

Nama: Siti Rohmah

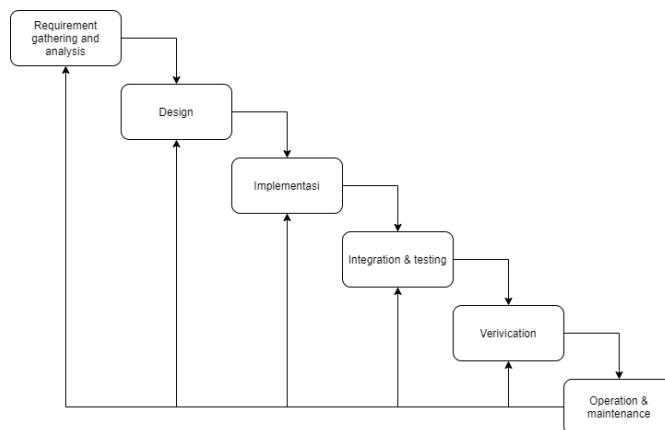
NPM: 222310054

Kelas: TI-22-PB

Tugas: IMK Model Pengembangan Software (SDLC)

1. Waterfall

Metode waterfall adalah metode kerja yang menekankan fase-fase yang berurutan dan sistematis. Disebut waterfall karena proses mengalir satu arah “ke bawah” seperti air terjun. Metode waterfall ini harus dilakukan secara berurutan sesuai dengan tahap yang ada.



Berikut adalah tahap-tahap pengembangan dalam metode waterfall.

- **Requirement gathering and analysis**
Mengumpulkan kebutuhan secara lengkap untuk dianalisis dan mendefinisikan kebutuhan apa saja yang harus dicapai oleh program. Informasi dapat diperoleh melalui wawancara, diskusi, atau survey.
- **Design**
Melakukan perancangan desain perangkat lunak sebagai perkiraan sebelum dibuatnya kode. Desain sistem dapat dibuat menggunakan Flowchart, Mind Map, atau Entity Relationship Diagram (ERD).
- **Implementasi**
Implementasi ini adalah tahap dimana seluruh desain yang sebelumnya sudah dibuat diubah menjadi kode-kode program. Kode yang dihasilkan masih berbentuk modul-modul yang harus digabungkan di tahap selanjutnya.
- **Integration & testing**
Di tahap ini dilakukan penggabungan modul-modul yang sudah dibuat sebelumnya dan melakukan pengujian untuk mengetahui apakah perangkat lunak yang dibuat telah sesuai dengan desain dan fungsinya atau tidak.
- **Verivication**
Di tahap ini, pengguna atau klien yang langsung melakukan pengujian pada sistem, apakah sistem telah sesuai dengan yang disetujui atau belum sesuai.
- **Operation & maintenance**
Tahap ini merupakan tahap terakhir dari model waterfall. Sistem yang sudah selesai dijalankan serta dilakukan pemeliharaan. Pemeliharaan berupa memperbaiki kesalahan yang tidak ditemukan pada langkah sebelumnya.

Setiap metode yang digunakan pasti memiliki kelebihan serta kekurangannya tersendiri. Berikut adalah kelebihan dari metode waterfall:

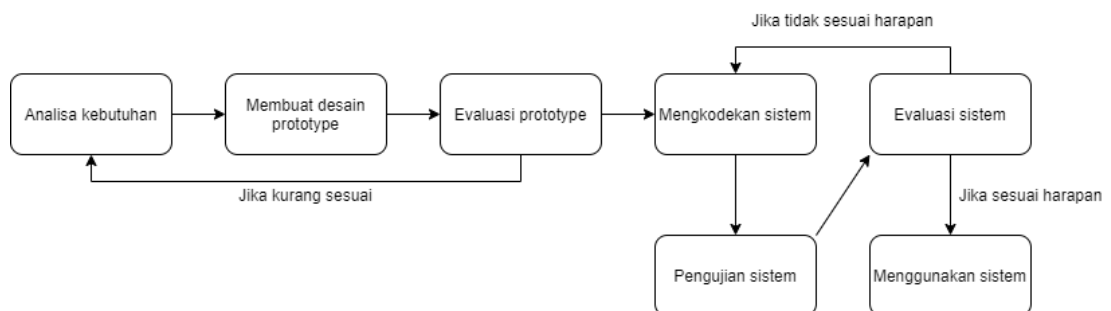
- Memiliki proses yang terurut, sehingga pengerjaan dapat terjadwal dengan baik dan mudah.
- Cocok untuk sistem dengan kompleksitas rendah (predictable).
- Setiap proses yang dilakukan tidak dapat saling tumpah tindih.

Berikut adalah kekurangan dari metode waterfall:

- Waktu pengerjaan relatif lebih lama, karena harus menunggu tahap sebelumnya selesai.
- Biaya yang dibutuhkan lebih mahal karena waktu pengembangan yang dibutuhkan lebih lama.
- Model waterfall ini kurang cocok untuk pengembangan proyek yang memiliki kompleksitas tinggi.

2. Prototype

Metode prototype adalah metode yang memungkinkan pengguna atau user memiliki gambaran awal tentang perangkat lunak yang akan dikembangkan, serta pengguna dapat melakukan pengujian di awal sebelum perangkat lunak dirilis. Metode ini bertujuan untuk mengembangkan model menjadi perangkat lunak yang final. Artinya sistem akan dikembangkan lebih cepat dan biaya yang dikeluarkan lebih rendah. Metode prototype ini memiliki tahap-tahap yang harus dilakukan dalam pengembangan perangkat lunak.



Berikut adalah tahap-tahap pengembangan perangkat lunak menggunakan metode prototype.

- Analisa kebutuhan
Pada tahap ini pengembang melakukan identifikasi perangkat lunak dan semua kebutuhan sistem yang akan dibuat.
- Membuat prototype
Membuat rancangan sementara yang berfokus pada alur program kepada pengguna.
- Evaluasi prototype
Evaluasi dilakukan untuk mengetahui apakah model prototype sudah sesuai dengan harapan.

- Mengkodekan sistem
Jika prototype disetujui maka akan diterjemahkan ke dalam bahasa pemrograman yang sesuai.
- Pengujian sistem
Setelah perangkat lunak sudah siap, perangkat lunak harus melewati pengujian. Pengujian ini biasanya dilakukan dengan White Box Testing, Black Box Testing, dan lain-lain.
- Evaluasi sistem
Pengguna melakukan evaluasi apakah perangkat lunak sudah sesuai dengan apa yang diharapkan atau tidak. Jika ya, lakukan tahap selanjutnya. Jika tidak, ulangi tahap mengkodekan sistem dan pengujian sistem.
- Menggunakan sistem
- Perangkat lunak yang telah diuji dan disetujui siap untuk digunakan.

Sebagai suatu metode yang sering digunakan, metode prototype pasti memiliki kelebihan dan kekurangan. Berikut adalah kelebihan dari metode prototype:

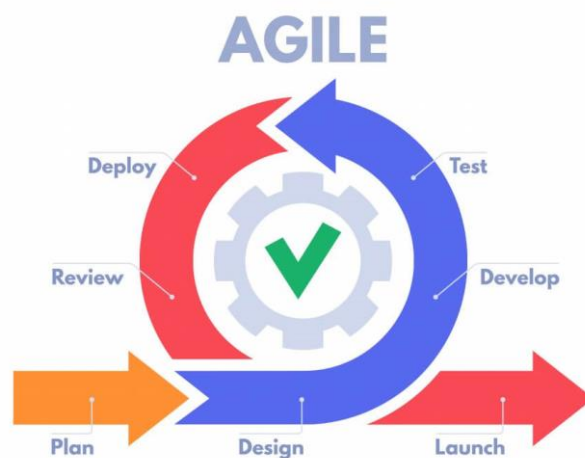
- Mempersingkat waktu pengembangan perangkat lunak.
- Penerapan fitur menjadi lebih mudah, karena pengembang mengetahui apa yang diharapkan.

Berikut adalah kekurangan dari metode prototype:

- Proses yang dilakukan untuk analisis dan perancangan terlalu singkat.
- Kurang fleksibel jika terjadi perubahan.

3. Agile

Metode agile adalah metode yang fleksibel di mana pengembangan dilakukan dalam jangka pendek. Namun diperlukan adaptasi yang cepat dari developer terhadap perubahan dalam bentuk apa pun.



Berikut merupakan tujuan dari agile, antara lain:

- High – value & working app system
Menghasilkan produk dengan kualitas yang baik, dan memiliki nilai jual yang tinggi.
- Iterative, incremental, evolutionary
Pengembangan dapat dilakukan secara iteratif, berulang-ulang, dan dapat mengalami perubahan jika diperlukan.
- Cost control & value – driven development
Pengembangan perangkat lunak dapat sesuai dengan kebutuhan pengguna dan tim dapat dengan cepat merespon kebutuhan, sehingga waktu dan biaya pembuatan dari perangkat lunak dapat dikendalikan.
- High – quality production
Kualitas dari perangkat lunak tetap terjaga, meskipun waktu dan biaya lebih sedikit.
- Flexible & risk management
Meminimalisir terjadinya kesalahan pada program ataupun produk sebelum dilakukannya proses deploy aplikasi.
- Collaboration
Kolaborasi ini dilakukan oleh setiap tim pengembang untuk mendiskusikan feedback yang diberikan oleh klien.
- Self – organizing, self – managing teams
Pengembang diberikan akses untuk manajemen sendiri urusan software development. Seorang manajer hanya bertugas sebagai penghubung antara pengembang dengan klien sehingga dapat mengurangi terjadinya miss communication.

Metode agile ini memiliki kelebihan dan kekurangan tersendiri. Berikut adalah kelebihan dari metode agile:

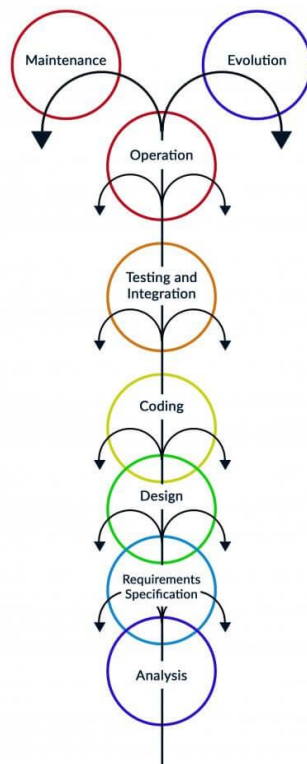
- Perubahan dapat dengan cepat ditangani.
- Proses pengembangan perangkat lunak membutuhkan waktu yang relatif cepat dan tidak memerlukan sumber daya yang besar.
- Klien dapat memberikan feedback kepada pengembang dalam proses pembuatan program.

Berikut adalah kekurangan dari metode agile:

- Metode ini kurang sesuai dengan tim yang besar (lebih dari 20 orang).
- Tim harus selalu siap, karena perubahan dapat terjadi kapan saja.
- Metode ini kurang cocok untuk tim yang berkomitmen untuk menyelesaikan proyek bersama-sama.

4. Fountain

Metode fountain adalah perbaikan dari metode waterfall, di mana jenis tahapan masih sama. Namun beberapa jenis tahapan boleh didahulukan atau dilewati, tetapi ada tahapan yang tidak bisa dilewati, contohnya seperti kamu memerlukan design sebelum melakukan implementasi, jika hal tersebut dilewati maka akan ada tumpang tindih.



Berikut adalah tahap-tahap pengembangan perangkat lunak menggunakan metode fountain.

- User requirement specification
Mencari tahu apa saja yang dibutuhkan oleh pengguna dalam perangkat lunak yang sedang dikembangkan.
- Software requirement specification
Penyesuaian perangkat lunak dari sisi pengguna.
- System design
Pembuatan desain sistem yang akan dibuat sebelum diimplementasikan.
- Program design
Pembuatan desain yang lebih sempurna dan hampir mendekati hasil akhir dari perangkat lunak.
- Implementation
Di tahap ini dilakukan implementasi sesuai dengan desain yang sudah dibuat di tahap sebelumnya.
- Program testing: unit

Dalam tahap ini dilakukan uji coba terhadap unit-unit yang dibutuhkan dalam perangkat lunak yang dikembangkan.

- Program testing: system
Di tahap ini dilakukan uji coba terhadap sistem dari perangkat lunak seutuhnya sebelum perangkat lunak digunakan.
- Program use
Dalam tahap ini dilakukan pengajaran kepada pengguna untuk menggunakan perangkat lunak yang telah dibuat.
- Software maintenance
Biasanya dalam tahap ini dilakukan perawatan terhadap perangkat lunak yang sudah dibuat, perawatan dapat berupa update sistem atau perbaikan kesalahan atau bugs yang ada.

Karena metode fountain ini adalah perbaikan dari metode waterfall, maka metode ini memiliki kelebihan dan kekurangan yang mirip dengan metode waterfall. Berikut adalah kelebihan dari metode fountain:

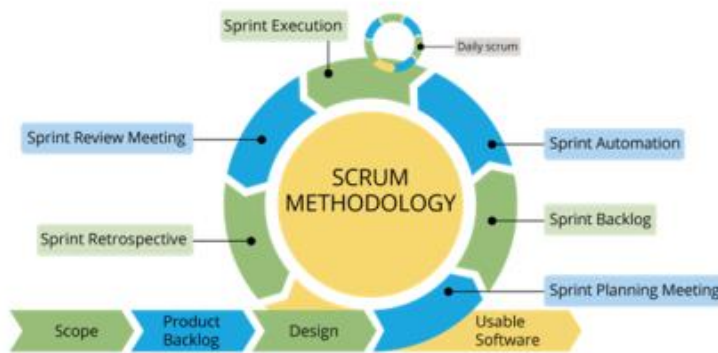
- Memiliki proses yang teratur, sehingga pengerjaan dapat terjadwal dengan baik dan mudah.
- Cocok untuk sistem dengan kompleksitas rendah (predictable).
- Dapat melewati atau mendahului beberapa tahapan .
- Setiap proses yang dilakukan tidak dapat saling tumpah tindih.

Berikut adalah kekurangan dari metode waterfall:

- Waktu pengerjaan relatif lebih lama, karena harus menunggu tahap sebelumnya selesai.
- Biaya yang dibutuhkan lebih mahal karena waktu pengembangan yang dibutuhkan lebih lama.
- Model fountain ini kurang cocok untuk pengembangan proyek yang memiliki kompleksitas tinggi.

5. Scrum Model

Scrum Pada dasarnya merupakan salah satu komponen dari metodologi pengembangan sistem Agile . Akhir-akhir ini scrum mulai marak di implemntasikan di perusahaan IT di Indonesia, dikarenakan maraknya perusahaan IT mengimplementasikan agile development. Scrum menguraikan proses untuk mengidentifikasi dan katalogisasi pekerjaan yang perlu dilakukan, memprioritaskan yang bekerja dengan berkomunikasi dengan pelanggan atau wakil pelanggan, dan pelaksanaan yang bekerja menggunakan rilis iterative dan memiliki tujuan utama untuk mendapatkan perkiraan berapa lama development akan dilakukan. Ada 3 elemen organisasi utama pada scrum yaitu product owner, Scrum master, dan the Scrum team.



- Product Owner mewakili bisnis, pelanggan atau pengguna dan memandu tim ke arah pengembangan produk yang tepat.
- Scrum Master dapat dianggap sebagai pemersatu bagi product owner dan scrum team (developer, QA, technical writer dll), membantu anggota tim menggunakan kerangka Scrum untuk menyelesaikan suatu project berdasarkan timeline yang ditentukan di awal.
- Scrum Team merupakan grup pengembang kecil biasanya terdiri dari 5-9 orang. Untuk proyek yang sangat besar, pekerjaan biasanya dibagi dan didelegasikan ke grup-grup kecil.

Scrum tepat digunakan saat kondisi:

- Keperluan berubah dengan cepat.
- Tim programmer sedikit, yaitu 5-9 orang.
- Pelanggan tidak terlalu paham dengan apa yang diinginkan.

Scrum memiliki prinsip yaitu:

- Ukuran tim yang kecil melancarkan komunikasi, mengurangi biaya, dan memberdayakan satu sama lain.
- Proses dapat beradaptasi terhadap perubahan teknis dan bisnis.
- Proses menghasilkan beberapa software increment.
- Pembangunan dan orang yang membangun dibagi dalam tim yang kecil.
- Dokumentasi dan pengujian terus menerus dilakukan setelah software dibangun.
- Proses scrum mampu menyatakan bahwa produk selesai kapanpun diperlukan

Kelebihan Scrum antara lain:

- Keperluan berubah dengan cepat.
- Tim berukuran kecil sehingga melancarkan komunikasi, mengurangi biaya dan memberdayakan satu sama lain.
- Pekerjaan terbagi-bagi sehingga dapat diselesaikan dengan cepat.
- Dokumentasi dan pengujian terus menerus dilakukan setelah software dibangun.
- Proses Scrum mampu menyatakan bahwa produk selesai kapanpun diperlukan

Kelemahan Scrum antara lain:

- Developer harus selalu siap dengan perubahan karena perubahan akan selalu diterima.

6. Iterative Model

Dalam Iterative model SDLC, proses iterative dimulai dengan implementasi sederhana dari komponen kecil dari software sampai dengan meningkatkan versi dari sebuah software dengan update-updateanya sehingga software siap digunakan ke user.

Di setiap Iterative nya, perubahan baik design maupun fungsi ditambahkan. Ide dasar di balik metode ini adalah untuk mengembangkan sistem melalui siklus berulang (iterative) dan dalam porsi kecil di setiap updatetanya.



Seperti model SDLC lainnya, Iterative model memiliki spesifikasi khusus di dalam industri software. Model ini paling sering digunakan dalam kondisi seperti:

- Requirement sistem dan design harus jelas dan mudah di pahami.
- Persyaratan Utama harus didefinisikan, namun nantinya akan ada request baru untuk penambahan fungsi pada saat sistem sedang berjalan.
- Teknologi yang sedang digunakan dalam pengembangan software bisa diganti apabila ada teknologi baru yang lebih bagus.
- Ada beberapa fitur berisiko tinggi dan tujuan yang mungkin berubah di masa depan.

Kelebihan dari Iterative Model SDLC:

- Beberapa fungsi dapat di kembangkan dengan cepat di awal pembuatan versi baru.
- hasil yang di peroleh secara berkala.
- Kemajuan sebuah sistem dapat di ukur.
- Development software mudah di rencanakan.
- Biaya yang dikeluarkan kecil apabila ingin merubah requirement.
- Testing dan debugging selama proses iterasi lebih mudah.
- Analisis resiko yang lebih baik.
- Mendukung perubahan requirement.
- Waktu operasional yang lebih singkat.
- Cocok untuk project besar

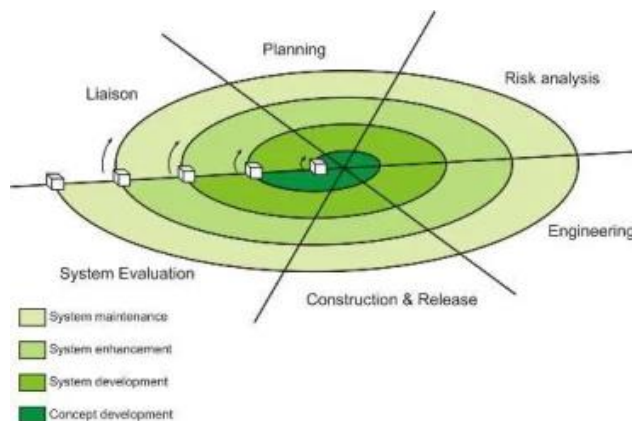
Kekurangan dari Iterative Model SDLC:

- Membutuhkan resource yang cukup banyak.
- Meski biaya perubahan rendah, tetapi sangat tidak cocok untuk mengubah persyaratan.
- Memerlukan Perhatian manajemen.
- Permasalahan sistem arsitektur dan desain mungkin akan timbul, karena tidak semua persyaratan di tentukan di awal pengembangan sistem.

- tidak cocok untuk project kecil.
- Kompleksitas manajemen.
- Membutuhkan tenaga ahli untuk analisis resiko yang timbul.

7. Spiral Model

Model Spiral SDLC adalah sebuah metode penggabungan antara Iterative Model dengan Waterfall Model. dengan penekanan yang tinggi pada analisis resiko yang akan di hadapi. Spiral model bertujuan untuk meningkatkan tingkat keberhasilan pada saat pengembangan suatu sistem.



Fase Spiral Model:

- **Identification**
Pada fase ini bertujuan untuk mengumpulkan kebutuhan bisnis di dasar spiral, Dalam spiral berikutnya disebut sebagai produk deawsa. Identifikasi persyaratan sistem, persyaratan subsistem, persyaratan unit dilakukan pada fase ini. Fase ini juga mencakup komunikasi antar sistem analis dengan klien.
- **Design**
Pada fase ini dimulai dengan desain konseptual di dasar spiral dan melibatkan desain arsitektur, desain logis dari modul, desain produk fisik dan desain akhir dalam spiral berikutnya.
- **Construct or Build**
Pada fase ini mengacu produksi produk perangkat lunak yang sebenarnya di setiap spiral.
- **Evaluation and Risk Analysis**
Pada fase ini mengidentifikasi, memperkirakan dan memantau kelayakan teknis dan risiko manajemen, seperti jadwal selip dan biaya lebih. Setelah pengujian sistem, akhir dari iterasi klien akan mengevaluasi produk yang sudah dibangun dan akan memberikan feedback.

Kelebihan dari Spiral Model:

- Perubahan kebutuhan dapat diakomodir.
- Persyaratan dapat diketahui lebih akurat.
- Pengguna dapat melihat sistem awal.

- Pembangunan dapat dibagi menjadi bagian-bagian yang lebih kecil dan bagian-bagian yang berisiko dapat dikembangkan sebelumnya yang membantu dalam manajemen risiko yang lebih baik

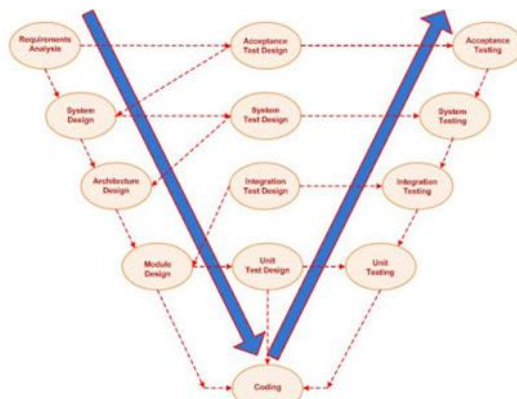
Kekurangan dari Spiral Model:

- Manajemen lebih kompleks.
- Akhir proyek mungkin tidak diketahui di awal.
- Tidak cocok untuk proyek-proyek berisiko kecil atau rendah dan bisa menjadi mahal untuk proyek-proyek kecil.
- Proses yang kompleks.
- Spiral mungkin berlangsung tanpa batas.

8. V Model

The V-Model adalah model SDLC dimana pelaksanaan proses yang terjadi secara berurutan dalam bentuk-V. Dikenal juga sebagai model Verifikasi dan Validasi.

The V-Model merupakan perluasan dari waterfall model dan didasarkan pada asosiasi dari fase pengujian untuk setiap tahap pengembangan yang sesuai. Ini berarti bahwa untuk setiap fase tunggal dalam siklus pengembangan, ada tahap pengujian terkait langsung. Ini adalah model yang sangat disiplin dan tahap berikutnya dimulai setelah selesainya tahap sebelumnya.



Ada beberapa tahapan verifikasi di V-Model, masing-masing dijelaskan secara rinci di bawah:

- **Business Requirement Analysis**
Ini adalah tahap pertama dalam siklus pengembangan di mana persyaratan produk dipahami dari perspektif pelanggan. Fase ini melibatkan komunikasi rinci dengan pelanggan untuk memahami harapan dan kebutuhan yang tepat. Ini merupakan kegiatan yang sangat penting dan perlu dikelola dengan baik, karena sebagian besar pelanggan tidak yakin tentang apa yang sebenarnya mereka butuhkan. Acceptance test desain dilakukan pada tahap ini sebagai kebutuhan bisnis dapat digunakan sebagai masukan untuk pengujian penerimaan.
- **System Design**
Setelah Anda memiliki persyaratan produk yang jelas dan rinci, sekarang saatnya untuk merancang sistem yang lengkap. Desain sistem akan memiliki pemahaman dan merinci hardware lengkap dan setup komunikasi untuk produk dalam pengembangan. Rencana pengujian sistem dikembangkan berdasarkan desain sistem.

Melakukan hal ini pada tahap awal membuat lebih banyak waktu untuk pelaksanaan tes yang sebenarnya nanti.

- **Architectural Design**
spesifikasi arsitektur dipahami dan dirancang dalam fase ini. Biasanya lebih dari satu pendekatan teknis diusulkan dan berdasarkan kelayakan teknis dan finansial keputusan akhir diambil. Desain sistem dipecah lebih jauh ke dalam modul mengambil fungsi yang berbeda. Hal ini juga disebut sebagai “Desain Tingkat Tinggi”.
- **Module Design**
Pada fase ini, desain internal rinci untuk semua modul sistem yang ditentukan, disebut “Desain Tingkat Rendah”. Penting bahwa desain tersebut kompatibel dengan modul lain dalam arsitektur sistem dan sistem eksternal lainnya.
- **Coding Phase**
Bahasa pemrograman yang paling cocok ditentukan berdasarkan sistem dan persyaratan arsitektur. pengkodean dilakukan berdasarkan pedoman coding dan standar. Kode berjalan melalui berbagai ulasan kode dan dioptimalkan untuk kinerja terbaik sebelum final membangun diperiksa ke dalam repositori

Kelebihan dari V-Model SDLC:

- Ini adalah model yang sangat-disiplin dan Tahapan selesai satu per satu.
- Bekerja dengan baik untuk proyek-proyek yang lebih kecil dimana persyaratan dipahami dengan baik.
- Sederhana dan mudah dimengerti dan digunakan.
- Mudah dikelola karena setiap fase memiliki spesifik kiriman dan proses review.

Kekurangan dari V-Model SDLC:

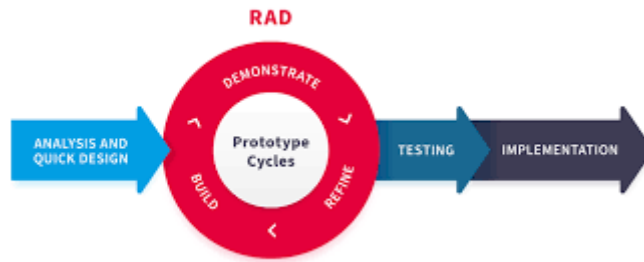
- Berisiko tinggi dan ketidakpastian.
- Tidak cocok untuk proyek-proyek yang kompleks dan berorientasi objek.
- Tidak cocok untuk proyek-proyek dimana persyaratan berisiko tinggi.
- Tidak cocok untuk proyek-proyek yang lama dan berkelanjutan.
- Setelah aplikasi dalam tahap pengujian, sulit untuk kembali dan mengubah fungsionalitas.

9. RAD (Rapid Application Development) Model

Dalam model RAD (Rapid Application Development), modul fungsional dikembangkan secara paralel sebagai prototip dan terintegrasi untuk membuat produk yang lengkap untuk pengiriman produk yang lebih cepat. Dikarenakan tidak ada rincian planning yang detail, maka memudahkan untuk melakukan perubahan pada saat development berjalan.

RAD Model Design

Model RAD mendistribusikan tahap analisis, perancangan, pembuatan dan pengujian ke dalam rangkaian siklus pengembangan jangka pendek yang singkat.



Berikut adalah fase-fase dari RAD:

- **Business Modeling (Bisnis Model)**
fase ini untuk perancangan dasar dari pengembangan produk berdasarkan informasi dan distribusi informasi antar saluran bisnis. Analisis bisnis yang lengkap dilakukan untuk menemukan informasi penting untuk bisnis, bagaimana hal itu dapat diperoleh, bagaimana dan kapan informasi diproses dan faktor apa yang mendorong arus informasi yang berhasil.
- **Data Modeling (Data Model)**
Fase ini untuk menganalisa informasi yang sudah dikumpulkan dari fase Business Modeling. semua kumpulan data diidentifikasi dan didefinisikan secara rinci untuk mencari model bisnis yang tepat.
- **Process Modeling (Proses Pemodelan)**
Fase ini untuk menetapkan arus informasi bisnis yang diperlukan untuk mencapai tujuan bisnis yang spesifik sesuai model bisnis. perubahan atau penyempurnaan pada kumpulan objek data didefinisikan dalam fase ini. Deskripsi proses untuk menambahkan, menghapus, mengambil atau memodifikasi objek data diberikan.
- **Application Generation (Generasi Aplikasi)**
Fase ini untuk Sistem yang sebenarnya dibangun dan pengkodean dilakukan dengan menggunakan automatic tools i untuk mengubah model proses dan data menjadi prototype yang aktual.
- **Testing and Turnover**
fase ini untuk pengujian keseluruhan sistem yang dibangun semua komponen perlu diuji secara menyeluruh dengan cakupan uji yang lengkap. Dengan pengujian yang lengkap dapat mengurangi risiko cacat sistem.

Kelebihan RAD (Rapid Application Development):

- Mudah mengakomodasi perubahan sistem.
- Progress development bisa di ukur.
- Waktu iterasi bisa di diperpendek menggunakan RAD Tools.

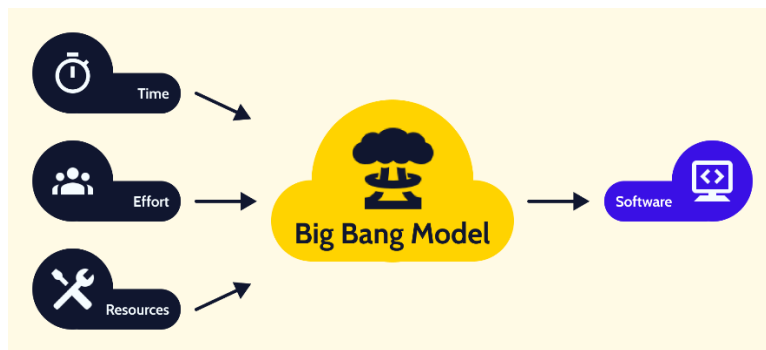
- Mengurangi waktu development.
- Mudah dalam menentukan dasar sistem.
- Mempermudah feedback customer.
- Cocok untuk proyek yang membutuhkan waktu pengembangan yang lebih pendek.
- Cocok untuk sistem yang berbasis komponen dan terukur.

Kekurangan RAD (Rapid Application Development):

- Ketergantungan pada anggota bisnis tim untuk mengidentifikasi persyaratan bisnis.
- Hanya sistem yang bisa di modularized yang bisa dibangun menggunakan RAD.
- Membutuhkan developer / designer yang berpengalaman.
- Ketergantungan pada keterampilan model.
- Kompleksitas manajemen.
- Tidak dapat diterapkan pada proyek yang kecil / murah

10. Big Bang Model

Pengertian dari SDLC Big Bang Model adalah Dimana kita tidak mengikuti proses tertentu. Perkembangan hanya dimulai dengan uang dan usaha yang dibutuhkan sebagai masukan, dan hasilnya adalah perangkat lunak yang dikembangkan yang mungkin atau mungkin tidak sesuai dengan kebutuhan pelanggan. Model Big Bang ini tidak mengikuti dan hanya ada sedikit perencanaan yang diperlukan. Bahkan pelanggan pun tidak yakin dengan apa yang sebenarnya dia inginkan dan persyaratannya diimplementasikan dengan cepat tanpa banyak analisis.



Spesifikasi Big Bang Model SDLC

Model Big Bang terdiri dari memfokuskan semua sumber daya yang mungkin dalam pengembangan perangkat lunak dan pembuatan code / coding, dengan perencanaan yang sangat sedikit atau tidak sama sekali. Requirement yang dibutuhkan terkadang datang pada saat pembuatan code. Setiap perubahan yang diperlukan mungkin atau mungkin tidak perlu mengubah perangkat lunak yang lengkap.

Keuntungan Big Bang Model antara lain:

- Model yang sangat sederhana.
- Sedikit atau tidak ada perencanaan yang dibutuhkan.
- Mudah dikelola.
- Sangat sedikit sumber daya yang dibutuhkan.
- Memberikan fleksibilitas kepada pengembang.

- Bagus untuk developer yang ingin belajar atau developer pendatang baru.

Kekurangan Big Bang Model antara lain:

- Beresiko tinggi dan kepastian dari requirement yang tidak jelas.
- Tidak cocok untuk project skala besar dan berorientasi objek.
- Model yang buruk untuk proyek yang panjang dan sedang berlangsung.
- Bisa berubah menjadi sangat mahal jika persyaratan disalahpahami