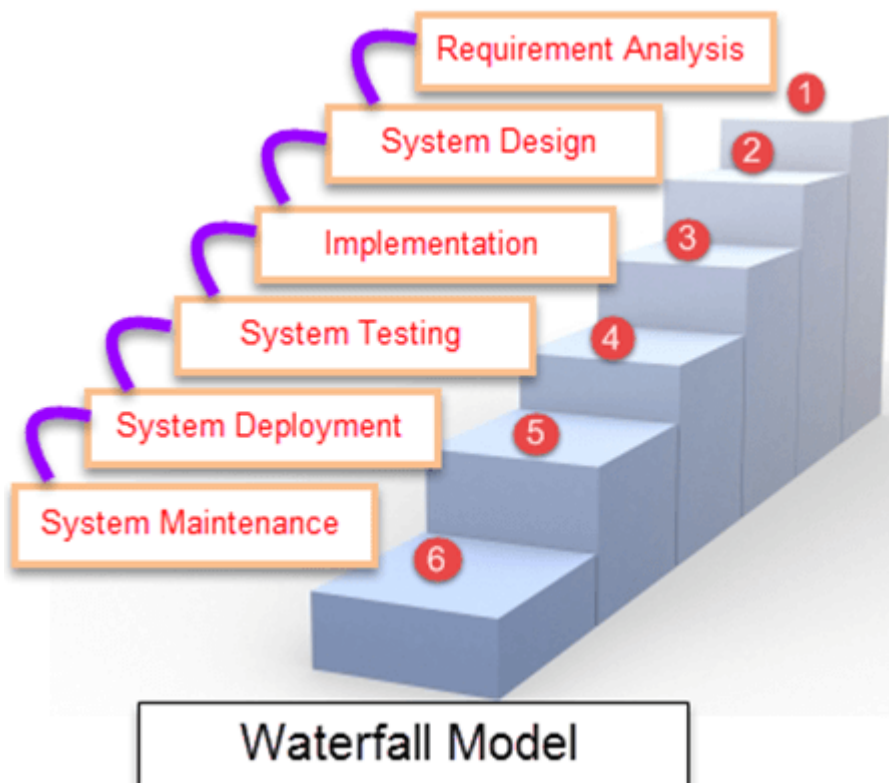
Metode ini mengurangi durasi aktivitas hingga 50% sehingga proyek dapat diselesaikan lebih awal. Metode ini menggunakan manajemen buffer untuk mengontrol rencana dan memastikan kesalahan seminimal mungkin.

Ringkasan :

Tidak ada standar khusus untuk menyelesaikan proyek yang sangat bervariasi. Namun, metodologi tertentu membantu menyelesaikan proyek dengan cara yang lebih efisien, mulai dari inisiasi hingga implementasi dan penutupan.

Apa itu Model Waterfall?

Model Waterfall adalah model sekuensial yang membagi pengembangan perangkat lunak ke dalam berbagai fase. Setiap fase dirancang untuk melakukan aktivitas spesifik selama siklus hidup pengembangan perangkat lunak (SDLC). Model ini diperkenalkan pada tahun 1970 oleh Winston Royce.



Berbagai Fase Model Waterfall dalam Rekayasa Perangkat Lunak

Fase yang berbeda	Aktivitas yang dilakukan di setiap tahap
Tahap Pengumpulan Persyaratan	• Pada fase ini, persyaratan rinci dari sistem perangkat lunak yang akan dikembangkan dikumpulkan dari klien.
Tahap Desain	• Rencanakan bahasa pemrograman yang akan digunakan, misalnya Java, PHP, .NET. • atau basis data seperti Oracle, MySQL, dll. • Atau detail teknis tingkat tinggi lainnya dari proyek tersebut.
Panggung yang Dibangun	• Setelah tahap desain, selanjutnya adalah tahap pembangunan, yang tidak lain adalah pengkodean perangkat lunak.
Tahap Pengujian	• Pada fase ini, Anda menguji perangkat lunak untuk memverifikasi bahwa perangkat lunak tersebut dibangun sesuai dengan spesifikasi yang diberikan oleh klien.
Tahap penyebaran	• Sebarkan aplikasi di lingkungan yang bersangkutan.

Tahap pemeliharaan	Setelah sistem Anda siap digunakan, Anda mungkin perlu mengubah kode sesuai permintaan pelanggan di kemudian hari.
---------------------------	--

Kapan Menggunakan Model Air Terjun SDLC?

Model air terjun dapat digunakan ketika

- Persyaratan tidak sering berubah.
- Aplikasi ini tidak rumit dan besar.
- Proyek ini singkat
- Persyaratannya jelas.
- Lingkungannya stabil.
- Teknologi dan alat yang digunakan tidak dinamis dan bersifat stabil.
- Sumber daya tersedia dan terlatih.

Keuntungan dan Kerugian Model Waterfall

Keuntungan	Kekurangan
Sebelum memasuki fase pengembangan selanjutnya, setiap fase harus diselesaikan terlebih dahulu.	Kesalahan hanya dapat diperbaiki selama fase tersebut.
Cocok untuk proyek-proyek kecil di mana persyaratannya terdefinisi dengan baik.	Hal ini tidak diinginkan untuk proyek kompleks yang persyaratannya sering berubah.

● Mereka harus melakukan uji penjaminan mutu (Verifikasi dan Validasi) sebelum menyelesaikan setiap tahapan.	● Masa pengujian dilakukan cukup terlambat dalam proses pengembangan.
● Dokumentasi yang rinci dilakukan pada setiap fase siklus pengembangan perangkat lunak.	● Dokumentasi menghabiskan banyak waktu pengembang dan penguji.
● Proyek ini sepenuhnya bergantung pada tim proyek dengan intervensi klien yang minimal.	● Masukan berharga dari klien tidak dapat dimasukkan dalam fase pengembangan yang sedang berlangsung.
● Perubahan apa pun pada perangkat lunak dilakukan selama proses pengembangan.	● Perubahan kecil atau kesalahan yang muncul pada perangkat lunak yang sudah jadi dapat menyebabkan banyak masalah.

Metode Agile vs Waterfall

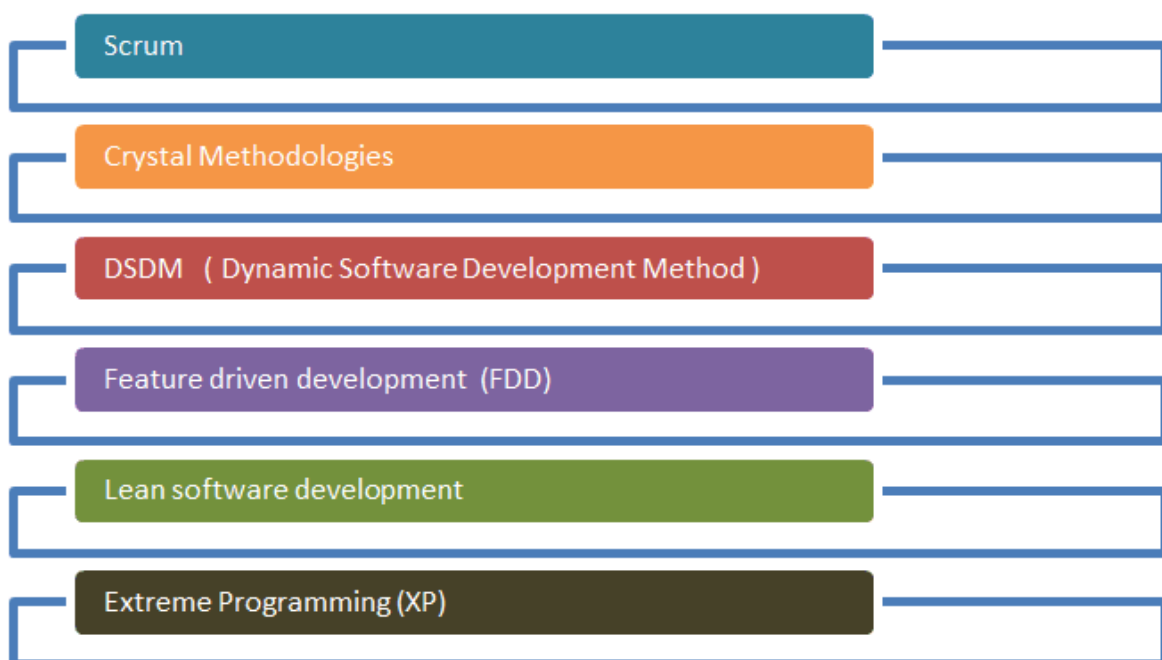
Model Agile dan Waterfall adalah dua metode berbeda untuk proses pengembangan perangkat lunak. Meskipun berbeda dalam pendekatannya, kedua metode ini berguna pada waktu-waktu tertentu, tergantung pada kebutuhan dan jenis proyeknya.

Model Agile	Model Air Terjun
● Metode Agile mengusulkan pendekatan inkremental dan iteratif untuk desain perangkat lunak.	● Pengembangan perangkat lunak berlangsung secara berurutan dari titik awal ke titik akhir.
● Proses agile dipecah menjadi model-model individual yang dikerjakan oleh para desainer.	● Proses desain tidak dipecah menjadi model-model individual.
● Pelanggan memiliki kesempatan sejak awal dan sering untuk melihat produk serta membuat keputusan dan perubahan pada proyek.	● Pelanggan hanya dapat melihat produk di akhir proyek.
● Model Agile dianggap tidak terstruktur dibandingkan dengan model Waterfall.	● Model waterfall lebih aman karena sangat berorientasi pada perencanaan.
● Proyek kecil dapat diimplementasikan dengan sangat cepat. Untuk proyek besar, sulit untuk memperkirakan waktu pengembangannya.	● Segala jenis proyek dapat diperkirakan dan diselesaikan.

● Kesalahan dapat diperbaiki di tengah proyek.	● Barulah di tahap akhir, seluruh produk diuji. Jika ditemukan kesalahan persyaratan atau ada perubahan yang perlu dilakukan, proyek harus dimulai dari awal.
● Proses pengembangan bersifat iteratif, dan proyek dieksekusi dalam iterasi singkat (2-4) minggu. Perencanaan sangat minim.	● Proses pengembangan dilakukan secara bertahap, dan setiap fase jauh lebih besar daripada iterasi. Setiap fase diakhiri dengan deskripsi rinci tentang fase berikutnya.
● Dokumentasi mendapat prioritas lebih rendah dibandingkan pengembangan perangkat lunak.	● Dokumentasi adalah prioritas utama dan bahkan dapat digunakan untuk melatih staf dan meningkatkan perangkat lunak dengan tim lain.
● Setiap iterasi memiliki fase pengujiannya sendiri. Hal ini memungkinkan penerapan pengujian regresi setiap kali fungsi atau logika baru dirilis.	● Fase pengujian baru dijalankan setelah fase pengembangan, karena bagian-bagian yang terpisah belum sepenuhnya berfungsi.
● Dalam pengujian agile, ketika sebuah iterasi berakhir, fitur-fitur produk yang siap dikirim akan diserahkan kepada pelanggan. Fitur-fitur baru dapat langsung digunakan setelah pengiriman. Hal ini bermanfaat jika Anda memiliki hubungan yang baik dengan pelanggan.	● Semua fitur yang dikembangkan dikirimkan sekaligus setelah fase implementasi yang panjang.

• Penguji dan pengembang bekerja sama	• Para penguji bekerja secara terpisah dari para pengembang.
• Di akhir setiap sprint, dilakukan pengujian penerimaan pengguna.	• Persetujuan pengguna dilakukan di akhir proyek.
• Hal ini membutuhkan komunikasi yang erat dengan para pengembang dan bersama-sama menganalisis persyaratan serta perencanaan.	• Pengembang tidak terlibat dalam proses persyaratan dan perencanaan. Biasanya, terjadi penundaan waktu antara pengujian dan pengkodean.

Metodologi Agile

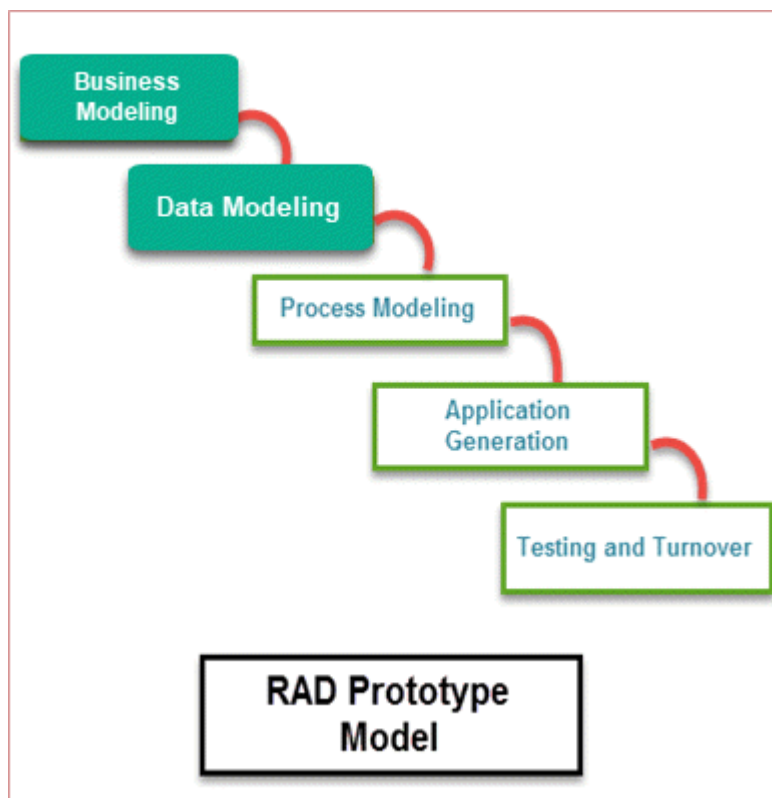


Apa itu Model RAD (Rapid Application Development)?

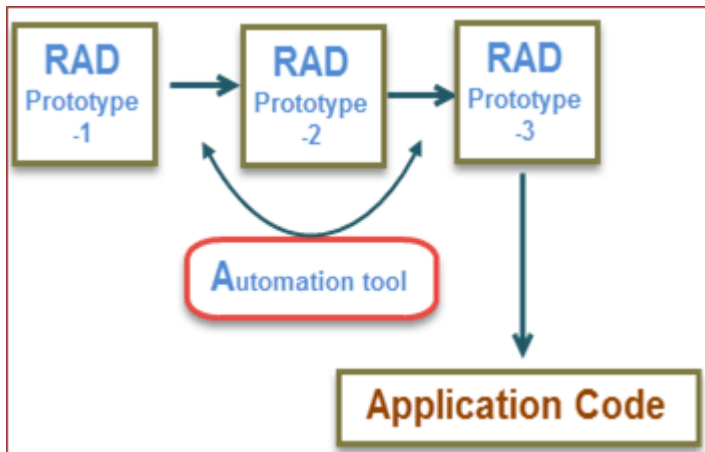
Proses RAD atau Rapid Application Development merupakan adopsi dari model waterfall; tujuannya adalah untuk mengembangkan perangkat lunak dalam waktu singkat. RAD mengikuti pendekatan iteratif.

Model SDLC RAD memiliki fase-fase berikut:

- Pemodelan Bisnis
- Pemodelan Data
- Pemodelan Proses
- Pembuatan Aplikasi
- Pengujian dan Perputaran



Model ini berfokus pada input-output, yaitu sumber dan tujuan informasi. Model ini menekankan pada penyelesaian proyek dalam bagian-bagian kecil; proyek yang lebih besar dibagi menjadi serangkaian proyek yang lebih kecil. Fitur utama model RAD adalah fokusnya pada penggunaan kembali templat, alat, proses, dan kode.



Model RAD dalam Rekayasa

Perangkat Lunak

Berbagai fase model RAD meliputi:

Fase-fase model RAD	Aktivitas yang dilakukan dalam Model RAD
Pemodelan Bisnis	Berdasarkan alur informasi dan distribusi antar berbagai saluran bisnis, produk ini dirancang.
Pemodelan Data	Informasi yang dikumpulkan dari pemodelan bisnis dimurnikan menjadi serangkaian objek data yang signifikan bagi bisnis.
Pemodelan Proses	Objek data yang dideklarasikan pada fase pemodelan data diubah untuk mencapai aliran informasi yang diperlukan untuk mengimplementasikan fungsi bisnis.
Pembuatan Aplikasi	Alat otomatis digunakan untuk membangun perangkat lunak, untuk mengubah model proses dan data menjadi prototipe.

Pengujian dan Perputaran	• Karena prototipe diuji secara individual selama setiap iterasi, waktu pengujian keseluruhan berkurang di RAD.
---------------------------------	---

Kapan menggunakan Metodologi RAD?

- Ketika suatu sistem perlu diproduksi dalam jangka waktu singkat (2-3 bulan)
- Ketika persyaratannya sudah diketahui
- Ketika pengguna akan terlibat sepanjang siklus hidup
- Ketika risiko teknis lebih rendah
- Ketika ada kebutuhan untuk membuat sistem yang dapat dimodularisasi dalam waktu 2-3 bulan.
- Ketika anggaran cukup tinggi untuk membayar desainer untuk pemodelan beserta biaya alat otomatis untuk pembuatan kode.

Keuntungan dan Kerugian Model SDLC RAD

Keuntungan	Kekurangan
• Fleksibel dan mudah beradaptasi terhadap perubahan.	• Tidak bisa digunakan untuk proyek-proyek kecil.
• Hal ini berguna ketika Anda perlu mengurangi risiko proyek secara keseluruhan.	• Tidak semua aplikasi kompatibel dengan RAD.
• Sistem ini mudah beradaptasi dan fleksibel terhadap perubahan.	• Jika risiko teknis tinggi, maka hal itu tidak cocok.
• Transfer hasil kerja akan lebih mudah jika menggunakan skrip, abstraksi tingkat tinggi, dan kode perantara.	• Jika pengembang tidak berkomitmen untuk mengirimkan perangkat lunak tepat waktu, proyek RAD dapat gagal.

• Berkat generator kode dan penggunaan kembali kode, terjadi pengurangan pengkodean manual.	• Pengurangan fitur karena pembatasan waktu (time boxing), di mana fitur-fitur tertentu ditunda ke versi selanjutnya untuk menyelesaikan rilis dalam waktu singkat.
• Karena sifatnya yang berupa pembuatan prototipe, kemungkinan terjadinya cacat lebih kecil.	• Skalabilitas yang berkurang terjadi karena aplikasi yang dikembangkan dengan RAD dimulai sebagai prototipe dan berkembang menjadi aplikasi yang sudah jadi.
• Setiap fase dalam RAD memberikan fungsionalitas prioritas tertinggi kepada klien.	• Kemajuan dan masalah yang terjadi sulit dilacak karena tidak ada dokumentasi yang menunjukkan apa yang telah dilakukan.
• Dengan jumlah personel yang lebih sedikit, produktivitas dapat ditingkatkan dalam waktu singkat.	• Membutuhkan desainer atau pengembang yang sangat terampil.