Diskusi.3

Lakukan: Kirim balasan: 1

Jatuh tempo: Senin, 27 Oktober 2025, 23:59

menampilkan balasan dalam bentuk bertingkat

Setelan

Sudah mencapai batas waktu untuk mengirim ke forum ini sehingga Anda tidak dapat lagi mengirim ke forum ini.

Diskusi.3
Rabu, 28 Mei 2025, 10:24

Sebuah startup bernama FoodFast ingin mengembangkan aplikasi pemesanan makanan online yang menghubungkan pelanggan dengan restoran terdekat. Aplikasi ini harus memiliki fitur utama seperti:

1. Pendaftaran pengguna dan restoran
2. Pencarian restoran berdasarkan lokasi dan kategori
3. Pemrosesan pesanan dan pembayaran online
4. Pelacakan status pesanan secara real-time
5. Sistem ulasan dan rating untuk restoran

Namun, startup ini menghadapi beberapa tantangan:

- Mereka ingin aplikasi ini dikembangkan dengan cepat agar segera dapat digunakan pelanggan.
- Beberapa fitur mungkin perlu diuji oleh pengguna sebelum dirilis secara penuh.
- Perubahan fitur mungkin akan sering terjadi berdasarkan umpan balik pengguna.
- Startup ingin meminimalkan risiko kegagalan proyek, karena ini adalah produk utama mereka di pasar digital.

Pertanyaan Diskusi:

1. Di antara model SDLC (Waterfall, RAD, Spiral, dan Prototipe), mana yang paling cocok digunakan untuk proyek ini? Jelaskan alasan pemilihannya berdasarkan karakteristik proyek.
2. Bagaimana model yang Anda pilih dapat mengatasi kemungkinan perubahan fitur yang sering terjadi?
3. Apa kelebihan dan kekurangan model SDLC yang Anda pilih dibandingkan dengan model lainnya dalam konteks proyek ini?

Gunakan BMP STSI4202 Rekayasa Perangkat Lunak sebagai acuan dalam menjawab forum diskusi ini.

Selamat berdiskusi!

Catatan: Saya sarankan, teman-teman mahasiswa menjawab langsung pada tempat yang disediakan, TIDAK mengupload jawaban berupa file, termasuk TIDAK mengupload jawaban dengan GAMBAR ya. Terima kasih!

Tautan permanen

Hide sidebars

Re: Diskusi.3

oleh RISKI SEPTIADI MURTI WAHYU 052218499 - Senin, 20 Oktober 2025, 12:25

Model Prototipe adalah model SDLC yang sangat sesuai digunakan oleh startup FoodFast dalam pengembangan aplikasi pesan makanan online.

1. Alasan Memilih Model Prototipe
Model ini cocok untuk proyek FoodFast karena:
a. Kebutuhan pengguna belum jelas dan bisa berubah karena masukan dari pengguna.
b. Proses pengembangan harus cepat agar aplikasi bisa segera dikeluarkan ke pasar.
c. Pada tahap awal, pengguna dan pengembang bisa saling berinteraksi, sehingga hasil akhir bisa cocok dengan kebutuhan pengguna.
d. Model ini membantu mengurangi risiko gagal karena masalah bisa terdeteksi lebih awal melalui versi prototipe.
Dalam model ini, pengembang membuat versi awal aplikasi dengan fitur sederhana seperti daftar akun, cari restoran, dan pesan makanan. Setelah itu, pengguna memberikan masukan dan usulan yang digunakan untuk memperbaiki atau menambah fitur hingga versi terakhir aplikasi siap dirilis.

2. Cara Model Prototipe Menangani Perubahan Fitur
Model ini sangat fleksibel terhadap perubahan. Setiap kali ada usulan atau perubahan dari pengguna, pengembang bisa memperbarui prototipe tanpa harus memulai dari awal. Dengan demikian:
a. Perubahan fitur bisa diterapkan bertahap.
b. Aplikasi terus berkembang sesuai kebutuhan nyata pengguna.
c. Pengembang dan pengguna bisa terus mengevaluasi hingga aplikasi benar-benar sesuai dengan harapan.

3. Kelebihan dan Kekurangan Model Prototipe
Kelebihan:
1) Cepat merespons perubahan kebutuhan pengguna.
2) Masukan langsung dari pengguna membuat aplikasi lebih baik dan lebih sesuai.
3) Risiko kegagalan berkurang karena pengguna terlibat dalam pengujian sejak awal.
4) Cocok untuk produk inovatif dan dinamis seperti aplikasi startup.
Kekurangan:
1) Biaya bisa meningkat jika terlalu banyak revisi.
2) Dokumen awal mungkin kurang lengkap karena fokus pada pengujian dan masukan cepat.
3) Jika tidak diatur dengan baik, revisi bisa terus berlanjut dan menyebabkan penundaan proyek.

Perbandingan dengan Model Lain:
1. Waterfall: Kurang cocok karena model ini kaku dan sulit untuk menyesuaikan perubahan.
2. RAD (Rapid Application Development): Cepat dan efektif, tetapi membutuhkan tim besar dan sumber daya yang banyak.
3. Spiral: Cocok untuk proyek besar dengan risiko tinggi, tapi membutuhkan modal dan tenaga yang besar, yang tidak cocok untuk startup.

Kesimpulan:
Model Prototipe adalah pilihan terbaik untuk FoodFast karena mendukung pengembangan cepat, fleksibel terhadap perubahan, dan melibatkan pengguna secara aktif dalam proses pengembangan. Dengan model ini, startup bisa meluncurkan aplikasi lebih cepat sekaligus terus meningkatkan kualitasnya berdasarkan pengalaman pengguna.

Sumber: BMP STSI4202 Rekayasa Perangkat Lunak

Tautan permanen Tampilkan induk

Re: Diskusi.3

oleh AJAY SUPRIADI 04001670 - Rabu, 22 Oktober 2025, 16:10

Terima kasih atas penjelasan Anda, analisis terhadap model Prototipe sudah tepat mengingat kebutuhan FoodFast yang dinamis.

Tautan permanen Tampilkan induk

Hide sidebars

Re: Diskusi.3

oleh 054276582 MUHAMMAD RIFKI ARSAH - Senin, 20 Oktober 2025, 21:23

1.Alasan pemilihan berdasarkan karakteristik proyek
- Model Prototipe: Cocok jika kebutuhan pengguna belum sepenuhnya jelas dan perlu divalidasi melalui interaksi langsung dengan model awal. Model ini menghasilkan prototipe fungsional yang dapat diuji oleh pengguna, sehingga umpan balik dapat segera diberikan untuk menyempurnakan fitur.
- Model Spiral: Ideal untuk proyek yang kompleks dan berisiko tinggi dengan persyaratan yang tidak jelas atau mungkin berubah. Model ini mengintegrasikan iterasi, prototipe, dan analisis risiko di setiap siklusnya, sehingga masalah atau perubahan dapat ditangani di awal proses pengembangan.

2.Mengatasi perubahan fitur yang sering terjadi
- Model Prototipe: Model ini secara inheren menangani perubahan karena tujuannya adalah untuk memvalidasi persyaratan melalui interaksi. Umpan balik dari pengujian prototipe dapat langsung digunakan untuk memodifikasi atau menambahkan fitur sebelum pengembangan penuh dilakukan.
- Model Spiral: Model ini dirancang untuk menangani perubahan. Setiap iterasi (putaran) melibatkan fase analisis risiko dan umpan balik. Jika ada perubahan fitur, model Spiral dapat mengintegrasikannya ke dalam siklus berikutnya tanpa menimbulkan dampak besar pada jadwal atau biaya secara keseluruhan.

3.kelebihan dan kekurangan model

a)Model Prototipe

- kelebihan
Fleksibilitas: Mampu mengakomodasi perubahan fitur secara efektif melalui umpan balik pengguna langsung.
- Umpan Balik Dini: Pengguna dapat menguji dan memberikan masukan lebih awal, memastikan produk sesuai dengan harapan.
- kekurangan
-Potensi Ketidakjelasan Akhir: Jika tidak dikelola dengan baik, fokus pada prototipe bisa mengabaikan kebutuhan integrasi dan kinerja akhir.

b)model spiral
- kelebihan
-Manajemen Risiko: Sangat baik untuk proyek dengan risiko tinggi atau ketidakpastian, memungkinkan identifikasi dan penyelesaian masalah di awal.
- Adaptif: Dapat beradaptasi dengan perubahan persyaratan karena siklusnya dapat diulang.

- kekurangan
-Biaya Lebih Tinggi: Proses iteratif dan analisis risiko yang terperinci membuatnya lebih mahal dan kompleks untuk dikelola dibandingkan model lain.
-Membutuhkan Keahlian: Analisis risiko yang efektif membutuhkan tim yang terampil dan berpengalaman.

Sumber:

https://www.dicoding.com/blog/metode-sdlc/

https://existek.com/blog/sdlc-models/

Tautan permanen Tampilkan induk

Re: Diskusi.3

oleh AJAY SUPRIADI 04001670 - Rabu, 22 Oktober 2025, 16:13

Anda telah memahami inti persoalan dengan baik. Model Prototipe memang fleksibel untuk perubahan fitur, Gunakan BMP STSI4202 Rekayasa Perangkat Lunak sebagai acuan dalam menjawab forum diskusi ini.

Tautan permanen Tampilkan induk

Hide sidebars

Re: Diskusi.3

oleh 054416248 PANDU WIJAYA - Senin, 20 Oktober 2025, 21:28

1. Model SDLC yang Paling Cocok: Model Spiral

Dari keempat model yang disebutkan (Waterfall, RAD, Spiral, dan Prototipe), model Spiral adalah yang paling cocok untuk proyek pengembangan aplikasi FoodFast.

•Alasan Pemilihan:
-Model Spiral menggabungkan elemen dari Prototyping dan Waterfall, serta menekankan pada analisis risiko dan iterasi pengembangan.
-Cocok untuk proyek besar dan kompleks dengan tingkat ketidakpastian tinggi, seperti aplikasi startup yang fiturnya masih dapat berubah berdasarkan umpan balik pengguna.
-Model ini memungkinkan tim untuk mengembangkan versi awal (prototype) dari sistem, menguji langsung ke pengguna, dan kemudian meningkatkan versi berikutnya berdasarkan hasil uji coba tersebut.

•Kesesuaian dengan Karakteristik Proyek FoodFast:
-Karakteristik Proyek Kesesuaian dengan Model Spiral
-Pengembangan cepat Setiap iterasi menghasilkan versi sistem yang dapat diuji pengguna.
-Fitur perlu diuji oleh pengguna Spiral memungkinkan pembuatan prototype pada tiap siklus.
-Fitur sering berubah Perubahan dapat dimasukkan ke iterasi berikut tanpa mengguncang keseluruhan sistem.
-Risiko proyek tinggi Spiral memiliki tahap analisis risiko yang mendalam di setiap siklus.

2. Cara Model Spiral Mengatasi Perubahan Fitur yang Sering Terjadi

•Model Spiral dibangun atas siklus berulang (iteratif), yang terdiri dari empat fase utama dalam setiap putaran:
1. Perencanaan dan penentuan tujuan
→ Menentukan tujuan sistem dan fitur prioritas yang akan dikembangkan dalam iterasi tersebut.
2. Analisis risiko dan perancangan prototipe
→ Tim mengidentifikasi risiko (teknis, waktu, biaya, kebutuhan pengguna) dan membuat prototype untuk menguji asumsi.
3. Pengembangan dan pengujian produk
→ Produk dikembangkan dan diuji bersama pengguna untuk mendapatkan umpan balik nyata.
4. Evaluasi dan perencanaan iterasi berikutnya
→ Berdasarkan umpan balik pengguna, fitur dapat diubah, diperbaiki, atau ditambahkan di siklus berikutnya.

•Dengan siklus seperti ini:
-Perubahan fitur tidak merusak sistem yang sudah ada karena perubahan diakomodasi pada fase iterasi berikutnya.
-Startup dapat menguji konsep lebih cepat, sambil tetap menjaga kualitas dan arah pengembangan secara terkontrol.

3. Kelebihan dan Kekurangan Model Spiral Dibandingkan Model Lain

Aspek Model Spiral Waterfall RAD (Rapid Application Development) Prototipe

•Spiral
Kelebihan : Fleksibel terhadap perubahan, fokus pada manajemen risiko, hasil tiap iterasi bisa diuji.
Kekurangan : Membutuhkan waktu dan biaya lebih, serta tim yang berpengalaman.

•Waterfall
Kelebihan : Terstruktur dan mudah dipahami.
Kekurangan : Tidak fleksibel terhadap perubahan kebutuhan.

•RAD
Kelebihan : Pengembangan cepat dengan pendekatan component-based.
Kekurangan : Kurang efektif untuk proyek kompleks dan berisiko tinggi.

•Prototipe
Kelebihan : Memberi gambaran cepat sistem ke pengguna. Kekurangan : Kurang memperhatikan risiko dan dokumentasi.

•Kesimpulan
Model SDLC yang paling tepat untuk proyek FoodFast adalah Model Spiral, karena:
-Dapat menangani perubahan fitur secara fleksibel.
-Menyediakan mekanisme pengujian dan evaluasi terus-menerus terhadap pengguna.
-Meminimalkan risiko kegagalan proyek melalui analisis risiko pada setiap iterasi.
-Mendukung pengembangan bertahap dan cepat sesuai kebutuhan startup digital.

Tautan permanen Tampilkan induk

Re: Diskusi.3
oleh AJAY SUPRIADI 04001670 - Rabu, 22 Oktober 2025, 16:16

Penjelasan Anda sederhana namun efektif, sangat mudah dicerna.
Gunakan BMP STSI4202 Rekayasa Perangkat Lunak sebagai acuan dalam menjawab forum diskusi ini.

Tautan permanen Tampilkan induk

Re: Diskusi.3
oleh MOCH RISWAN LUTFIN ANFA 051141089 - Senin, 20 Oktober 2025, 23:37

Nama: Moch Riswan lutfin Anfa
Nim: 051141089

Izin menjawab pertanyaan dari diskusi diatas, trimakasih

Berdasarkan karakteristik proyek aplikasi pemesanan makanan FoodFast dan tantangan yang dihadapi, model SDLC yang paling cocok adalah Prototipe atau, yang lebih ideal dalam konteks pengembangan cepat dan adaptif, RAD (Rapid Application Development), yang sangat menekankan pada pembuatan prototipe.
Namun, mengacu pada kebutuhan spesifik untuk pengembangan cepat, pengujian fitur, dan penanganan perubahan yang sering, model Prototipe sering kali menjadi fondasi terbaik, dan seringkali diimplementasikan bersama prinsip-prinsip iteratif dari RAD/Agile.
Berikut adalah penjelasan dan diskusi berdasarkan acuan rekayasa perangkat lunak:
1. Model SDLC yang Paling Cocok: Model Prototipe
Model SDLC yang paling cocok untuk proyek aplikasi FoodFast adalah Model Prototipe.
Alasan Pemilihan Berdasarkan Karakteristik Proyek:
- Karakteristik Proyek FoodFast:
A. Fitur utama jelas (Pendaftaran, Pencarian, Pembayaran, Pelacakan, Ulasan).

Tantangan yang Dihadapi:
Ingin dikembangkan dengan cepat. (Tantangan 1).
- Dukungan dari model prototipe:
Fase pengembangan inti (coding) bisa dimulai lebih cepat setelah prototipe awal disetujui.

B. Perlu umpan balik dan pengujian untuk fitur-fitur baru (misalnya, desain UI/UX, mekanisme pelacakan).
- Tantangan yang Dihadapi:
Beberapa fitur perlu diuji oleh pengguna sebelum dirilis secara penuh. (Tantangan 2).
- Dukungan dari model prototipe:
Prototipe memungkinkan pengguna mencoba fitur kunci lebih awal untuk memberikan umpan balik nyata (Validasi Awal).

C. Persaingan pasar digital yang ketat.
- Tantangan yang Dihadapi:
Perubahan fitur mungkin akan sering terjadi berdasarkan umpan balik pengguna. (Tantangan 3).
- Dukungan dari model prototipe:
Sifat iteratif model ini memungkinkan perubahan cepat pada prototipe tanpa harus menunggu siklus panjang (seperti pada Waterfall).

D. Produk utama startup.
- Tantangan yang Dihadapi:
Ingin meminimalkan risiko kegagalan proyek. (Tantangan 4)
- Dukungan dari model prototipe:
Risiko kesalahan dalam memahami kebutuhan berkurang drastis karena pelanggan/stakeholder terlibat sejak awal. Produk akhir lebih sesuai dengan harapan pasar.

• Proses Inti Model Prototipe:
Model ini dimulai dengan mengumpulkan kebutuhan dasar dan cepat membuat prototipe awal yang berfungsi. Prototipe ini kemudian diuji oleh pelanggan/calon pengguna. Umpan balik yang diperoleh digunakan untuk memperbaiki dan menyempurnakan prototipe hingga pelanggan puas. Setelah disepakati, prototipe tersebut dijadikan dasar untuk pengembangan sistem final.

2. Cara Model Prototipe Mengatasi Perubahan Fitur
Model Prototipe sangat efektif dalam mengatasi perubahan fitur yang sering terjadi (Tantangan 3) melalui proses Iteratif dan Umpan Balik Cepat:
• Validasi Kebutuhan Awal: Prototipe awal, meskipun sederhana, berfungsi sebagai representasi visual dan fungsional dari fitur. Pengguna dapat melihat dan berinteraksi dengannya.
• Mekanisme Umpan Balik Terstruktur: Startup dapat secara formal mengumpulkan feedback pengguna pada setiap iterasi prototipe (misalnya, fitur Pelacakan Pesanan versi 1.0).
• Biaya Perubahan Rendah: Karena perubahan dilakukan pada model (prototipe) dan bukan pada sistem yang sudah dikembangkan secara penuh, biaya dan waktu untuk mengubah desain atau fungsi menjadi jauh lebih rendah dibandingkan model Waterfall.
• Konvergensi Cepat: Iterasi berulang (buat /rightarrow uji /rightarrow perbaiki) memastikan bahwa prototipe berkembang semakin mendekati kebutuhan riil pengguna, sehingga perubahan fitur yang fundamental akan berhenti lebih cepat, dan tim dapat beralih ke pengembangan final.
Kesimpulan: Model Prototipe mengakomodasi perubahan sebagai bagian alami dari proses, bukan sebagai penyimpangan.

3. Kelebihan dan Kekurangan Model Prototipe
• Kelebihan:
A. Pengurangan Risiko, kebutuhan pengguna diverifikasi lebih awal. Jauh lebih baik dari Waterfall, yang baru tahu kesalahannya di fase pengujian akhir. Lebih baik dari Spiral dalam hal kecepatan karena fokusnya lebih sempit pada fungsionalitas.
B. Kepuasan Pelanggan, pelanggan terlibat dan merasakan produk sejak dini. RAD juga unggul di sini, tetapi Prototipe

menekankan validasi fitur spesifik. Waterfall memiliki kepuasan pelanggan yang rendah karena output akhir mungkin tidak sesuai harapan.
C. Penanganan Perubahan, perubahan dapat diakomodasi dengan mudah dan cepat. Jauh lebih adaptif daripada Waterfall. Mirip dengan RAD dan Spiral dalam fleksibilitas perubahan.

• Kekurangan
A. Potensi Scope Creep, pelanggan mungkin terus meminta fitur baru setelah melihat prototipe, sehingga proyek tidak selesai-selesai. Risiko ini juga ada pada model iteratif lainnya, tetapi lebih rentan di Prototipe jika tidak dikelola dengan baik.
B. Kualitas Implementasi Akhir, fokus pada kecepatan pembuatan prototipe bisa mengabaikan kualitas desain, arsitektur, atau dokumentasi jangka panjang. Waterfall dan Spiral unggul dalam aspek dokumentasi dan arsitektur yang terstruktur.
C."Prototipe Jadi Produk", prototipe yang terburu-buru bisa digunakan sebagai sistem akhir, padahal arsitekturnya belum matang. Jika tim FoodFast tidak disiplin, mereka berisiko memiliki fondasi aplikasi yang lemah.

Kesimpulan Pilihan:
Model Prototipe (atau RAD) adalah pilihan yang paling strategis karena aplikasi FoodFast membutuhkan pengembangan cepat dan validasi pasar yang intensif di awal proyek. Meminimalkan risiko kegagalan di pasar digital (Tantangan 4) dan mengakomodasi umpan balik yang cepat (Tantangan 3) adalah kunci sukses, yang ditawarkan oleh model ini, meskipun tim harus berhati-hati dalam mengelola scope creep dan memastikan kualitas arsitektur saat membangun sistem final.

Tautan permanen Tampilkan induk

Re: Diskusi.3
oleh AJAY SUPRIADI 04001670 - Rabu, 22 Oktober 2025, 16:39

Penjelasan Anda cukup lengkap, hanya perlu menambahkan sedikit referensi teoretis
unakan BMP STSI4202 Rekayasa Perangkat Lunak sebagai acuan dalam menjawab forum diskusi ini

Tautan permanen Tampilkan induk

Re: Diskusi.3
oleh RATIH NIKEN PRATIWI RAMADHANI 053714635 - Selasa, 21 Oktober 2025, 02:59

Izinkan saya menjawab soal diskusi 3 ini, jika ada kesalahan mohon dikoreksi.

Terima Kasih.

rekayasa perangkat lunak diskusi 3 ratih.docx

Tautan permanen Tampilkan induk

Re: Diskusi.3
oleh AJAY SUPRIADI 04001670 - Rabu, 22 Oktober 2025, 16:40

Saya sarankan, teman-teman mahasiswa menjawab langsung pada tempat yang disediakan, TIDAK mengupload jawaban berupa file, termasuk TIDAK mengupload jawaban dengan GAMBAR ya. Terima kasih!

Tautan permanen Tampilkan induk

Re: Diskusi.3
oleh SUPRIYADI 050127551 - Selasa, 21 Oktober 2025, 16:27

Ijin Menjawab Diskusi 3 Pemilihan Model SDLC untuk Proyek FoodFast

Hide sidebars

1. Menurut saya, model Spiral paling cocok digunakan untuk proyek FoodFast karena mampu menyeimbangkan kecepatan pengembangan, fleksibilitas terhadap perubahan, dan manajemen risiko yang baik. Model ini menggabungkan kelebihan Waterfall yang sistematis dan Prototyping yang iteratif, sehingga sesuai dengan kebutuhan startup yang ingin segera merilis aplikasi namun tetap adaptif terhadap umpan balik pengguna.

2. Model Spiral dapat mengatasi perubahan fitur melalui pendekatan iteratif dan incremental, di mana setiap siklus pengembangan mencakup tahapan perencanaan, analisis risiko, pengembangan, pengujian, dan evaluasi pengguna. Dengan cara ini, pengembang bisa menyesuaikan fitur di setiap tahap berdasarkan hasil uji coba dan masukan pengguna tanpa harus memulai dari awal.

3. Kelebihan model Spiral adalah fleksibilitas tinggi, mampu mengidentifikasi risiko lebih awal, serta cocok untuk proyek kompleks seperti FoodFast. Kekurangannya, model ini membutuhkan waktu dan biaya lebih besar serta tim yang berpengalaman. Dibandingkan dengan model lain seperti Waterfall, RAD, atau Prototipe, model Spiral lebih unggul dalam menghadapi dinamika perubahan dan menjaga kualitas sistem.

Sumber Referansi:
BMP STSI4202 – Rekayasa Perangkat Lunak, Modul 2, Universitas Terbuka; Pressman, 2015
Sommerville, 2016; BMP STSI4202 – Modul 3
BMP STSI4202 – Modul 4, Universitas Terbuka).

Tautan permanen Tampilkan induk

Re: Diskusi.3
oleh AJAY SUPRIADI 04001670 - Rabu, 22 Oktober 2025, 16:41

Penjelasan Anda mengenai model Spiral sangat komprehensif. Pertahankan gaya berpikir analitis seperti ini.

Tautan permanen Tampilkan induk

Re: Diskusi.3
oleh ILHAN RAHARJA NALANKO 051475472 - Selasa, 21 Oktober 2025, 16:37

Assalamualaikum wr.wb🙏

Izin memberikan Jawaban bapak tutor untuk diskusi di atas

1.alasan pemilihannya berdasarkan karakteristik proyek.

Dalam pengembangan aplikasi pemesanan makanan online seperti *FoodFast*, pemilihan model *Software Development Life Cycle* (SDLC) yang tepat sangat penting untuk memastikan produk dapat dikembangkan secara cepat, fleksibel, dan sesuai kebutuhan pengguna. Berdasarkan karakteristik proyek, model SDLC yang paling sesuai untuk FoodFast adalah **model Prototipe**. Menurut buku *STSI4202 Rekayasa Perangkat Lunak* terbitan Universitas Terbuka, model Prototipe digunakan ketika kebutuhan pengguna belum sepenuhnya dipahami di awal, dan diperlukan interaksi langsung dengan pengguna untuk memperoleh umpan balik yang berkelanjutan.

2.Mengatasi perubahan fitur yang sering terjadi

Model Prototipe memungkinkan pengembang membuat versi awal aplikasi atau *prototype* yang dapat segera diuji oleh pengguna. Dari hasil pengujian tersebut, tim pengembang mendapatkan masukan yang digunakan untuk memperbaiki dan menyempurnakan sistem pada iterasi berikutnya. Pendekatan ini sangat sesuai dengan kebutuhan FoodFast yang ingin mengembangkan produk dalam waktu cepat dan siap beradaptasi terhadap perubahan fitur berdasarkan masukan pengguna. Misalnya, jika pengguna merasa sistem pelacakan pesanan kurang akurat, tim dapat segera memperbaikinya tanpa harus mengulang seluruh proses pengembangan seperti pada model Waterfall.

3.kelebihan dan kekurangan model

model Prototipe memiliki keunggulan utama dalam mengatasi perubahan kebutuhan yang sering terjadi. Sifat iteratifnya memungkinkan setiap perubahan diterapkan secara bertahap, sehingga risiko kegagalan dapat diminimalkan. Dengan keterlibatan pengguna sejak tahap awal, hasil akhir lebih sesuai dengan ekspektasi pasar.

Namun demikian, model ini juga memiliki beberapa kelemahan. Proses revisi yang berulang dapat menambah waktu dan biaya pengembangan apabila tidak dikendalikan dengan baik. Selain itu, pengguna terkadang lebih fokus pada tampilan antarmuka daripada fungsi utama sistem, sehingga pengembang harus tetap menjaga keseimbangan antara estetika dan fungsionalitas.

Sekian Dari pendapat saya jika ada kekurangan nya mohon di berikan tambahan nya Bapak/ibu Tutor Terimakasih.

Sumber refrensi :

Universitas Terbuka. (2021). *Modul STSI4202 – Rekayasa Perangkat Lunak*. Tangerang Selatan: Universitas Terbuka.Modul 3 Software development life Cycle

https://surabaya.telkomuniversity.ac.id/model-model-software-development-life-cycle-sdlc/

https://www.dicoding.com/blog/metode-sdlc/

Tautan permanen Tampilkan induk

Re: Diskusi.3

oleh AJAY SUPRIADI 04001670 - Rabu, 22 Oktober 2025, 16:41

Anda telah memahami inti persoalan dengan baik. Model Prototipe memang fleksibel untuk perubahan fitur. Gunakan BMP STSI4202 Rekayasa Perangkat Lunak sebagai acuan dalam menjawab forum diskusi ini.

Tautan permanen Tampilkan induk

Re: Diskusi.3

oleh 054518659 DHAFFA NIZHAR ADINDHA PUTRA - Selasa, 21 Oktober 2025, 19:25

1. Di antara model SDLC (Waterfall, RAD, Spiral, dan Prototipe), mana yang paling cocok digunakan untuk proyek ini? Jelaskan alasan pemilihannya berdasarkan karakteristik proyek.
Saya pikir model prototipe (atau model prototipe/iteratif) adalah yang terbaik untuk startup. Sebagai hasil dari model prototipe, pembuatan "contoh cepat" atau versi minimal yang dapat segera diuji oleh pengguna mempercepat waktu pengembangan perangkat lunak. Prototipe memungkinkan pengguna mencoba versi awal (pilot) dan memberikan masukan, yang sangat baik untuk skenario "uji pengguna terlebih dahulu" sebelum produksi penuh. Selain itu, tim dapat melihat gambaran awal dan memberikan umpan balik lebih awal. Selain itu, protokol lebih tahan terhadap perubahan dibandingkan model yang kaku seperti Waterfall, yang memungkinkan iterasi cepat untuk penyesuaian berdasarkan umpan balik pengguna. Terakhir, menggunakan prototipe mengurangi kemungkinan kegagalan proyek karena memungkinkan penemuan kesalahan desain atau fitur lebih awal daripada menunggu hingga akhir fase pengembangan.
Dengan demikian, model prototipe sangat cocok untuk memenuhi kebutuhan startup dalam pengembangan cepat, pengujian fitur awal, fleksibilitas perubahan, dan minimisasi risiko.

2. Bagaimana model yang Anda pilih dapat mengatasi kemungkinan perubahan fitur yang sering terjadi?

Dengan memilih model prototipe, ada beberapa mekanisme yang membantu menangani perubahan fitur:
-Tahap awal adalah membuat prototipe atau model sederhana dari aplikasi. Ini dapat berupa versi minimal, seperti pendaftaran pengguna atau pencarian restoran. Aplikasi ini kemudian diuji pada pengguna atau stakeholder. Pengguna memberikan tanggapan.
-Tim pengembang menggunakan umpan balik ini untuk memperbaiki atau mengubah prototipe, dan kemudian melakukannya lagi (iterasi) hingga prototipe disetujui.
-Karena ini bukan pengembangan satu kali besar di akhir (seperti Waterfall), perubahan fitur yang muncul dapat ditangani dalam siklus prototipe sebelum peluncuran penuh.
-Dengan menggunakan protokol, penelitian pengguna yang sebenarnya (user testing) dapat dilakukan, sehingga fitur yang tidak sesuai dapat dibuang atau diubah lebih awal, yang mengurangi "biaya perubahan" yang signifikan di kemudian hari.
-Oleh karena itu, model ini mengurangi risiko karena fitur yang tidak sesuai tidak dibangun secara lengkap dan mahal

sebelum diuji.

3. Apa kelebihan dan kekurangan model SDLC yang Anda pilih dibandingkan dengan model lainnya dalam konteks proyek ini?

Keunggulan model prototipe untuk proyek FoodFast: Pengembangan dapat dilakukan lebih cepat dan pengguna dapat memberikan umpan balik langsung, sehingga fitur lebih sesuai dengan kebutuhan. Selain itu, karena iterasi yang mudah dilakukan, model ini mengurangi kemungkinan kegagalan dan lebih fleksibel terhadap perubahan.

Kekurangan model prototipe untuk proyek FoodFast : Analisis kebutuhan seringkali tidak cukup mendalam, dan ada kemungkinan munculnya scope creep (penambahan fitur terus-menerus), serta hasil akhir yang buruk jika prototipe tidak dirancang dengan baik. Selain itu, dokumentasi dan struktur pengembangannya tidak seformal Waterfall.

Perbandingan:
-Waterfall: cocok jika kebutuhan jelas, tapi kurang fleksibel.
-RAD: cepat, namun butuh tim ahli dan komponen siap pakai.
-Spiral: baik untuk proyek besar berisiko tinggi, tetapi terlalu kompleks untuk startup seperti FoodFast.

sumber referensi :
Universitas Terbuka. (2021). Modul STSI4202 – Rekayasa Perangkat Lunak. Jakarta: Universitas Terbuka.
Dicoding Indonesia. (2023). Metode SDLC (Software Development Life Cycle).
https://www.dicoding.com/blog/metode-sdlc/

Tautan permanen Tampilkan induk

Re: Diskusi.3
oleh AJAY SUPRIADI 04001670 - Rabu, 22 Oktober 2025, 16:43

Terima kasih atas penjelasan Anda, analisis terhadap model Prototipe sudah tepat mengingat kebutuhan FoodFast yang dinamis.

Tautan permanen Tampilkan induk

Re: Diskusi.3
oleh MARTO 055035643 - Selasa, 21 Oktober 2025, 20:20

Assalamu'alaikum

mohon izin untuk memberikan tanggapan pada sesi diskusi ini

1. Diantara model SDLC yang paling cocok digunakan untuk proyek ini adalah Model Spiral, karena model spiral menyediakan pengembangan dengan cara cepat yang sesuai dengan apa yang diinginkan oleh sebuah startup tersebut “mereka ingin aplikasi ini dikembangkan dengan cepat agar segera dapat digunakan pelanggan”.
2. Model ini dapat mengatasi kemungkinan perubahan fitur yang sering terjadi, karena pada Model spiral dibagi menjadi beberapa kerangka aktivitas, salah satunya adalah komunikasi dengan pelanggan, jadi model ini dapat mengatasi perubahan fitur yang sering terjadi, karena akivitas ini dapat membangun komunikasi yang efektif antara pengembang dan pelanggan.
3. Kelebihannya adalah model spiral merupakan model yang bisa memberikan jaminan kualitas yang paling baik untuk aplikasi berskala besar, model spiral menyediakan pengembangan dengan cara cepat dengan perangkat lunak yang memiliki versi yang terus bertambah fungsinya
Kekurangannya adalah model spiral jauh lebih kompleks dibanding dengan model lainnya, tidak cocok untuk proyek berskala kecil, dan setiap perubahan spesifikasi pasti beresiko pada molornya waktu pengerjaan dan membengkaknya biaya proyek.
referensi : BUKU MODUL MSIM4303 REKAYASA PERANGKAT LUNAK Rosa Ariani Sukamto.

Terima kasih

Tautan permanen Tampilkan induk

Re: Diskusi.3

oleh AJAY SUPRIADI 04001670 - Rabu, 22 Oktober 2025, 16:44

Anda memahami esensi SDLC dengan sangat baik, terutama terkait adaptasi terhadap perubahan.

Tautan permanen Tampilkan induk

Hide sidebars

Re: Diskusi.3

oleh 053929596 FEIZA RAHMAH ASSYIFA - Selasa, 21 Oktober 2025, 20:34

Nama: Feiza Rahmah Assyifa
NIM: 053929596
Prodi: Sistem Informasi

Selamat malam Ajay Supriadi, S.Kom., M.Kom. Izinkan saya memberikan tanggapan atas soal pada diskusi sesi 3 ini.

Pertanyaan

1. Di antara model SDLC (Waterfall, RAD, Spiral, dan Prototipe), mana yang paling cocok digunakan untuk proyek ini? Jelaskan alasan pemilihannya berdasarkan karakteristik proyek.
2. Bagaimana model yang Anda pilih dapat mengatasi kemungkinan perubahan fitur yang sering terjadi?
3. Apa kelebihan dan kekurangan model SDLC yang Anda pilih dibandingkan dengan model lainnya dalam konteks proyek ini?

Tanggapan diskusi :

1. Berdasarkan BMP Rekayasa perangkat lunak STSI4202 Modul 3 Hal 3.5 – 3.12, proyek pengembangan aplikasi foodfast ini paling cocok menggunakan model prototipe, jadi apa alasan saya memilih model prototipe ?

Alasan-nya :

- Pengembangan yang cepat, model ini memungkinkan pembuatan versi awal (prototipe) aplikasi dengan fitur utama sehingga startup bisa segera menunjukan fungsionalitas kepada pengguna.
- Uji coba fitur oleh pengguna, prototipe dapat di uji langsung oleh pengguna untuk mendapatkan umpan balik sebelum versi final di kembangkan.
- Fleksibel terhadap perubahan, model ini mendukung literasi dan revisi berdasarkan masukan pengguna, cocok untuk startup yang ingin terus menyempurnakan fitur.
- Minim resiko kegagalan, dengan melibatkan pengguna sejak awal, risiko pengembangan fitur yang tidak dibutuhkan atau tidak sesuai harapan bisa dikurangi.

2. Bagaimana model prototipe mengatasi fitur yang sering terjadi ?

Model prototipe adalah metode pengembangan perngembangan perangkat lunak dimana tim membuat versi awal atau contoh dari aplikasi terlebih dahuluu untuk menunjukan bagaimana aplikasi akan berkerja, sehingga dengan mudah mengatasi perubahan fitur, yaitu :

- Mudah di ubah, karena belum adanya final pada aplikasi tentu saja pengembang bisa dengan cepat mengganti atau menambahkan fitur sesuai masukan dari pengguna. Misalnya Ketika pengguna ingin mengubah menu, tim dapat dengan mudah mengubah atau menyisipkan menu tambahan ke fitur prototipe.
- Umpan balik langsung dari pengguna, pengguna bisa mencoba prototipr dengan memberi tahu apa yang kurang, tidak jelas, hingga apa yang perlu di tambahkan. Sehingga membantu pengembang memahami kebutuhan jangka Panjang pengguna, bukan hanya berasal dari asumsi pribadi.
- Tidak perlu menunggu lama, startup tidak perlu menunggu aplikasi selesai 100 % untuk memulai menguji. Prototipe bisa digunakan untuk uji coba awal, dan di perbaiki secara bertahap.
- Mengurangi risiko salah fitur, pengujian fitur di awal memungkinkan fitur tersebut lebih di butuhkan dan di sukai pengguna. Sehingga risiko tidak dibutuhkan atau tidak di sukai pengguna sangatlah kecil.

3. Kelebihan dan kekurangan metode prototipe, metode ini cocok untuk perusahaan startup dengan perubahan berulang, sehingga startup / pengguna dapat dengan cepat menggunakan aplikasi. Metode ini sangan fleksibel sehingga memudahkan pengguna.

Namun, tentu saja dimana ada kelebihan tentu ada kekurangannya, hal ini cenderung pada dokumentasi yang kurang lengkap karena metode ini tidak memerlukan dokumentasi lengkap di awal, dan juga fokus terlalu besar pada tampilan awal.

Sumber referensi:

- Rosa Ariani Sukamto. BMP Rekayasa Perangkat Lunak. MSIM430301. Tanggerang Selatan : Universitas Terbuka.

Tautan permanen Tampilkan induk

Hide sidebars

Re: Diskusi.3

oleh AJAY SUPRIADI 04001670 - Rabu, 22 Oktober 2025, 16:45

Sangat baik! Jawaban Anda sudah mencerminkan pemahaman komprehensif terhadap konsep Rekayasa Perangkat Lunak

Tautan permanen Tampilkan induk

Re: Diskusi.3

oleh MUHAMMAD RIZQI PRATAMA 053580633 - Selasa, 21 Oktober 2025, 21:39

1. Menurut saya Model Spiral cocok untuk aplikasi FoodFast.

Kenapa Model Spiral lebih cocok? karena model ini dirancang untuk menangani proyek yang kompleks, Meminimalkan risiko dan dapat dilakukan perubahan kebutuhan yang besar jika memang itu diperlukan.
Model Spiral bersifat iteratif yang dimaksudkan aplikasi dikembangkan dalam siklus-siklus kecil, sehingga aplikasi dapat dirilis lebih awal dengan fitur inti agar bisa digunakan segera oleh pelanggan sambil fitur lainnya dikembangkan ke siklus berikutnya. Model ini mengakomodasi pengujian dan umpan balik setiap putaran spiral memiliki tahapan evaluasi sehingga sangat cocok dengan kebutuhan aplikasi FoodFast. Maka model Spiral sangat cocok untuk pengembangan aplikasi FoodFast.

2. Model Spiral untuk mengatasi perubahan fitur melalui proses siklusnya, terdiri dari fase utama:
-Planning (perencanaan) untuk menentukan tujuan, batasan, atau alternatif jika terdapat umpan balik dari pengguna.
-Risk Analysis (Analisis Resiko) adalah akitifitas untuk menganalisis potensi risiko dari perubahan yang diusulkan, misal menambahkan sistem voucher / diskon yang diberikan pelanggan apakah berpotensi overload kunjungan server pada aplikasi?
-Engineering (Rekayasa)adalah aktifitas untuk membangun satu atau lebih representasi dari aplikasi perangkat lunak
-Evaluation (Evaluasi) adalah aktivitas yang dibutuhkan untuk mendapatkan umpan balik dari pengguna sebagai dasar untuk fase perencanaan ke siklus berikutnya.

3. Kelebihan Model Spiral:
- Jauh lebih fleksibel terhadap perubahan dibanding model Waterfall, karena model waterfall mengharuskan semua persyaratan ditentukan di awal dan tidak bisa diubah.
-Lebih kuat dalam manajemen risiko, model spiral memberikan keseimbangan antara kecepatan dan kehati-hatian.

Kekurangan Model Sprial:
-Sangat Kompleks sehingga sulit dikelola dibutuhkan manajer proyek yang berpengalaman untuk mengawasi masing-masing siklusnya
-Biaya dan waktu, karena setiap siklus melibatkan analisis risiko dan evaluasi secara berkala, sehingga model ini sudah bisa dipastikan memakan biaya besar dan waktu untuk setiap aktivitasnya.

Sumber: BMP Rekayasa Perangkat Lunak MSIM430301

Tautan permanen Tampilkan induk

Re: Diskusi.3

oleh AJAY SUPRIADI 04001670 - Rabu, 22 Oktober 2025, 16:46

Anda sudah mengaitkan kecepatan pengembangan dengan keberhasilan pasar. Analisis yang bagus!

Tautan permanen Tampilkan induk

Hide sidebars

Re: Diskusi.3

oleh 052471106 MUHAMAD FUJI SUBEKTI - Rabu, 22 Oktober 2025, 02:46

Nama: Muhamad Fuji Subekti
NIM: 052471106
Ijin Menanggapi

1. Model SDLC yang paling cocok adalah model Prototipe adalah pilihan yang paling cocok untuk proyek FoodFast. Alasan pemilihannya adalah:
- Pengembangan Cepat: Prototipe memungkinkan tim untuk mengembangkan versi awal aplikasi dengan cepat, sehingga dapat segera diuji oleh pengguna.
- Umpan Balik Pengguna: Dengan prototipe, pengguna dapat memberikan umpan balik lebih awal dan sering, yang membantu dalam penyesuaian fitur sesuai kebutuhan mereka.
- Fleksibilitas: Model ini memungkinkan perubahan fitur yang lebih mudah dilakukan selama proses pengembangan, sesuai dengan kebutuhan dan keinginan pengguna.

2. Mengatasi perubahan fitur
Model prototipe dapat mengatasi kemungkinan perubahan fitur melalui:
- Iterasi yang Sering: Setelah setiap fase prototipe, tim dapat melakukan review dan mengumpulkan umpan balik dari pengguna. Ini memungkinkan penyesuaian fitur sebelum melanjutkan ke pengembangan lebih lanjut.
- Fokus pada Pengguna: Dengan melibatkan pengguna dalam proses, tim dapat memahami kebutuhan mereka dengan lebih baik dan membuat perubahan yang relevan.

3. Kelebihan dan Kekurangan Model Prototipe
Kelebihan:
- Responsif terhadap Umpan Balik: Prototipe memungkinkan tim untuk beradaptasi dengan cepat terhadap umpan balik pengguna, menjadikan produk lebih sesuai dengan harapan pasar.
- Pengurangan Resiko: Dengan mengidentifikasi masalah lebih awal melalui pengujian, resiko kegagalan proyek dapat diminimalkan.
Kekurangan:
- Potensi Keterlambatan: Jika perubahan fitur terlalu sering, proyek bisa mengalami keterlambatan karena waktu tambahan diperlukan untuk mengimplementasikan perubahan.
- Keterbatasan Dokumentasi: Prototipe sering kali tidak didokumtasikan dengan baik, yang dapat menyebabkan kebingungan di masa depan saat mengembangkan fitur lebih lanjut.

Tautan permanen Tampilkan induk

Re: Diskusi.3

oleh AJAY SUPRIADI 04001670 - Rabu, 22 Oktober 2025, 16:47

Anda telah memahami inti persoalan dengan baik. Model Prototipe memang fleksibel untuk perubahan fitur. Gunakan BMP STSI4202 Rekayasa Perangkat Lunak sebagai acuan dalam menjawab forum diskusi ini.

Tautan permanen Tampilkan induk

Re: Diskusi.3

oleh TRIANA PUTRI WAHYUNI 053839459 - Rabu, 22 Oktober 2025, 09:32

Hide sidebars

Assalamu'alaikum wr.wb

1. Model SDLC yang Paling Cocok
Model yang paling cocok untuk proyek FoodFast adalah model Prototipe (Prototype Model).
Alasannya karena proyek ini perlu dikembangkan dengan cepat, fitur-fiturnya harus bisa diuji langsung oleh pengguna, dan ada kemungkinan besar terjadi perubahan berdasarkan masukan dari pengguna.
Dengan model Prototipe, pengembang bisa membuat versi awal aplikasi (prototipe) yang dapat dicoba oleh pengguna. Setelah diuji, tim bisa memperbaiki atau menambah fitur sesuai kebutuhan sebelum aplikasi resmi diluncurkan.

2. Cara Model Prototipe Mengatasi Perubahan Fitur
Model Prototipe sangat fleksibel terhadap perubahan.
Jika pengguna memberikan umpan balik atau ide baru, pengembang bisa langsung memperbarui prototipe tanpa harus memulai dari nol.
Proses ini bisa dilakukan berulang kali sampai aplikasi benar-benar sesuai dengan kebutuhan pengguna.
Dengan cara ini, perubahan fitur yang sering terjadi bisa diatasi dengan cepat dan efisien.

3. Kelebihan dan Kekurangan Model Prototipe Dibanding Model Lain
Kelebihan:
• Cepat menyesuaikan perubahan kebutuhan pengguna.
• Mempercepat proses pengembangan karena aplikasi awal bisa langsung diuji.
• Mengurangi risiko kegagalan karena pengguna ikut memberi masukan sejak awal.
• Komunikasi antara pengguna dan pengembang jadi lebih baik.
Kekurangan:
• Bisa memakan waktu lebih lama jika terlalu sering direvisi.
• Biaya bisa meningkat karena ada banyak perubahan dan perbaikan.
• Dokumentasi terkadang kurang lengkap karena fokus pada pengujian cepat.
Perbandingan singkat dengan model lain:
• Waterfall: Kurang cocok karena tidak fleksibel terhadap perubahan. Sekali satu tahap selesai, sulit untuk kembali ke tahap sebelumnya.
• RAD (Rapid Application Development): Hampir mirip dengan Prototipe, tapi lebih cocok untuk proyek dengan sumber daya dan tim besar.
• Spiral: Cocok untuk proyek besar yang berisiko tinggi, tetapi terlalu rumit dan mahal untuk ukuran startup seperti FoodFast.

Kesimpulan
Model Prototipe paling sesuai untuk proyek FoodFast karena mendukung pengembangan cepat, fleksibel terhadap perubahan, dan memungkinkan pengguna menguji langsung sebelum peluncuran. Dengan cara ini, startup bisa menyesuaikan aplikasinya dengan kebutuhan pasar dan mengurangi risiko kegagalan proyek.

Sumber
• Universitas Terbuka. STSI4202 – Rekayasa Perangkat Lunak. Jakarta: Universitas Terbuka.
• Pressman, R. S. Software Engineering: A Practitioner's Approach. McGraw-Hill.
• Sommerville, I. Software Engineering. Pearson Education.

Tautan permanen Tampilkan induk

Re: Diskusi.3
oleh AJAY SUPRIADI 04001670 - Rabu, 22 Oktober 2025, 16:52

Anda telah memahami inti persoalan dengan baik. Model Prototipe memang fleksibel untuk perubahan fitur.

Tautan permanen Tampilkan induk

Hide sidebars

Re: Diskusi.3
oleh MUHAMMAD FERDIANSYAH 053535258 - Rabu, 22 Oktober 2025, 19:08

Izin menjawab

1. Model SDLC yang paling cocok adalah Rapid Application Development (RAD). RAD cocok karena proyek FoodFast harus dikembangkan dengan cepat dan memiliki kemungkinan perubahan fitur yang sering terjadi. RAD menekankan pengembangan cepat melalui prototype yang bisa diuji oleh pengguna, memungkinkan iterasi dan revisi berdasarkan umpan balik pengguna. Contoh penelitian menggunakan RAD untuk aplikasi pemesanan makanan menunjukkan sistem dapat dikembangkan dengan cepat dan berfokus pada kemudahan pengguna seperti sistem pemesanan menggunakan QR Code di cafe.
2. RAD mengatasi perubahan fitur yang sering terjadi dengan pendekatan iterative dan incremental. Setiap modul atau prototype dikembangkan, diuji, dan diperbaiki berdasarkan masukan pengguna sebelum seluruh sistem dirilis penuh. Hal ini meminimalkan risiko kegagalan proyek karena masalah terdeteksi lebih awal dan perubahan dapat dilakukan secara cepat tanpa mengulang keseluruhan pengembangan.
3. Kelebihan RAD dibanding model lain (Waterfall, Spiral, Prototipe) dalam konteks ini:

- Kecepatan pengembangan lebih tinggi karena prototyping dan iterasi singkat.
- Fleksibel terhadap perubahan fitur berdasarkan umpan balik pengguna.
- Risiko kegagalan lebih rendah karena evaluasi dan pengujian berulang.

Kekurangan RAD:

- Membutuhkan keterlibatan pengguna aktif dan komitmen waktu mereka.
- Kurang cocok untuk proyek berskala sangat besar atau yang membutuhkan dokumentasi lengkap sejak awal.
- Bisa menimbulkan biaya lebih tinggi jika iterasi terlalu banyak.

Model Waterfall kurang cocok karena sifatnya linear dan sulit mengakomodasi perubahan setelah tahap awal. Spiral lebih fokus pada pengendalian risiko tapi bisa lebih lambat dan kompleks. Model Prototipe cenderung mirip RAD, tetapi RAD lebih terstruktur dalam iterasi dan pengembangan cepat.

Dengan mempertimbangkan kebutuhan startup FoodFast yang ingin cepat rilis, sering melakukan perubahan fitur, serta meminimalkan risiko kegagalan pada produk utama, RAD menjadi model paling ideal untuk digunakan dalam pengembangan aplikasi pemesanan makanan ini.

Sumber referensi:

- Pendapat sendiri
- MSIM 4303 / Modul 3

Tautan permanen Tampilkan induk

Re: Diskusi.3
oleh AJAY SUPRIADI 04001670 - Jumat, 24 Oktober 2025, 19:49

. Ide bagus! Model RAD memang cocok untuk pengembangan cepat seperti pada kasus FoodFast

Tautan permanen Tampilkan induk

Diskusi.3
oleh MIA AULIA 051618649 - Rabu, 22 Oktober 2025, 21:49

Assalamualaikum selamat malam pak, Izin melampirkan jawaban diskusi 3

Diantara Model SDLC (Waterfall, RAD, Spiral, dan Prototipe), Mana yang paling cocok digunakan untuk proyek ini? Jelaskan alasan pemilihannya berdasarkan karakteristik proyek.

Model Spiral adalah pilihan paling ideal untuk proyek FoodFast karena fleksibilitas inherentnya secara khusus menjawab dinamika dan kebutuhan mendesak yang khas dalam pengembangan produk startup.

Model Spiral adalah pendekatan SDLC yang bersifat iteratif dan berfokus pada manajemen risiko. Model ini tidak berjalan secara linier, tetapi melalui serangkaian siklus (spiral) yang berulang. Setiap siklus terdiri dari empat fase utama: penetapan tujuan, analisis risiko, rekayasa (pembangunan), dan perencanaan untuk iterasi berikutnya.

Kecocokan Model Spiral dengan FoodFast dapat dilihat dari poin-poin berikut:

• Pengembangan Cepat dan Bertahap (Iteratif)

Model Spiral ideal untuk memenuhi tuntutan kecepatan FoodFast. Melalui siklus iteratifnya, model ini memfasilitasi rilis fitur-fitur fundamental dalam waktu singkat. Pendekatan ini memungkinkan pelanggan untuk mengakses produk inti lebih awal, sambil memberi ruang bagi tim untuk melakukan penyempurnaan dan pengembangan berkelanjutan pada versi-versi selanjutnya.

• Pengujian Pengguna dan Umpan Balik Dini

FoodFast memanfaatkan Model Spiral untuk mengakomodasi kebutuhan validasi pengguna dan perubahan fitur yang dinamis. Setiap siklusnya menghasilkan prototipe yang siap diuji, dan umpan balik dari pengguna langsung diintegrasikan ke dalam perencanaan siklus berikutnya. Mekanisme ini memastikan fitur produk dapat beradaptasi dengan cepat terhadap kebutuhan pasar.

• Penanganan Perubahan Fitur (Fleksibilitas)

Kemampuan adaptif Model Spiral menjawab langsung kebutuhan FoodFast dalam menanggapi umpan balik pengguna secara cepat. Sifat fleksibel model ini memungkinkan penyesuaian fitur dilakukan secara teratur di setiap awal siklus, menghindarkan kebutuhan untuk merombak total produk. Bagi sebuah startup, keunggulan ini sangat vital untuk mempertahankan daya saing produk di tenguah pasar yang terus berevolusi.

• Meminimalkan Risiko Kegagalan Proyek (Manajemen Risiko)

FoodFast, sebagai produk utama, berfokus pada minimalisasi risiko kegagalan. Pendekatan ini didukung oleh Model Spiral yang memiliki tahap penilaian risiko dalam setiap putarannya. Proses berulang ini memastikan bahwa ancaman potensial terus dipantau dan ditangani secara sistematis, sehingga mengurangi kemungkinan kegagalan besar di kemudian hari.

Bagaimana model yang anda pilih dapat mengatasi kemungkinan perubahan fitur yang terjadi?

Keunggulan Model Spiral terletak pada kemampuannya mengakomodasi perubahan fitur yang dinamis melalui pendekatan iteratif dan manajemen risiko. Fleksibilitas ini memungkinkan setiap umpan balik atau kebutuhan baru diintegrasikan dengan lancar pada awal siklus, sehingga tim dapat tetap lincah beradaptasi tanpa mengganggu stabilitas proyek.

Berikut adalah mekanisme utama bagaimana Model Spiral menangani perubahan:

• Pengembangan Bertahap (Iteratif)

Keunggulan Model Spiral bagi FoodFast adalah kemampuannya mengintegrasikan perubahan dengan cerdas. Umpan balik tidak harus menunggu akhir proyek, tetapi dapat langsung dijadikan tujuan pada siklus berikutnya. Karena setiap spiral bersifat mandiri, proses ini meminimalkan gangguan pada pekerjaan yang sudah ada.

• Pemanfaatan Prototipe dan Umpan Balik

Model Spiral menciptakan siklus umpan balik yang cepat dengan secara rutin menghasilkan prototipe. Setiap rilis segera diuji oleh pengguna, dan umpan balik yang didapat wajib diintegrasikan ke dalam perencanaan siklus berikutnya. Proses ini memastikan bahwa setiap fitur yang dibangun bukan hanya bekerja dengan baik, tetapi juga benar-benar dibutuhkan dan disukai oleh pasar FoodFast.

• Manajemen Risiko yang Berkelanjutan

Fokus unik Model Spiral pada analisis risiko di setiap siklus memungkinkannya mengatasi ketidakpastian dengan cerdas. Model ini secara proaktif menganggap perubahan kebutuhan seperti umpan balik pengguna FoodFast sebagai risiko nyata yang perlu dikelola. Dengan mengalokasikan waktu dan anggaran secara eksplisit untuk perubahan di setiap iterasi, tim dapat menghindari asumsi kaku dan menjaga ketangguhan proyek terhadap dinamika pasar.

Apa kelebihan dan kekurangan proyek SDLC yang anda pilih dibandingkan model lainnya dibandingkan model lainnya dalam konteks proyek ini?

Kelebihan Model Spiral:

• Manajemen Risiko Tinggi

Model Spiral memiliki mekanisme manajemen risiko bawaan melalui fase Analisis Risiko di setiap siklus. Hal ini memastikan semua risiko proyek mulai dari aspek teknis hingga pasar teridentifikasi dan termitigasi secara sistematis.

• Fleksibilitas terhadap Perubahan

Model ini memiliki fleksibilitas tinggi terhadap perubahan berkat sifatnya yang iteratif dan inkremental. Setiap umpan balik atau permintaan fitur baru dapat dengan mudah diadopsi dan dijadikan bagian dari perencanaan siklus pengembangan berikutnya.

• Prototipe Awal

Menghasilkan prototipe fungsional lebih awal, sehingga memungkinkan validasi konsep dan pengujian pengguna sejak dini sesuai kebutuhan FoodFast.

• Pengembangan Bertahap

Produk dikembangkan dalam tahapan-tahapan yang teratur dan evolusioner. Pendekatan ini memungkinkan startup untuk melakukan rilis awal produk inti kepada pasar dengan lebih cepat.

Kekurangan Model Spiral:

• Biaya dan Waktu yang Mahal

Kelemahan utama model ini terletak pada biaya dan waktu yang lebih tinggi. Aktivitas seperti analisis risiko dan penyesuaian rencana yang dilakukan berulang di setiap spiral menambah kompleksitas dan beban operasional.

• Ketergantungan pada Analisis Risiko

Efektivitas model ini sangat bergantung pada kompetensi tim dalam melakukan identifikasi dan manajemen risiko. Keberhasilannya sangat ditentukan oleh kapabilitas analisis dan antisipasi risiko dari pihak yang menerapkannya.

• Kompleksitas Manajemen

Model ini memiliki tingkat kompleksitas manajemen yang tinggi, karena kerangka kerjanya mengintegrasikan berbagai fase yang saling terkait, mencakup aspek perencanaan, teknis, dan manajemen risiko secara simultan.

• Potensi Proyek Berjalan Tanpa Henti

Model ini memiliki potensi proyek berjalan tanpa henti karena sifatnya yang berulang. Tanpa kontrol yang ketat, dapat terjadi scope creep dan proyek dapat kehilangan batas akhir yang jelas.

Sumber:

BMP MSIM4303 Rekayasa Perangkat Lunak Modul 3

https://www.dicoding.com/blog/metode-sdlc/

Aspriyono, H. (2023). IMPLEMENTASI SPIRAL MODEL DALAM PENGEMBANGAN APLIKASI PEMBAYARAN KULIAH PADA ITBM BANYUWANGI. Fakultas Ilmu Komputer Universitas Dehasen Bengkulu.

Tautan permanen Tampilkan induk

Hide sidebars

Re: Diskusi.3
oleh AJAY SUPRIADI 04001670 - Jumat, 24 Oktober 2025, 19:50

Argumentasi Anda logis dan didukung dengan alasan yang kuat.

Tautan permanen Tampilkan induk

Re: Diskusi.3
oleh 054453816 ACH. RIYANTO - Kamis, 23 Oktober 2025, 08:27

1. Model SDLC pilihan kami adalah *Model Prototipe*

Berdasarkan analisis terhadap kebutuhan proyek FoodFast, dan merujuk pada materi BMP Rekayasa Perangkat Lunak (STSI4202 Modul 3), Model Prototipe (Prototype Model) merupakan pilihan yang paling tepat.

Alasan pemilihan ini didasarkan pada kesesuaian karakteristik model dengan tantangan yang dihadapi startup:

- Mendukung pengembangan cepat
  Model ini berfokus pada pembuatan versi awal (prototipe) yang mendemonstrasikan fungsionalitas inti. Ini sejalan dengan kebutuhan FoodFast untuk segera meluncurkan produk dan menunjukkannya kepada pengguna atau investor.
- Memfasilitasi uji coba pengguna
  Prototipe yang fungsional dapat segera diuji oleh pengguna akhir. Hal ini krusial bagi FoodFast yang ingin "menguji beberapa fitur oleh pengguna" sebelum dirilis penuh.
- Adaptif terhadap perubahan
  Model ini bersifat iteratif. Umpan balik dari pengguna dapat langsung dianalisis dan diimplementasikan dalam revisi prototipe selanjutnya. Ini sangat cocok untuk mengatasi skenario "perubahan fitur akan sering terjadi."
- Mitigasi risiko produk
  Dengan melibatkan pengguna secara intensif sejak fase awal, FoodFast dapat memvalidasi asumsi dan fitur. Risiko kegagalan proyek karena membangun produk yang tidak sesuai dengan kebutuhan pasar dapat diminimalkan.

2. Cara model prototipe mengatasi perubahan fitur

Model Prototipe secara inheren dirancang untuk mengelola dan mengakomodasi perubahan. Proses ini bekerja dengan cara berikut:

- Iterasi dan modifikasi cepat
  Prototipe bukanlah produk final; ia adalah model yang dapat diubah. Ketika pengguna memberikan masukan (misalnya, perubahan tata letak menu, penambahan filter pencarian), tim pengembang dapat memodifikasi prototipe tersebut dengan cepat tanpa harus membongkar arsitektur yang sudah kaku.
- Validasi umpan balik langsung
  Model ini menciptakan siklus umpan balik (feedback loop) yang ketat. Pengguna dapat langsung mencoba fitur dan memberikan koreksi. Ini membantu pengembang memahami kebutuhan riil pengguna, bukan hanya bekerja berdasarkan asumsi awal tim.

- Pengujian parsial
  Startup tidak perlu menunggu aplikasi selesai 100% untuk melakukan pengujian. Fitur-fitur utama dapat diuji secara terpisah dan diperbaiki secara bertahap, memungkinkan penyempurnaan yang konstan.
- Mengurangi risiko fitur gagal
  Dengan menguji fitur di awal, tim dapat segera mengidentifikasi fitur mana yang disukai, yang membingungkan, atau yang tidak dibutuhkan. Ini memastikan sumber daya pengembangan difokuskan pada fungsionalitas yang paling bernilai bagi pengguna.

Hide sidebars

3. Kelebihan dan kekurangan model prototipe (Konteks FoodFast)

Kelebihan:

- Keunggulan utama bagi FoodFast adalah kemampuannya beradaptasi dengan cepat. Startup dapat segera memiliki sesuatu yang fungsional untuk diuji, dan model ini sangat fleksibel dalam menerima perubahan berulang.
- Keterlibatan pengguna yang tinggi sejak awal memastikan produk akhir memiliki tingkat penerimaan (user acceptance) yang lebih baik.

Kekurangan:

- Potensi dokumentasi yang lemah
  Karena fokusnya adalah pada kecepatan dan iterasi prototipe, aspek dokumentasi formal sering terabaikan. Ini bisa menjadi masalah teknis di kemudian hari.
- Risiko terjebak pada aspek permukaan
  Ada kecenderungan bagi tim dan pengguna untuk terlalu fokus pada antarmuka pengguna (UI/UX) atau "tampilan awal". Hal ini dapat menyebabkan pengabaian aspek non-fungsional yang krusial, seperti keamanan, skalabilitas, dan arsitektur *backend* yang robust, yang sangat penting untuk aplikasi pemesanan makanan.

Referensi:

Rosa Ariani Sukamto. BMP Rekayasa Perangkat Lunak. MSIM430301.

Tautan permanen Tampilkan induk

Re: Diskusi.3
oleh AJAY SUPRIADI 04001670 - Jumat, 24 Oktober 2025, 19:51

Terima kasih atas penjelasan Anda, analisis terhadap model Prototipe sudah tepat mengingat kebutuhan FoodFast yang dinamis

Tautan permanen Tampilkan induk

Re: Diskusi.3
oleh I NYOMAN ARYA YUDIHARTA 052254338 - Kamis, 23 Oktober 2025, 08:43

Ijin menjawab Diskusi
Berdasarkan karakteristik proyek yang dihadapi oleh FoodFast, model Prototipe merupakan pilihan yang paling tepat untuk digunakan dalam pengembangan aplikasi pemesanan makanan online.
1. Alasan Pemilihan Model Prototipe
Model Prototipe sangat cocok diterapkan pada proyek yang:
Membutuhkan pengembangan cepat untuk segera diuji oleh pengguna, memiliki ketidakpastian pada kebutuhan sistem di awal, memungkinkan adanya perubahan dan penyempurnaan fitur secara berulang.
Dalam kasus FoodFast, startup ingin merilis aplikasi secepat mungkin sambil tetap menguji fitur-fitur utama seperti pemesanan, pembayaran online, dan pelacakan pesanan. Model Prototipe mendukung proses tersebut karena pengembang dapat membuat versi awal aplikasi (mockup) yang langsung diuji ke pengguna, sehingga tim bisa

mendapatkan umpan balik nyata sebelum sistem dikembangkan lebih lanjut.
kenapa tidak memilih model Spral meskipun sekilas tampak cocok karena
Model Spiral memang unggul dalam pengendalian risiko dan pengujian sistem besar, tetapi tidak efisien bagi startup seperti FoodFast yang memiliki sumber daya terbatas, membutuhkan pengembangan cepat, dan sering melakukan perubahan fitur berdasarkan umpan balik. Karena itu, model Prototipe lebih tepat karena fokusnya pada kecepatan, fleksibilitas, dan keterlibatan pengguna secara langsung.

2. Cara Model Prototipe Mengatasi Perubahan Fitur

Model Prototipe memiliki pendekatan iteratif, artinya sistem dikembangkan melalui beberapa versi atau iterasi. Setiap iterasi memungkinkan: Penambahan atau pengubahan fitur sesuai hasil uji coba pengguna, perbaikan terhadap kesalahan rancangan, dan penyesuaian antarmuka agar lebih ramah pengguna (user-friendly).
Dengan demikian, jika FoodFast menerima umpan balik bahwa, misalnya, pelanggan ingin menambahkan fitur “chat langsung dengan kurir” atau “pembatalan pesanan otomatis”, maka pengembang dapat memperbarui prototipe dan mengujinya kembali tanpa harus membangun sistem dari awal.

3. Kelebihan dan Kekurangan Model Prototipe

Kelebihan:
- Dapat mengurangi risiko kegagalan proyek karena pengguna dilibatkan sejak awal.
- Kebutuhan sistem lebih akurat karena dikonfirmasi langsung dengan pengguna melalui uji coba berulang.
- Waktu peluncuran lebih cepat karena versi awal aplikasi bisa segera dirilis untuk uji pasar (beta release).

Kekurangan:
- Membutuhkan komitmen waktu dan komunikasi intensif antara pengembang dan pengguna.
- Bisa menimbulkan ekspektasi pengguna yang berlebihan, karena mereka melihat prototipe seolah sudah produk final.
- Dokumentasi teknis terkadang kurang terfokus karena lebih banyak waktu digunakan untuk membangun versi uji coba.

4. Perbandingan dengan Model SDLC Lainnya

Waterfall: Tidak fleksibel terhadap perubahan karena setiap tahap harus diselesaikan sebelum tahap berikutnya. Cocok untuk kebutuhan yang sudah stabil, bukan untuk proyek yang masih bereksperimen seperti FoodFast.
RAD (Rapid Application Development): Sama-sama cepat, tetapi membutuhkan tim besar dan sumber daya tinggi untuk bekerja paralel. Tidak cocok untuk startup dengan sumber daya terbatas.
Spiral: Baik untuk proyek besar yang berisiko tinggi, tetapi relatif kompleks dan mahal untuk proyek startup kecil.

Kesimpulan

Model Prototipe adalah pilihan yang paling masuk akal karena secara langsung mengatasi tantangan kecepatan dan fleksibilitas yang merupakan kunci bertahan bagi sebuah startup. Risiko teknis dapat diatasi pada iterasi prototipe yang lebih matang atau diatasi dengan beralih ke Model Spiral setelah MPV (Minimum Viable Product) terbukti berhasil.

Sumber Referensi

- Sukamto, R. A. (2021). Rekayasa perangkat lunak. Universitas Terbuka.
- Pressman, R. S., & Maxim, B. R. (2020). Software engineering: A practitioner’s approach (9th ed.). McGraw-Hill Education.
- GeeksforGeeks. (2024). Prototyping Model in Software Engineering. Retrieved October 22, 2025, from https://www.geeksforgeeks.org/software-engineering-prototyping-model

Tautan permanen Tampilkan induk

Re: Diskusi.3

oleh AJAY SUPRIADI 04001670 - Jumat, 24 Oktober 2025, 19:52

Anda telah memahami inti persoalan dengan baik. Model Prototipe memang fleksibel untuk perubahan fitur.

Tautan permanen Tampilkan induk

Re: Diskusi.3

oleh RIANDI SOLIHIN 051676705 - Kamis, 23 Oktober 2025, 19:48

Hide sidebars

Izin menjawab diskusi tutor

Model SDLC yang paling cocok buat proyek FoodFast adalah model Spiral. Soalnya model ini fleksibel, bisa menyesuaikan kalau ada perubahan fitur, dan tiap tahap pengembangannya dilakukan bertahap sambil terus diuji. Ini pas banget buat startup yang butuh cepat rilis tapi masih sering ubah fitur dari masukan pengguna.
Model Spiral juga kuat dalam analisis risiko, jadi bisa bantu startup menghindari kegagalan besar di tengah jalan. Tiap siklusnya ada tahap perencanaan, pengembangan, dan evaluasi, jadi kalau ada masukan baru bisa langsung diperbaiki di tahap berikutnya.
Dibanding model lain:
• Waterfall kaku dan susah kalau banyak perubahan.
• RAD cepat tapi kadang kurang rapi kalau sistemnya kompleks.
• Prototipe bagus buat uji fitur awal tapi gak fokus ke manajemen risiko.
Jadi, Spiral paling pas buat FoodFast yang butuh cepat, fleksibel, dan tetap aman dari risiko gagal.

Referensi:
• Universitas Terbuka. (2020). BMP STSI4202 Rekayasa Perangkat Lunak.
• Pressman, R. S. & Maxim, B. R. (2020). Software Engineering: A Practitioner's Approach.

Tautan permanen Tampilkan induk

Re: Diskusi.3
oleh AJAY SUPRIADI 04001670 - Jumat, 24 Oktober 2025, 19:53

Bagus sekali, Anda sudah mengaitkan antara kebutuhan pengguna dengan model pengembangan yang dipilih.

Tautan permanen Tampilkan induk

Re: Diskusi.3
oleh AHMAD MINAN NAJIB 050480425 - Jumat, 24 Oktober 2025, 10:21

Assalamualaikum wr wb
Izin menjawab diskusi 3
1. Model SDLC yang Paling Cocok
Berdasarkan karakteristik proyek FoodFast yang memerlukan pengembangan cepat, pengujian fitur oleh pengguna sebelum rilis penuh, serta kemungkinan perubahan fitur yang sering akibat umpan balik, model RAD (Rapid Application Development) adalah yang paling cocok.
Alasan pemilihannya:
- Pengembangan cepat: RAD dirancang untuk mempercepat proses pengembangan melalui pendekatan iteratif dan paralel, di mana tim dapat bekerja secara simultan pada komponen-komponen aplikasi. Ini sesuai dengan tantangan proyek pertama, yaitu kebutuhan untuk segera meluncurkan aplikasi agar dapat digunakan pelanggan, tanpa menunggu siklus pengembangan yang panjang seperti pada model Waterfall.
- Pengujian oleh pengguna: RAD stres pembuatan prototipe fungsional yang dapat diuji oleh pengguna secara dini, memungkinkan validasi fitur seperti pencarian restoran, pemrosesan pesanan, dan pelacakan status secara real-time sebelum rilis penuh. Ini langsung mengatasi tantangan kedua.
- Fleksibilitas terhadap perubahan: Model ini mendukung iterasi cepat berdasarkan feedback, yang cocok dengan tantangan ketiga di mana fitur mungkin berubah sering. RAD tidak mengunci persyaratan di awal, berbeda dari Waterfall yang linier dan kaku.
- Kegagalan risiko minimal: Sebagai produk utama di pasar digital, RAD membantu mengurangi risiko dengan memungkinkan pembuatan prototipe dan pengujian secara bertahap, sehingga masalah dapat dideteksi lebih awal. Ini lebih baik daripada Spiral yang lebih kompleks untuk risiko tinggi, atau Prototyping yang lebih fokus pada eksplorasi awal tanpa struktur pengembangan penuh.
Dalam konteks Rekayasa Perangkat Lunak, RAD ditekankan sebagai model yang efektif untuk proyek dengan persyaratan yang berubah-ubah dan kebutuhan waktu yang ketat, karena mengintegrasikan prototyping dengan siklus pengembangan yang terstruktur.
2. Cara Model RAD Mengatasi Perubahan Fitur yang Sering Terjadi
RAD mengatasi perubahan fitur melalui pendekatan iteratif dan modular. Setiap iterasi (biasanya 2-3 bulan)

Hide sidebars

melibatkan:
- Pembuatan prototipe cepat: Tim membangun versi awal fitur (misalnya, sistem ulasan dan rating) yang dapat diuji oleh pengguna. Jika feedback menunjukkan perlunya perubahan (seperti menambah filter kategori restoran), prototipe dapat direvisi tanpa mengganggu seluruh sistem.
- Integrasi umpan balik: Setelah pengujian, perubahan dimasukkan ke dalam iterasi berikutnya, memungkinkan penyesuaian fitur seperti membaca pesanan atau pembayaran online berdasarkan masukan pelanggan. Ini berbeda dari model linier seperti Waterfall, yang sulit menangani perubahan tanpa memulai kembali proyek.
- Manajemen risiko: RAD menggunakan alat seperti CASE (Computer-Aided Software Engineering) untuk mendokumentasikan perubahan, memastikan bahwa modifikasi tidak meningkatkan risiko kegagalan. Dalam proyek FoodFast, ini berarti fitur dapat disesuaikan secara bertahap, seperti menambah opsi lokasi GPS, tanpa menghentikan pengembangan secara keseluruhan.
Dengan demikian, RAD meminimalkan dampak perubahan dengan fokus pada iterasi kecil dan responsif, sesuai dengan prinsip rekayasa perangkat lunak yang kaku.
3. Kelebihan dan Kekurangan Model RAD Dibandingkan Model Lain dalam Konteks Proyek Ini
Kelebihan RAD dibandingkan Waterfall, Spiral, dan Prototyping:
- Vs. Waterfall: Waterfall linier dan tidak fleksibel untuk perubahan, sehingga risiko tinggi jika feedback pengguna memerlukan revisi besar (misalnya, mengubah sistem pembayaran). RAD lebih cepat dan berulang, memungkinkan pengembangan paralel yang cocok untuk startup seperti FoodFast yang butuh peluncuran cepat. Kelebihan RAD: waktu pengembangan lebih singkat (2-6 bulan vs. 6-12 bulan Waterfall), dan lebih adaptif terhadap umpan balik.
- Vs. Spiral: Spiral baik untuk risiko tinggi dengan loop penilaian risiko, tapi lebih kompleks dan memakan waktu untuk proyek sederhana seperti aplikasi pemesanan makanan. RAD lebih efisien untuk kecepatan, dengan fokus pada pembuatan prototipe tanpa analisis risiko mendalam di setiap fase. Kelebihan RAD: sederhana dan langsung, mengurangi overhead untuk startup kecil.
- Vs. Prototyping: Prototyping murni bagus untuk persyaratan eksplorasi, namun kurang terstruktur untuk pengembangan penuh, sehingga risiko kegagalan tinggi jika prototipe tidak dikonversi ke produk akhir. RAD menggabungkan prototyping dengan siklus pengembangan yang lengkap, memastikan fitur seperti pendaftaran pengguna dapat diuji dan diintegrasikan secara sistematis. Kelebihan RAD: lebih komprehensif dan terukur untuk aplikasi multi-fitur.
Kekurangan RAD dibandingkan model lain:
- Vs. Waterfall: RAD kurang cocok untuk proyek dengan persyaratan stabil, karena iterasi dapat menyebabkan scope creep (perluasan cakupan) jika tidak dikelola, meningkatkan biaya. Dalam proyek FoodFast, ini bisa terjadi jika perubahan fitur terlalu sering tanpa kontrol.
- Vs. Spiral: RAD tidak memiliki mekanisme penilaian risiko eksplisit seperti Spiral, sehingga kurang ideal jika ada risiko teknis tinggi (misalnya, integrasi pembayaran online yang kompleks). Spiral lebih baik untuk mitigasi risiko jangka panjang, tetapi RAD lebih cepat untuk peluncuran awal.
- Vs. Pembuatan prototipe: RAD memerlukan tim yang terampil dalam alat CASE dan kolaborasi, yang mungkin mahal untuk startup kecil. Prototyping lebih murah untuk eksplorasi awal, tapi RAD lebih baik untuk pengembangan end-to-end.
Secara keseluruhan, RAD seimbang antara kecepatan, kerutan, dan risiko, menjadikannya pilihan optimal untuk proyek FoodFast meskipun memiliki kekurangan dalam lingkup pengelolaan. Jika risiko perubahan sangat tinggi, Spiral bisa menjadi alternatif, tapi RAD lebih sesuai untuk kebutuhan cepat startup ini.
Referensi:
BMP MSIM4303
Pembelajaran sesi 3
Terima kasih

Tautan permanen Tampilkan induk

Re: Diskusi.3
oleh AJAY SUPRIADI 04001670 - Jumat, 24 Oktober 2025, 19:54

Penjelasan singkat namun tepat sasaran! Model RAD memang unggul dalam kecepatan pengembangan.

Tautan permanen Tampilkan induk

Hide sidebars

Re: Diskusi.3

oleh BUNGA NUR ROSEANA 050428449 - Jumat, 24 Oktober 2025, 16:36

1. Startup FoodFast menghadapi tantangan klasik di dunia digital yaitu kebutuhan akan kecepatan, fleksibilitas terhadap perubahan, dan mitigasi risiko tinggi semua berpusat pada peluncuran produk digital utama mereka. Di antara berbagai Model Software Development Life Cycle (SDLC), menurut saya Model Spiral adalah yang paling sesuai untuk mengatasi tantangan ini.

Model Spiral dipilih karena ia secara unik menggabungkan karakteristik terbaik dari Model Prototipe yang berfokus pada validasi pengguna dan Model Waterfall yang berfokus pada kontrol dan dokumentasi, sambil menambahkan elemen kritis berupa Manajemen Risiko di setiap langkahnya.

Alasan memilih model spiral yaitu FoodFast ingin segera digunakan pelanggan. Model Spiral bekerja secara increment dan iteratif. Tim dapat merilis produk dalam versi yang fungsionalitasnya bertambah (increment) di setiap putaran spiral. Ini memungkinkan peluncuran versi paling dasar dari suatu produk yang dirilis untuk menguji ide di pasar dengan fitur-fitur esensial saja atau MVP (Minimum Viable Product) dengan fitur inti seperti pendaftaran pengguna dan pencarian sederhana dengan cepat, alih-alih menunggu aplikasi selesai sepenuhnya seperti pada Model Waterfall.

Kebutuhan untuk menguji fitur sebelum dirilis secara penuh dapat ditangani dengan sempurna. Dalam setiap putaran spiral, terdapat fase Rekayasa yang mencakup pembuatan prototipe dan pengujian. Prototipe ini bisa langsung diuji oleh sekelompok pengguna untuk memvalidasi ide, memastikan fitur pelacakan real-time atau sistem pembayaran bekerja sesuai ekspektasi pasar, dan mengumpulkan feedback untuk siklus berikutnya.

Karena ini adalah produk utama, risiko harus diminimalkan. Keunggulan terbesar Model Spiral adalah penekanan pada Analisis Risiko di awal setiap putaran. Sebelum tim mulai mengembangkan atau merencanakan fitur baru, mereka menganalisis potensi risiko. Dengan menganalisis dan memitigasi risiko secara berkelanjutan, FoodFast dapat menghindari pengembangan fitur mahal yang tidak dibutuhkan atau kegagalan teknis yang mendasar, sehingga meminimalkan risiko proyek secara keseluruhan.

2. Model Spiral dibangun untuk merangkul perubahan. Ini adalah model yang adaptif, berbeda jauh dengan Model Waterfall yang kaku.

Model ini mengatasi perubahan melalui siklus umpan balik yang berulang (iterasi). Setiap putaran spiral adalah siklus pengembangan lengkap.

Di akhir setiap putaran, tim melakukan Evaluasi Pelanggan, di mana produk seperti prototipe atau inkremen disajikan dan umpan balik mengenai perubahan fitur atau penyesuaian diterima. Umpan balik tersebut langsung menjadi persyaratan masukan untuk fase Perencanaan putaran spiral berikutnya.

Dengan cara ini, FoodFast dapat sering berinteraksi dengan pengguna. Misalnya, jika pengguna awal mengatakan fitur pencarian restoran berdasarkan kategori tidak intuitif, perubahan ini dapat segera diintegrasikan dan dirilis pada putaran spiral berikutnya tanpa perlu menunda peluncuran fitur inti lainnya. Proses ini memastikan aplikasi selalu berevolusi sesuai kebutuhan nyata pasar.

3. Kelebihan Model Spiral dibanding dengan model lain

Dibandingkan dengan Model Waterfall, Model Spiral jauh lebih unggul dalam proyek seperti FoodFast. Waterfall bersifat linear dan kaku, di mana perubahan besar di tengah jalan hampir mustahil dan biayanya sangat tinggi. Sebaliknya, sifat iteratif dari Model Spiral memungkinkan FoodFast untuk menerima perubahan fitur yang sering misalnya, menambahkan opsi pembayaran baru atau mengubah antarmuka pengguna tanpa mengacaukan keseluruhan projek. Selain itu, Waterfall hanya menguji di akhir, sementara Spiral menguji dan mengevaluasi di setiap putaran, secara signifikan mengurangi risiko kegagalan besar pada tahap akhir.

Dan ketika dibandingkan dengan Model Prototipe, Model Spiral membawa disiplin yang hilang. Model Prototipe

memungkinkan pengembangan yang cepat dan mendapatkan umpan balik awal yang baik, tetapi sering kali terjadi penambahan fitur atau persyaratan yang tidak terencana pada lingkup proyek yang sudah disepakati dan kurang memiliki manajemen proyek yang terstruktur. Model Spiral mengatasi hal ini dengan memasukkan fase Perencanaan dan Analisis Risiko di setiap putaran. Ini memastikan bahwa setiap prototipe yang dibuat atau fitur yang diuji memiliki batasan yang jelas, anggaran yang terukur, dan risiko yang sudah dimitigasi, sehingga proyek tidak menyimpang dan terkontrol.

Dibandingkan dengan Model RAD, yang juga fokus pada kecepatan, Model Spiral unggul dalam proyek dengan tingkat ketidakpastian yang tinggi. Sementara RAD cepat untuk fitur-fitur yang sudah terdefinisi, Model Spiral secara eksplisit didedikasikan untuk mengidentifikasi dan mengelola risiko. Untuk startup seperti FoodFast yang memasuki pasar digital dengan banyak ketidakpastian, fokus Model Spiral pada mitigasi risiko teknis, operasional, dan pasar di setiap fase memberikan lapisan keamanan yang tidak dimiliki RAD.

Kekurangan Model Spiral disbanding dengan model lain

Model Spiral lebih kompleks dan lebih mahal untuk diimplementasikan daripada Model Waterfall atau Model Prototipe sederhana. Ini membutuhkan tim yang sangat terampil dan berorientasi pada analisis risiko di setiap fase. Untuk startup FoodFast dengan anggaran terbatas, investasi awal dalam sumber daya manajerial dan teknis yang ahli dalam manajemen risiko mungkin terasa lebih berat.

Karena model ini sangat bergantung pada umpan balik pelanggan dan keputusan yang diambil setelah analisis risiko di setiap putaran, waktu penyelesaian total proyek bisa menjadi tidak jelas di awal. Ini berbeda dengan Model Waterfall, di mana jadwal sudah terdefinisi sejak awal. Bagi FoodFast yang ingin segera rilis, ketidakpastian ini bisa menjadi tantangan dalam hal perencanaan bisnis jangka panjang.

Secara keseluruhan, bagi FoodFast, keunggulan Model Spiral dalam mengelola risiko dan mengakomodasi perubahan melampaui kekurangannya dalam kompleksitas, menjadikannya pilihan paling strategis untuk membangun aplikasi yang adaptif dan sukses di pasar yang dinamis.

Sumber : Materi Inisiasi dan BMP MSISM 4303 Rekayasa Perangkat Lunak

Tautan permanen Tampilkan induk

Re: Diskusi.3

oleh AJAY SUPRIADI 04001670 - Jumat, 24 Oktober 2025, 19:54

> Penjelasan Anda cukup runtut. Model Spiral yang dipilih memang bisa menekan risiko dengan pengujian bertahap.

Tautan permanen Tampilkan induk

Re: Diskusi.3

oleh 053883097 SYAHRUL HASANUDIN - Sabtu, 25 Oktober 2025, 11:13

Nama : Syahrul Hasanudin
NIM : 053883097
UPBJJ : UT Jakarta Timur

Assalamualaikum Wr. Wb., Yth. Bapak Ajay Supriadi. Izin meberikan tanggapan untuk menjawab pertanyaan diskusi sesi 3 ini.

Jawaban No. 1

Karena startup FoodFast ini membutuhkan pembuatan aplikasi yang cepat, maka model SDLC yang tepat adalah model prototipe. Model prototipe memungkinkan tim mebuat versi awal aplikasi. Versi ini kemudian dicoba langsung oleh user, dan dari sana tim dapat mengumpulkan masukan. Jadi, aplikasi yang dikembangkan secara bertahap sambil

menyesuaikan dengan kebutuhan user yang terus berkembang. Contoh nyata perusahaan yang memakai model prototipe ini adalah Gojek dan Grab. Mereka dulu juga memulai dengan versi prototipe yang sederhana sebelum menambahkan fitur-fitur canggih seperti pelacakan real time atau sistem rating.

Jawaban No. 2

Model prototipe sangat cocok untuk menghadapi pendekatan ulang dan perbaikan (iteratif). Tim membuat versi awal, lalu user mencoba dan memberi tahu mana yang kurang nyaman atau butuh tambahan fitur. Lalu tim pengembang memperbaiki aplikasi sesuai masukan itu dan mengirim versi barunya lagi ke pengguna. Proses ini bisa diulang beberapa kali sampai aplikasi benar-benar sesuai dengan yang diharapkan.

Jawaban No. 3

Kelebihan model prototipe dibandingkan model lainnya, yaitu:

1. Risiko lebih rendah: karena tiap fitur diuji langsung sebelum dikembangkan sepenuhnya, jadi tim mengetahui lebih awal mana yang penting dan mana yang tidak diharapkan user.
2. Cepat diuji pasar: Dengan menggunakan model ini, FoodFast bisa segera me-release versi awal aplikasi untuk dicoba pelanggan.
3. Fleksibel: Sangat mudah menyesuaikan aplikasi jika ada masukan baru, seperti menambah fitur diskon atau opsi pembayaran baru.

Kekurangan dari model prototipe ini, yaitu:
1. Terdapat biaya tambahan: meskipun cepat di awal, tapi perbaikan yang sering bisa menambahkan biaya pengembangan.
2. Revisi yang berulang: Jika masukan dari user terus berubah, proses iterasi ini bisa memakan waktu yang lebih lama dari yang direncanakan.
3. Dokumentasi sering terabaikan: Karena fokus pada pengembangan yang cepat, terkadang catatan teknis atau dokumentasi menjadi kurang lengkap.

Berikut perbandingan dengan model yang lainnya:
1. Spiral, bagus untuk proyek yang risikonya sangat besar seperti pembangunan sistem untuk perusahaan besar, tapi terlalu kompleks untuk aplikasi startup yang lebih ringan seperti FoodFast ini.
2. RAD (*Rapid Application Development*) juga cepat digunakan, namun lebih cocok jika tim sudah berpengalaman penuh dengan ahli i bidangnya.
3. Jika menggunakan model Air Terjun (*Waterfall*), semua fitur harus selesai dahulu baru bisa diuji dan sulit diperbaiki jika ada perubahan yang mendadak.

Sumber Referensi: BMP MSIM 4303 Rekayasa Perangkat Lunak ditulis oleh Rosa Ariani Sukamto

Tautan permanen Tampilkan induk

Re: Diskusi.3
oleh AJAY SUPRIADI 04001670 - Minggu, 26 Oktober 2025, 16:41

Terima kasih atas penjelasan Anda, analisis terhadap model Prototipe sudah tepat mengingat kebutuhan FoodFast yang dinamis.

Tautan permanen Tampilkan induk

Re: Diskusi.3
oleh TRI WAHYU WIDIARTONO 052401701 - Sabtu, 25 Oktober 2025, 12:06

Hide sidebars

1. Di antara model SDLC (Waterfall, RAD, Spiral, dan Prototipe), mana yang paling cocok digunakan untuk proyek ini? Jelaskan alasan pemilihannya berdasarkan karakteristik proyek.

Berdasarkan hasil analisis saya terhadap karakteristik proyek aplikasi FoodFast, model Rapid Application Development (RAD) merupakan pendekatan yang paling tepat digunakan. Model RAD merupakan model SLDC yang ditujukan untuk pengembangan aplikasi secara cepat dalam waktu pengembangan yang relatif singkat. Model ini menggabungkan beberapa konsep dari moddel lain seperti waterfall dan incremental development, di mana setiap komponen perangkat lunak dikembangkan secara bertahap dan dapat dikerjakan paralel oleh beberapa tim pengembang.

Proyek FastFood memiliki karakteristik yang sangan sesuai dengan prinsip RAD. Startup ingin aplikasinya cepat dirilis ke pasar, sering melakukan perubahan fitur berdasarkan umpan balik pengguna, serta meminimalkan risiko kegagalan proyek. Model RAD memungkinkan hal ini karena proses pengembangannya dibagi menjadi beberapa fase yang bisa dijalankan secara bersamaan, yakni pemodelan bisnis, pemodelan data, pemodelan proses, pembuatan aplikasi, serta pengujian dan pergantian. Setiap fase menghasilkan komponen yang dapat diuji oleh pengguna dan diperbaiki sebelum sistem diintegrasikan secara keseluruhan. Dengan cara ini, FoodFast dapat melakukan evaluasi terhadap fitur penting seperti pencarian restoran, pembayaran online, dan pelacakan pesanan secara real-time tanpa harus menunggu penyelesaian seluruh sistem.

2. Bagaimana model yang anda pilih dapat mengatasi kemungkinan perubahan fitur yang sering terjadi?

Model RAD mendukung adaptasi terhadap perubahan fitur dengan cepat. Pengembangan RAD bersifat incremental dan reusable, artinya setiap modul aplikasi dapat dikembangkan dan dimodifikasi tanpa harus membangun ulang sistem dari awal. Ketika pengguna memberikan umpan balik, pengembang dapat memperbarui modul tertentu tanpa mengganggu modul lain. Fleksibilitas ini sangat sesuai dengan kondisi startup yang masih dalam tahap validasi produk dan pasar.

3. Apa kelebihan dan kekurangan model SLDC yang anda pilih dibandingkan dengan model lainnya dalam konteks proyek ini.

RAD memiliki sejumlah keunggulan. Dibandingkan Waterfall, RAD jauh lebih fleksibel terhadap perubahan dan kemungkinan keterlibatan pengguna secara langsung selama proses pengembangan. Dibandingkan Prototipe, RAD memiliki perencanaan dan tahapan yang lebih sistematis serta mendukung pengerjaan paralel. Sedangkan dibandingkan Spiral, RAD lebih efisien untuk proyek dengan skala menengah seperti startup yang memiliki lingkup fitur yang jelas. Namun, RAD juga memiliki kekurangan. Kelemahan utama RAD terletak pada kebutuhan sumber daya manusia yang besar dan berpengalaman. Model ini membagi proyek menjadi beberapa tim yang bekerja secara paralel, sehingga membutuhkan koordinasi dan komunikasi yang sangat baik antar tim. Model RAD tidak cocok digunakan untuk sistem yang sangat besar dan kompleks, karena semakin banyak komponen yang dikerjakan secara bersamaan, semakin sulit pula melakukan integrasi dan pengujian menyeluruh. Selain itu juga, RAD memerlukan keterlibatan pengguna secara intensif selama proses pengembangan. Apabila pengguna tidak aktif memberikan umpan balik, maka proses iterasi yang cepat justru menjadi tidak efektif. RAD juga kurang sesuai jika digunakan pada proyek yang memiliki risiko teknis tinggi atau menggunakan teknologi baru yang belum banyak dikuasai oleh pengebang.

Referensi

Universitas Terbuka. (2022). MSIM4303 - Rekayasa Perangkat Lunak. Modul 3

Tautan permanen Tampilkan induk

Re: Diskusi.3

oleh AJAY SUPRIADI 04001670 - Minggu, 26 Oktober 2025, 16:41

Ide bagus! Model RAD memang cocok untuk pengembangan cepat seperti pada kasus FoodFast.

Tautan permanen Tampilkan induk

Re: Diskusi.3

oleh YUSUP SUPRIATNA 052621317 - Sabtu, 25 Oktober 2025, 14:11

Hide sidebars

Selamat siang Bapak/Ibu Dosen dan teman-teman mahasiswa, izin menyampaikan pendapat.

Berdasarkan studi pada BMP STSI4202 Rekayasa Perangkat Lunak, model Spiral merupakan model SDLC yang paling sesuai untuk proyek pengembangan aplikasi *FoodFast*. Model ini menekankan pada proses iteratif yang disertai dengan evaluasi risiko di setiap tahap, sehingga sangat cocok untuk proyek startup yang dinamis dan berorientasi pada umpan balik pengguna.

1. Alasan pemilihan model Spiral

Model Spiral menggabungkan karakteristik dari model Waterfall dan Iteratif dengan menambahkan analisis risiko di setiap siklusnya. Pada proyek *FoodFast*, di mana fitur aplikasi bisa berubah sesuai kebutuhan pasar dan umpan balik pengguna, model ini memberikan fleksibilitas tinggi untuk menyesuaikan perubahan tersebut tanpa mengorbankan kestabilan sistem. Selain itu, setiap siklus Spiral memungkinkan prototipe diuji oleh pengguna sebelum versi final dikembangkan, sehingga risiko kesalahan atau kegagalan dapat diminimalkan sejak dini.

2. Cara model Spiral mengatasi perubahan fitur

Model Spiral memiliki tahapan berulang yang memungkinkan pengembang melakukan *refinement* terhadap kebutuhan dan desain setiap kali mendapatkan masukan dari pengguna. Dengan demikian, perubahan fitur dapat diakomodasi secara cepat di siklus berikutnya tanpa harus memulai dari awal seperti pada model Waterfall. Pendekatan ini juga membantu tim mengontrol biaya dan waktu karena perubahan dilakukan secara terencana dan terukur.

3. Kelebihan dan kekurangan model Spiral dibandingkan model lain

- Kelebihan:
  - Adaptif terhadap perubahan kebutuhan pengguna.
  - Mengutamakan identifikasi dan mitigasi risiko.
  - Memungkinkan uji coba bertahap dengan pengguna.
  - Memberikan kualitas produk yang lebih baik karena melibatkan evaluasi berulang.
- Kekurangan:
  - Membutuhkan sumber daya dan tim yang berpengalaman dalam manajemen risiko.
  - Biaya dan waktu bisa meningkat jika jumlah iterasi terlalu banyak.

Dibandingkan model Waterfall, model Spiral lebih fleksibel terhadap perubahan. Dibandingkan RAD, model ini lebih sistematis dan terstruktur. Sementara dibandingkan Prototipe, model Spiral memberikan pendekatan yang lebih komprehensif karena mencakup analisis risiko dan perencanaan jangka panjang.

Dengan demikian, model Spiral merupakan pilihan yang paling tepat untuk proyek *FoodFast* karena mampu menyeimbangkan antara kecepatan pengembangan, fleksibilitas terhadap perubahan, dan manajemen risiko yang baik.

Demikian pendapat saya, semoga dapat menambah wawasan dalam diskusi ini. Terima kasih.

Sumber Referensi:

- Munawar, M. (2020). *Rekayasa Perangkat Lunak (STSI4202)*. Universitas Terbuka.
- Pressman, R. S. (2010). *Software Engineering: A Practitioner's Approach*. McGraw-Hill.
- BMP STSI4202 Rekayasa Perangkat Lunak

Hide sidebars

Tautan permanen Tampilkan induk

Re: Diskusi.3

oleh AJAY SUPRIADI 04001670 - Minggu, 26 Oktober 2025, 16:43

Sangat baik! Jawaban Anda sudah mencerminkan pemahaman komprehensif terhadap konsep Rekayasa Perangkat Lunak

Tautan permanen Tampilkan induk

Re: Diskusi.3

oleh ULIKTA MUTAWAFINA 051517222 - Sabtu, 25 Oktober 2025, 19:04

1. Model SDLC yang Paling Cocok: Model Prototipe
Model Prototipe adalah model yang paling cocok digunakan untuk proyek pengembangan aplikasi FoodFast.
Hal ini karena proyek ini memiliki karakteristik yang membutuhkan:
• Pengembangan cepat (time-to-market tinggi),
• Keterlibatan pengguna secara aktif dalam pengujian fitur, dan
• Kemungkinan perubahan fitur yang sering berdasarkan umpan balik pengguna.
Dalam model Prototipe, tim pengembang membuat versi awal (prototype) dari aplikasi dengan fitur-fitur utama, kemudian menguji langsung kepada pengguna untuk mendapatkan umpan balik. Berdasarkan masukan tersebut, prototype disempurnakan secara berulang hingga memenuhi kebutuhan pengguna secara optimal.

_____

2. Cara Model Prototipe Mengatasi Perubahan Fitur yang Sering Terjadi
Model Prototipe bersifat iteratif dan fleksibel, sehingga sangat mudah beradaptasi terhadap perubahan kebutuhan. Ketika pengguna memberikan umpan balik tentang fitur yang perlu ditambah, diubah, atau diperbaiki, tim pengembang dapat segera memperbarui prototype dan mengujinya kembali.
Dengan demikian, kesalahan desain atau ketidaksesuaian kebutuhan dapat diketahui lebih awal sebelum sistem dikembangkan secara penuh, sehingga mengurangi risiko kegagalan proyek.
Contoh penerapan:
Jika pengguna merasa tampilan pelacakan pesanan terlalu rumit, tim dapat langsung memperbaiki antarmuka prototype dan meminta pengguna mencoba ulang tanpa perlu menunggu versi final aplikasi.

_____

3. Kelebihan dan Kekurangan Model Prototipe dalam Konteks Proyek FoodFast
Kelebihan:
• Responsif terhadap perubahan kebutuhan: Dapat menyesuaikan dengan cepat terhadap umpan balik pengguna.
• Keterlibatan pengguna tinggi: Pengguna terlibat sejak awal, sehingga hasil akhir lebih sesuai kebutuhan pasar.
• Waktu pengembangan lebih singkat: Versi awal aplikasi bisa segera digunakan untuk uji coba pasar (early release).
• Risiko kegagalan berkurang: Kesalahan dapat diperbaiki lebih awal sebelum peluncuran final.
Kekurangan:
• Biaya bisa meningkat jika terlalu banyak iterasi atau perubahan dilakukan berulang.
• Dokumentasi kadang kurang lengkap, karena fokus pada pengembangan cepat.
• Potensi kesalahpahaman antara pengguna dan pengembang jika prototype dianggap sebagai produk akhir sebelum waktunya.

_____

Hide sidebars

_____
Perbandingan Singkat dengan Model Lainnya
Model SDLC Kelebihan Kekurangan dalam konteks FoodFast
Waterfall Terstruktur dan mudah dikelola Tidak fleksibel terhadap perubahan, tidak cocok untuk kebutuhan yang sering berubah
RAD (Rapid Application Development) Cepat dalam pengembangan sistem besar dengan modul-modul Membutuhkan tim besar dan pengguna yang aktif, sulit jika sumber daya terbatas
Spiral Baik untuk proyek besar dan berisiko tinggi karena fokus pada analisis risiko Terlalu kompleks untuk startup kecil, memakan waktu dan biaya tinggi
Prototipe Cepat, fleksibel, dan melibatkan pengguna Risiko biaya meningkat jika iterasi berlebihan, dokumentasi minim

Kesimpulan
Model Prototipe merupakan pilihan paling tepat bagi startup FoodFast karena:
• Dapat mengakomodasi perubahan fitur dengan cepat,
• Memungkinkan pengujian langsung oleh pengguna,
• Mempercepat waktu peluncuran produk,
• Dan meminimalkan risiko kegagalan proyek di tahap awal pasar digital.

_____
Referensi:
• Pressman, R. S. (2015). Software Engineering: A Practitioner's Approach. McGraw-Hill Education.
• Sommerville, I. (2016). Software Engineering (10th Edition). Pearson Education.
• Universitas Terbuka. (2021). Modul MSIM4307 – Rekayasa Perangkat Lunak. Jakarta: Universitas Terbuka.

Tautan permanen Tampilkan induk

Re: Diskusi.3
oleh ULIKTA MUTAWAFINA 051517222 - Sabtu, 25 Oktober 2025, 19:15

mohon maaf izin merevisi jawaban pak
1. Model SDLC yang Paling Cocok menurut saya adalah Model Prototipe
Model Prototipe adalah model yang paling cocok digunakan untuk proyek pengembangan aplikasi FoodFast.
Hal ini karena proyek ini memiliki karakteristik yang membutuhkan:
• Pengembangan cepat (time-to-market tinggi),
• Keterlibatan pengguna secara aktif dalam pengujian fitur, dan
• Kemungkinan perubahan fitur yang sering berdasarkan umpan balik pengguna.

Dalam model Prototipe, tim pengembang membuat versi awal (prototype) dari aplikasi dengan fitur-fitur utama, kemudian menguji langsung kepada pengguna untuk mendapatkan umpan balik. Berdasarkan masukan tersebut, prototype disempurnakan secara berulang hingga memenuhi kebutuhan pengguna secara optimal.

2. Cara Model Prototipe Mengatasi Perubahan Fitur yang Sering Terjadi
Model Prototipe bersifat iteratif dan fleksibel, sehingga sangat mudah beradaptasi terhadap perubahan kebutuhan.
Ketika pengguna memberikan umpan balik tentang fitur yang perlu ditambah, diubah, atau diperbaiki, tim pengembang dapat segera memperbarui prototype dan mengujinya kembali.
Dengan demikian, kesalahan desain atau ketidaksesuaian kebutuhan dapat diketahui lebih awal sebelum sistem dikembangkan secara penuh, sehingga mengurangi risiko kegagalan proyek.

=> Contoh penerapan:
Jika pengguna merasa tampilan pelacakan pesanan terlalu rumit, tim dapat langsung memperbaiki antarmuka prototype dan meminta pengguna mencoba ulang tanpa perlu menunggu versi final aplikasi.

3. Kelebihan dan Kekurangan Model Prototipe dalam Konteks Proyek FoodFast
=> Kelebihan:

• Responsif terhadap perubahan kebutuhan: Dapat menyesuaikan dengan cepat terhadap umpan balik pengguna.
• Keterlibatan pengguna tinggi: Pengguna terlibat sejak awal, sehingga hasil akhir lebih sesuai kebutuhan pasar.
• Waktu pengembangan lebih singkat: Versi awal aplikasi bisa segera digunakan untuk uji coba pasar (early release).
• Risiko kegagalan berkurang: Kesalahan dapat diperbaiki lebih awal sebelum peluncuran final.

=> Kekurangan:
• Biaya bisa meningkat jika terlalu banyak iterasi atau perubahan dilakukan berulang.
• Dokumentasi kadang kurang lengkap, karena fokus pada pengembangan cepat.
• Potensi kesalahpahaman antara pengguna dan pengembang jika prototype dianggap sebagai produk akhir sebelum waktunya.

sumber Referensi:
• Pressman, R. S. (2015). Software Engineering: A Practitioner's Approach. McGraw-Hill Education.
• Sommerville, I. (2016). Software Engineering (10th Edition). Pearson Education.
• Modul MSIM4307 – Rekayasa Perangkat Lunak. Jakarta: Universitas Terbuka.

Tautan permanen Tampilkan induk

Re: Diskusi.3
oleh AJAY SUPRIADI 04001670 - Minggu, 26 Oktober 2025, 16:44

Argumentasi Anda jelas dan disampaikan dengan runtut

Tautan permanen Tampilkan induk

Re: Diskusi.3
oleh 054313301 ADISTI MELANI DIVARA MAHARANI - Sabtu, 25 Oktober 2025, 22:57

Nama : Adisti Melani Divara Maharani
NIM : 054313301
UPBJJ : Surabaya

Selamat malam, izin menjawab pertanyaan - pertanyaan diskusi 3 diatas :

1. Untuk proyek foodfast, **model spiral** adalah yang paling cocok.
- Alasan Pemilihan
a. Model spiral menggabungkan kelebihan dari model *Prototipe* dan *Waterfall*, serta fokus kuat pada analisis risiko.
b. Model Spiral memungkinkan perubahan atau penambahan fitur pada setiap tahapan pengembangan (*iterasi*).
c. Model Spiral memungkinkan pengembangan sistem dilakukan secara bertahap.
Ciri khas utama model ini adalah siklus pengembangan yang dilakukan berulang (*spiral*), di mana setiap putaran menghasilkan versi produk yang lebih lengkap dan stabil berdasarkan hasil evaluasi pengguna serta analisis risiko yang dilakukan sebelumnya.

2. Cara model Spiral mengatasi perubahan fitur :
Model Spiral bersifat iteratif dan evolusioner, artinya:
- Setiap *spiral* (putaran) menghasilkan versi yang lebih baik dari aplikasi.
- Perubahan fitur atau masukan pengguna bisa langsung diterapkan pada putaran berikutnya.
- Startup dapat meluncurkan versi awal (MVP) lebih cepat, lalu melakukan penyesuaian bertahap tanpa mengulang dari awal.
- Analisis risiko di setiap tahap membantu meminimalkan dampak dari perubahan besar.

3. Kelebihan dan Kekurangan Model Spiral :
a. Kelebihan
- Pengembangan lebih cepat dan efisien, ideal untuk startup yang ingin meluncurkan produk segera.
- Pengguna dapat memberikan umpan balik langsung pada tiap iterasi.

- Risiko kegagalan rendah karena ada analisis risiko di setiap tahap.
- Dapat dikombinasikan dengan elemen dari Waterfall atau prototyping, memberikan keseimbangan antara struktur dan adaptasi.

b. Kekurangan

- Membutuhkan analisis risiko mendalam di setiap siklus, yang bisa memakan waktu dan sumber daya.
- Analisis risiko memerlukan tenaga ahli dan waktu tambahan.
- Diperlukan tim yang berpengalaman dalam manajemen risiko, karena kesalahan dalam analisis bisa menyebabkan siklus yang tidak produktif.
- Biaya bisa meningkat jika terlalu banyak iterasi dilakukan.

Tautan permanen Tampilkan induk

Hide sidebars

Re: Diskusi.3

oleh AJAY SUPRIADI 04001670 - Minggu, 26 Oktober 2025, 16:46

> Pemilihan model Spiral Anda sudah tepat karena menekankan aspek kontrol risiko, tambahkan refernesi BMP STSI4202 Rekayasa Perangkat Lunak sebagai acuan dalam menjawab forum diskusi in

Tautan permanen Tampilkan induk

Re: Diskusi.3

oleh MUHAMAD SATRIA RIZKY 051464226 - Minggu, 26 Oktober 2025, 09:38

> Izin menanggapi diskusi
>
> 1. Model SDLC yang paling cocok:
> Model yang paling cocok untuk proyek FoodFast adalah model Spiral.
> Model ini sesuai karena proyek startup seperti FoodFast membutuhkan pengembangan cepat, fleksibel, dan bertahap, serta memiliki risiko tinggi jika fitur gagal di pasaran. Spiral menggabungkan kelebihan Waterfall (terstruktur) dan Prototipe (pengujian berulang), sehingga cocok untuk proyek yang fiturnya masih bisa berubah mengikuti umpan balik pengguna.
>
> 2. Cara mengatasi perubahan fitur:
> Model Spiral bersifat iteratif (berulang). Setiap putaran atau tahap memungkinkan tim melakukan evaluasi, memperbaiki, dan menambah fitur baru berdasarkan masukan pengguna. Dengan begitu, perubahan dapat dilakukan tanpa harus memulai pengembangan dari awal, sehingga proses tetap efisien dan terkontrol.
>
> 3. Kelebihan dan kekurangan dibanding model lain:
> - Kelebihan:
> - Fleksibel terhadap perubahan.
> - Ada evaluasi risiko di setiap tahap.
> - Hasil bisa diuji secara bertahap sebelum rilis penuh.
> - Kekurangan:
> - Membutuhkan tim yang berpengalaman dalam analisis risiko.
> - Bisa memakan waktu dan biaya lebih besar jika tidak direncanakan dengan baik.
>
> Dibanding model lain:
> - Waterfall kurang cocok karena sulit menyesuaikan perubahan.
> - RAD cepat tapi butuh kebutuhan yang sudah jelas.
> - Prototipe membantu memahami kebutuhan, tapi kurang fokus pada pengendalian risiko.
>
> Kesimpulan:
> Model Spiral paling tepat untuk proyek FoodFast karena mampu menyeimbangkan kecepatan, fleksibilitas, dan pengelolaan risiko dalam pengembangan aplikasi startup.
>
> Referensi:
> Universitas Terbuka. (2023). BMP STSI4202 Rekayasa Perangkat Lunak.
> Materi Inisiasi 3 RPL – Software Development Life Cycle (SDLC), Fakultas Sains dan Teknologi UT.

Tautan permanen Tampilkan induk

Re: Diskusi.3

oleh AJAY SUPRIADI 04001670 - Minggu, 26 Oktober 2025, 16:46

> Penjelasan Anda cukup runtut. Model Spiral yang dipilih memang bisa menekan risiko dengan pengujian bertahap.

Tautan permanen Tampilkan induk

Hide sidebars

Re: Diskusi.3

oleh 054284802 YULI ASMARAWATI - Minggu, 26 Oktober 2025, 13:58

Nama: Yuli Asmarawati
NIM: 054284802

1. Di antara model SDLC (Waterfall, RAD, Spiral, dan Prototipe), mana yang paling cocok digunakan untuk proyek ini adalah Model Prototipe (Prototyping Model) karena model ini termasuk dalam kategori pengembangan ditambahkan secara bertahap/inkremental dan dapat digunakan untuk menyambungkan ketidakpahaman pelanggan mengenai hal teknis dan memperjelas spesifikasi kebutuhan yang diinginkan pelanggan kepada pengembang perangkat lunak, atau dengan kata lain pelanggan tidak memahami mengenai perangkat lunak sehingga perlua adanya contoh tampilan yang dapat membuat pelanggan terbayang akan perangkat lunak yang akan dikembangkan. Program prototipe ini dievaluasi oleh pelanggan atau user sampai ditemukan spesifikasi yang sesuai dengan keinginan pelanggan atau user.
2. Model yang saya pilih dapat mengatasi kemungkinan perubahan fitur yang sering terjadi dengan cara melakukan perjanjian antara pengembangan perangkat lunak dengan pelanggan (customer) atau usernagar model prototipe hanya digunakan untuk mendefinisikan spesifikasi kebutuhan perangkat lunak, tapi tidak untuk seluruh proses pengembangan seluruh sistem perangkat lunak.
Model prototipe cocok digunakan untuk menjabarkan kebutuhan pelanggan secara lebih detail karena pelanggan sering kali kesulitan menyampaikan kebutuhannya secara detail tanpa melihat gambaran yang jelas. Untuk mengantisipasi agar proyek dapat berjalan sesuai dengan target waktu dan biaya di awal, maka sebaiknya spesifikasi kebutuhan sistem harus sudah disepakati oleh pengembang dengan pelanggan secara tertulis. Dokumen tersebut akan menjadi patokan agar spesifikasi kebutuhan sistem masih dalam ruang lingkup proyek.
3. Kelebihan model Prototipe yang saya pilih dibandingkan dengan model lainnya dalam konteks proyek ini adalah
a. model ini termasuk dalam kategori pengembangan ditambahkan secara bertahap/inkremental
b. dapat digunakan untuk menyambungkan ketidakpahaman pelanggan mengenai hal teknis
c. memperjelas spesifikasi kebutuhan yang diinginkan pelanggan kepada pengembang perangkat lunak
Sedangkan Kelemahan Model Prototipe yaitu
a. pelanggan dapat sering mengubah-ubah atau menambah-tambah spesifikasi kebutuhan karena menganggap aplikasi sudah dengan cepat dikembangkan karena adanya iterasi ini dapat menyebabkan pengembangan banyak mengalah dengan pelanggan karena perubahan atau penambahan spesifikasi kebutuhan perangkat lunak.
b. pengembangan lebih sering mengambil kompromi dengan pelanggan untuk mendapatkan prototipe dengan waktu yang cepat sehingga pengembangan lebih sering melakukan segala cara (tanpa idealisme) guna menghasilkan prototipe untuk didemonstrasikan.

Sumber: Modul MSIM4303 Rekayasa Perangkat Lunak

Tautan permanen Tampilkan induk

Re: Diskusi.3

oleh AJAY SUPRIADI 04001670 - Minggu, 26 Oktober 2025, 16:47

> Penjelasan Anda menunjukkan pemahaman mendalam terhadap fleksibilitas model Prototipe

Tautan permanen Tampilkan induk

Hide sidebars

Re: Diskusi.3

oleh WENDI BUDIARSO 053647749 - Minggu, 26 Oktober 2025, 14:16

Nama : Wendi Budiarso
NIM : 053647749

1. Model SDLC yang cocok untuk proyek tersebut adalah model Spiral
Alasan : Model ini dipilih untuk proyek FoodFast karena menekankan manajemen risiko, memungkinkan pengembangan bertahap dengan umpan balik pengguna, memberikan fleksibilitas untuk perubahan, keterlibatan pengguna ada di setiap pengujian, dan berfokus pada kualitas. Ini semua aspek yang penting untuk kesuksesan proyek fastfood tersebut.

2. Model Spiral dapat mengatasi kemungkinan perubahan fitur yang sering terjadi dengan beberapa cara berikut:

- Model Spiral dibangun di atas siklus iteratif yang memungkinkan pengembangan dilakukan secara bertahap. Setiap iterasi mencakup perencanaan, analisis risiko, pengembangan, dan evaluasi, sehingga perubahan dapat diintegrasikan dalam setiap siklus.

- Di setiap putaran siklus, analisis risiko dilakukan untuk mengidentifikasi potensi masalah atau kebutuhan baru. Dengan melakukan evaluasi ini secara berkala, tim dapat memahami dan merespons kebutuhan perubahan fitur sebelum melanjutkan ke fase pengembangan berikutnya.

- Model Spiral mendorong pembuatan prototipe selama fase pengembangan. Prototipe ini dapat diuji oleh pengguna, dan umpan balik yang diterima dapat langsung diterapkan untuk memperbaiki atau mengubah fitur yang ada, memungkinkan pengembang untuk merespons dengan cepat terhadap perubahan.

- Pengguna dilibatkan dalam setiap siklus, pengembang dapat mengumpulkan umpan balik yang relevan mengenai fitur yang ada. Jika pengguna menyarankan perubahan, hal ini dapat langsung dipertimbangkan dan diimplementasikan di iterasi berikutnya

- Model Spiral tidak terikat pada rencana yang kaku. Ini memberi pengembang fleksibilitas untuk melakukan perubahan desain dan fitur berdasarkan umpan balik dan analisis risiko, sehingga dapat menyesuaikan aplikasi dengan kebutuhan pengguna yang terus berkembang.

- Model ini menekankan pentingnya komunikasi yang efektif antara tim pengembang dan pemangku kepentingan. Dengan menjaga saluran komunikasi terbuka, perubahan yang diperlukan dapat segera dibahas dan diimplementasikan.

3. Berikut adalah kelebihan dan kekurangan model Spiral dibandingkan dengan model SDLC lainnya

Kelebihan :
a. Manajemen Risiko yang Baik:
- Model Spiral fokus pada identifikasi dan mitigasi risiko di setiap iterasi. Ini membantu mengurangi kemungkinan kegagalan di tahap akhir dan meningkatkan peluang keberhasilan proyek.

b. Fleksibilitas untuk Perubahan :
- Model ini mendukung perubahan selama proses pengembangan. Jika ada kebutuhan baru atau umpan balik pengguna, perubahan dapat diterapkan tanpa mengganggu keseluruhan proyek.

c. Iterasi dan Prototipe :
- Setiap siklus dalam model Spiral melibatkan pembuatan prototipe, yang memungkinkan pengguna untuk menguji fitur dan memberikan umpan balik secara langsung. Ini membantu memastikan bahwa produk akhir sesuai dengan harapan pengguna.

d. Keterlibatan Pengguna :

- Pengguna dilibatkan dalam setiap fase, yang meningkatkan kepuasan dan memastikan bahwa kebutuhan mereka terpenuhi.

e. Pengujian Berkelanjutan :
- Dengan fokus pada pengujian di setiap tahap, model Spiral membantu menjaga kualitas produk dan memungkinkan perbaikan berkelanjutan.

Kekurangan :

a. Kompleksitas dalam Manajemen :
- Model Spiral dapat menjadi kompleks dan sulit dikelola, terutama untuk tim yang tidak terbiasa dengan pendekatan ini. Memerlukan pemahaman yang baik tentang manajemen risiko dan perencanaan.

b. Biaya yang Lebih Tinggi :
- Karena melibatkan beberapa iterasi dan pengujian, biaya pengembangan bisa lebih tinggi dibandingkan dengan model yang lebih sederhana seperti Waterfall.

c. Waktu yang Lebih Lama :
- Proses iteratif dan evaluasi risiko mungkin memerlukan lebih banyak waktu dibandingkan dengan model linear. Ini bisa menjadi tantangan jika ada kebutuhan untuk pengembangan yang sangat cepat.

d. Ketergantungan pada Umpan Balik Pengguna : Keberhasilan model ini sangat bergantung pada umpan balik yang konstruktif dari pengguna. Jika pengguna tidak terlibat secara aktif, perubahan yang diperlukan mungkin tidak teridentifikasi dengan baik.

Kesimpulan
Model Spiral menawarkan kelebihan dalam fleksibilitas dan manajemen risiko, yang sangat relevan untuk proyek FoodFast yang memerlukan adaptasi cepat terhadap umpan balik pengguna.

Sumber : MSIM4303

Tautan permanen Tampilkan induk

Re: Diskusi.3
oleh AJAY SUPRIADI 04001670 - Minggu, 26 Oktober 2025, 16:52

Anda sudah memaparkan alasan pemilihan model dengan tepat dan didukung logika yang baik.

Tautan permanen Tampilkan induk

Re: Diskusi.3
oleh RACHMAT SAISHO 050216292 - Minggu, 26 Oktober 2025, 19:08

1. Pemilihan Model RAD dan Alasan

RAD dipilih karena karakteristik proyek FoodFast yang membutuhkan:

- Kecepatan pengembangan untuk segera diluncurkan ke pasar
- Fleksibilitas menerima perubahan fitur berdasarkan umpan balik pengguna
- Siklus iteratif untuk menguji fitur sebelum rilis penuh

Karakteristik RAD yang sesuai dengan kebutuhan FoodFast:

- Berorientasi pada komponen reusable - komponen seperti sistem pembayaran dan registrasi dapat dikembangkan paralel

- Siklus perkembangan pendek - memungkinkan rilis fitur secara bertahap
- Prototyping iteratif - fitur dapat diuji dan disempurnakan berdasarkan feedback

2. Kemampuan RAD Menangani Perubahan Fitur

Model RAD mengatasi perubahan fitur melalui:

- Siklus Development Pendek: Setiap siklus 2-4 minggu memungkinkan adaptasi cepat terhadap perubahan requirement
- Feedback Langsung: Pengguna terlibat aktif dalam mengevaluasi prototype setiap iterasi
- Komponen Modular: Perubahan pada satu fitur (contoh: sistem pembayaran) tidak mengganggu fitur lainnya
- Prioritas Fleksibel: Fitur dapat diatur ulang berdasarkan urgensi dan nilai bisnis

3. Kelebihan dan Kekurangan RAD vs Model Lain

Kelebihan RAD dibanding model lain:

- vs Waterfall: RAD lebih adaptif terhadap perubahan, sedangkan Waterfall terlalu kaku untuk requirement yang dinamis
- vs Spiral: RAD lebih cepat untuk produk sederhana seperti aplikasi makanan, siklus Spiral lebih panjang karena fokus pada risk analysis
- vs Prototype: RAD menghasilkan produk siap pakai, bukan sekedar prototype untuk studi kelayakan

Kekurangan RAD:

- Membutuhkan tim dengan skill teknis tinggi
- Kurang cocok untuk proyek dengan skala sangat besar
- Ketergantungan pada komponen reusable
- Membutuhkan keterlibatan aktif pengguna

Berdasarkan BMP STSI4202, RAD tepat untuk proyek dengan waktu terbatas, requirement jelas meski berpotensi berubah, dan tim berpengalaman - yang semuanya sesuai dengan kondisi FoodFast.

Dengan RAD, FoodFast dapat merilis MVP (Minimum Viable Product) dalam waktu singkat, kemudian menyempurnakannya secara iteratif berdasarkan feedback pasar, sehingga meminimalkan risiko kegagalan produk.

Tautan permanen Tampilkan induk

Re: Diskusi.3

oleh 054546411 AIMAN YUSUF WICAKSONO - Minggu, 26 Oktober 2025, 20:06

Model SDLC yang paling cocok untuk proyek FoodFast adalah Model Spiral.

1. Alasan Pemilihan Model Spiral

1. Model Spiral paling ideal karena mampu menangani risiko tinggi (ini produk utama *startup*) dan perubahan yang sering (dari *feedback* pengguna), sambil tetap memungkinkan rilis bertahap (*prototipe*).
2. Mengatasi Risiko & Perubahan: Setiap putaran (*spiral*) mencakup fase Analisis Risiko dan fase Prototipe. Ini artinya FoodFast bisa menguji fitur inti (misalnya, pembayaran) dengan pengguna sejak dini dan menyesuaikan persyaratan sebelum mengeluarkan biaya besar.
3. Kecepatan Awal: Model ini memungkinkan rilis MVP (*Minimum Viable Product*) atau versi beta awal dengan cepat, memenuhi kebutuhan *time-to-market* mereka.

2. Mengatasi Perubahan Fitur

Spiral mengatasi perubahan melalui sifat Iteratif dan Bertahapnya.

Umpan balik (*feedback*) pengguna dari iterasi *N* langsung diolah menjadi persyaratan baru untuk iterasi *N+1*. Perubahan ditangani secara bertahap, bukan menunggu di akhir proyek, sehingga risiko pengerjaan ulang (*rework*) besar diminimalkan.

Hide sidebars

3. Kelebihan dan Kekurangan dengan Model lain

Aspek	Kelebihan Spiral	Kekurangan Spiral
Risiko	Manajemen Risiko Unggul (lebih baik dari Waterfall/Prototipe murni)	Biaya dan Waktu bisa lebih besar daripada RAD, karena ada analisis risiko yang ketat
Fleksibilitas	Sangat Adaptif terhadap perubahan persyaratan	Kompleksitas manajemen tinggi, perlu tim yang disiplin.
Rilis	Memungkinkan **Rilis Prototipe/MVP** bertahap.	

Tautan permanen Tampilkan induk

Re: Diskusi.3

oleh AJAY SUPRIADI 04001670 - Selasa, 28 Oktober 2025, 09:29

Tambahkan referensi dan gunkan BMP STSI4202 Rekayasa Perangkat Lunak sebagai acuan dalam menjawab forum diskusi ini.

Tautan permanen Tampilkan induk

Re: Diskusi.3

oleh ELVIRA CITRA PARDENY 051115567 - Minggu, 26 Oktober 2025, 21:26

Menurut saya model SDLC yang paling pas buat proyek startup FoodFast adalah Rapid Application Development (RAD). Soalnya, RAD cocok banget untuk aplikasi yang butuh dikembangkan cepat dan sering mengalami perubahan fitur seperti pendaftaran pengguna dan restoran, pencarian tempat makan, pemesanan dan pembayaran online, pelacakan pesanan secara real-time, sampai sistem ulasan.

Kenapa RAD cocok untuk FoodFast?
• Pengembangan cepat: RAD memungkinkan tim bikin aplikasi dalam waktu singkat lewat proses iteratif dan pembuatan prototipe, jadi FoodFast bisa segera punya versi awal aplikasinya untuk diuji langsung oleh pengguna.
• Mudah menyesuaikan perubahan: Karena prosesnya fleksibel dan berbasis iterasi, tim bisa dengan cepat menambahkan atau mengubah fitur sesuai masukan pengguna.
• Bisa dikerjakan paralel: RAD memecah proyek menjadi beberapa bagian kecil yang bisa dikerjakan bersamaan oleh tim berbeda, jadi pengembangannya lebih efisien.
• Melibatkan pengguna secara langsung: RAD menekankan uji coba dan feedback dari pengguna di setiap versi prototipe, jadi fitur yang dihasilkan lebih sesuai kebutuhan dan risiko gagal lebih kecil.

Cara RAD menangani perubahan fitur
• Dengan prototipe yang terus diperbarui di setiap iterasi singkat (biasanya 2–3 bulan).
• Adaptif terhadap kebutuhan baru, sehingga kalau ada fitur tambahan, bisa langsung diuji dan diimplementasikan.
• Kolaborasi cepat antara pengembang dan pengguna, jadi validasi ide atau fitur baru bisa dilakukan tanpa menunggu lama.

Kalau dibandingkan dengan model SDLC lain seperti Waterfall, Spiral, dan Prototipe, model Rapid Application Development (RAD) punya keunggulan dan kelemahan tersendiri.
Dibanding Waterfall, RAD jauh lebih cepat dan fleksibel. Waterfall berjalan secara berurutan dari tahap analisis, desain, implementasi, sampai pengujian, jadi kalau ada perubahan di tengah jalan, akan sulit dilakukan.

Hide sidebars

Sementara itu, RAD memungkinkan pengembang langsung membuat prototipe dan memperbaruinya sesuai kebutuhan, jadi cocok untuk proyek seperti FoodFast yang butuh respons cepat terhadap masukan pengguna.
Kalau dibandingkan dengan Spiral, keduanya sama-sama fleksibel dan berbasis iterasi, tapi pendekatannya berbeda. Spiral fokus pada analisis risiko di setiap tahap, sehingga lebih cocok untuk proyek besar dan kompleks dengan tingkat ketidakpastian tinggi. Sementara RAD lebih menekankan kecepatan pengembangan dan kolaborasi intens dengan pengguna, jadi hasilnya bisa langsung diuji di dunia nyata tanpa proses analisis yang terlalu panjang.

Sedangkan dibandingkan dengan model Prototipe, RAD punya cakupan yang lebih luas. Model Prototipe biasanya hanya digunakan untuk membuat versi awal atau contoh produk untuk memahami kebutuhan pengguna, tapi belum tentu dikembangkan menjadi produk akhir. RAD, di sisi lain, menggunakan konsep prototipe berulang sebagai bagian dari keseluruhan siklus pengembangan, sehingga hasil akhirnya bisa langsung digunakan sebagai produk jadi.

Kesimpulan
RAD adalah pilihan terbaik untuk FoodFast karena mampu mempercepat proses pengembangan, memudahkan penyesuaian fitur berdasarkan masukan pengguna, dan menekan risiko kegagalan dengan pengujian prototipe yang dilakukan terus-menerus.

Referensi :
- https://jurnal.goretanpena.com/index.php/JSSR/article/viewFile/2809/1610
- https://www.geeksforgeeks.org/software-engineering/software-engineering-rapid-application-development-model-rad
- https://theappsolutions.com/blog/development/rad-model
- Modul STSI424

Tautan permanen Tampilkan induk

Re: Diskusi.3
oleh AJAY SUPRIADI 04001670 - Selasa, 28 Oktober 2025, 09:30

Tambahkan referensi dan gunkan BMP STSI4202 Rekayasa Perangkat Lunak sebagai acuan dalam menjawab forum diskusi ini.

Tautan permanen Tampilkan induk

Re: Diskusi.3
oleh MUHAMMAD HAIDAR ALESSANDRO ABROR 050757557 - Minggu, 26 Oktober 2025, 21:53

Pemilihan Model SDLC untuk FoodFast: Rapid Application Development (RAD)

Model Rapid Application Development (RAD) merupakan pilihan paling tepat untuk proyek startup FoodFast karena menekankan kecepatan pengembangan, fleksibilitas perubahan fitur, dan keterlibatan aktif pengguna dalam setiap tahap iterasi.

Alasan Pemilihan RAD

FoodFast membutuhkan pengembangan aplikasi yang cepat dan mudah disesuaikan karena fitur-fiturnya dinamis—mulai dari registrasi pengguna dan restoran, pemesanan, pembayaran online, hingga pelacakan pesanan real-time. RAD mendukung kebutuhan tersebut melalui:

-Siklus iteratif singkat (2–4 minggu) untuk menguji dan memperbarui fitur secara cepat.
-Prototyping berkelanjutan, memungkinkan versi awal segera diuji oleh pengguna.
-Pengembangan paralel dengan membagi proyek menjadi modul (contoh: sistem pembayaran, pendaftaran) agar efisien.
-Kolaborasi intens dengan pengguna, memastikan fitur sesuai kebutuhan dan mengurangi risiko gagal.

Hide sidebars

Kemampuan RAD Menangani Perubahan

RAD unggul dalam menangani perubahan fitur karena:

-Menggunakan siklus pengembangan pendek dan fleksibel.
-Melibatkan feedback langsung pengguna di setiap iterasi.
-Memiliki komponen modular, sehingga perubahan satu fitur tidak mengganggu fitur lain.
-Dapat menyesuaikan prioritas fitur sesuai urgensi bisnis.

Perbandingan dengan Model Lain

-Waterfall: Kaku dan sulit beradaptasi, sedangkan RAD cepat dan fleksibel.
-Spiral: Cocok untuk proyek besar dan kompleks, tetapi prosesnya lebih lama karena analisis risiko mendalam. RAD lebih efisien untuk aplikasi skala startup seperti FoodFast.
-Prototype: Hanya menghasilkan contoh produk awal, sementara RAD melanjutkan prototipe menjadi produk akhir siap pakai.

Kelebihan dan Kelemahan RAD

Kelebihan:

-Cepat menghasilkan MVP (Minimum Viable Product).
-Responsif terhadap masukan pengguna.
-Efisien melalui penggunaan komponen reusable.

Kelemahan:

-Membutuhkan tim berpengalaman dan kolaboratif.
-Kurang optimal untuk proyek berskala besar.
-Ketergantungan pada komponen yang dapat digunakan ulang.

Kesimpulan

Model RAD paling sesuai untuk FoodFast karena mampu mempercepat peluncuran produk, menyesuaikan fitur berdasarkan umpan balik pengguna, dan mengurangi risiko kegagalan melalui pengujian iteratif yang berkelanjutan. Dengan RAD, FoodFast dapat merilis MVP lebih cepat dan mengembangkannya secara bertahap sesuai kebutuhan pasar.

Referensi:

https://jurnal.goretanpena.com/index.php/JSSR/article/viewFile/2809/1610
https://agus-hermanto.com/blog/detail/metode-pengembangan-rad-rapid-application-development
Modul STSI4202 & STSI424

Tautan permanen Tampilkan induk

Re: Diskusi.3
oleh AJAY SUPRIADI 04001670 - Selasa, 28 Oktober 2025, 09:31

Gunkan BMP STSI4202 Rekayasa Perangkat Lunak sebagai acuan dalam menjawab forum diskusi ini.

Tautan permanen Tampilkan induk

Re: Diskusi.3
oleh HASAN RAPI 053782613 - Senin, 27 Oktober 2025, 07:16

Hide sidebars

Assalamualaikum izin menaggapi diskusinya bapak/ibu tutor.

menurut saya, model prototipe merupakan model yang sangat cocok digunakan dalam proyek pengembangan aplikasi foodfast, alasannya karena model ini memungkinkan pengembang dan pengguna (dalam hal ini restoran dan pelanggan) untuk berinteraksi langsung dengan versi awal sistem , kemudian memberikan umpan balik cepat yang telah dibuat. berdasarkan buku BMP MSIM4303 Modul 3 rekayasa perangkat lunak (Universitas Terbuka) model prototipe sangat efektif digunankan ketika kebutuhan pengguna belum spenuhnya jelas di awal dan memungkinkan terjadinya perubahan secara dinamis.

1. Alasan memilih model waterfall

proyek foodFast sangat cocok dengan model waterfall karena:

- analisis kebutuhan dilakukan secara intens agar kebutuhan perangkat lunak yang di inginkan oleh pelanggan dapat mudah dipahami.
- cocok untuk perusahaan yang butuh panduan kerja terarah
- fitur aplikasi seperti pembiayaan online,pelacakan pesanan, dan rating perlu diuji oleh pengguna sebelum di resmikan.
- mendukung pengembangan iteratif, setiap versi di uji dieavaluasi lalu diperbaiki sesuai masukan pengguna.

2. Cara model prototoipe mengatasi perubahan fitur

model ini bersifat flexibel dan interaktif jika ada perubahan fitur dari hasil uji coba, pengembang bisa menyesuaikan desain tanpa harus mengulang seluruh tahapan seperti di model waterfall.

3. Kelemahan dan kelebihan model prototipe

kelebihan

- Pengguna berperan aktif sejak awal pengembangan
- Mempercepat proses validasi sistem
- mengurangi resiko ketidaksesuaian antara kebutuhan pengguna dan hasil akhir
- cocok untuk proyek yang butuh flexibilitas tinggi dan waktu rilis cepat

Kekurangan

- Membutuhkan komunikasi intens antara pengembang dan pengguna
- memakan waktu yang lama apabila sering terjadi perubahan
- dokumentasi terkadang kurang rapih

Perbandingan prototipe dengan model lain

- Waterfall terlalalu kaku apabila mengalami perubahanfitur di tengah jalan
- RAD, cocok untuk proyek tim besar dan kebutuhan jelas
- Spiral, kuat dalam menajemen resiko akan tetapi lebih kompleks dan mahal untuk starup kecil.

Referensi

- Modul BMP MSIM4303 modul : 3 software development life cycle
- Yuliana, D. (2020). “Penerapan Model Prototyping dalam Pengembangan Sistem Informasi E-Commerce.” *Jurnal Ilmiah Teknologi Informasi Terapan*, 6(2), 45–52.

Tautan permanen Tampilkan induk

Re: Diskusi.3

oleh HASAN RAPI 053782613 - Senin, 27 Oktober 2025, 08:10

Maaf izin koreksi point 1 maksudnya alasan penggunaan model prototipe bukan waterfall

Tautan permanen Tampilkan induk

Re: Diskusi.3

oleh AJAY SUPRIADI 04001670 - Selasa, 28 Oktober 2025, 09:31

Gunakan BMP STSI4202 Rekayasa Perangkat Lunak sebagai acuan dalam menjawab forum diskusi ini.

Tautan permanen Tampilkan induk

Re: Diskusi.3

oleh AJAY SUPRIADI 04001670 - Selasa, 28 Oktober 2025, 09:32

ok

Tautan permanen Tampilkan induk

Hide sidebars

Re: Diskusi.3

oleh VEYSTI DICNORYA BERNANTI 052338378 - Senin, 27 Oktober 2025, 09:36

Selamat Siang
Izin menjawab diskusi

1. Model Prototipe paling sesuai digunakan untuk proyek pengembangan aplikasi FoodFast.
Alasan utama pemilihannya adalah karena model ini menekankan pembangunan cepat versi awal sistem (prototype) untuk diuji langsung oleh pengguna. Melalui umpan balik dari pengguna, sistem kemudian disempurnakan hingga mencapai versi akhir yang siap digunakan.
Karakteristik proyek FoodFast sangat sesuai dengan pendekatan ini karena:
Startup ingin mengembangkan aplikasi secara cepat agar segera dapat digunakan pelanggan.
Beberapa fitur masih perlu diuji terlebih dahulu (seperti pelacakan real-time dan sistem ulasan).
Perubahan fitur akan sering terjadi berdasarkan umpan balik pengguna.
Startup ingin meminimalkan risiko kegagalan karena produk ini menjadi kunci utama bisnis mereka di pasar digital.
Model Prototipe mendukung semua kebutuhan tersebut karena memberikan fleksibilitas tinggi dan iterasi berkelanjutan antara pengembang dan pengguna.

2. Cara Model Prototipe Mengatasi Perubahan Fitur
Model Prototipe memungkinkan pengembang membuat versi awal sistem (mockup atau prototype) yang menampilkan fungsionalitas utama, meskipun belum sempurna. Setelah pengguna mencoba dan memberi masukan, pengembang dapat: Menyesuaikan fitur sesuai kebutuhan pengguna tanpa harus mengubah keseluruhan sistem, Melakukan iterasi cepat antara versi satu dan berikutnya, Menguji ide dan antarmuka pengguna (UI/UX) sebelum implementasi penuh.
Dengan cara ini, perubahan fitur yang sering terjadi tidak menjadi hambatan besar, tetapi justru menjadi bagian dari proses pengembangan yang adaptif.
Pendekatan ini memastikan aplikasi yang dirilis benar-benar sesuai dengan ekspektasi pasar.

3. Kelebihan dan Kekurangan Model Prototipe Dibandingkan Model Lain
Kelebihan Prototipe :
- Dapat menyesuaikan kebutuhan pengguna dengan cepat.
- Mengurangi risiko kesalahan kebutuhan (requirement error).
- Mempercepat waktu pengembangan produk awal.
- Pengguna terlibat langsung dalam setiap tahap pengembangan.
Kekurangan :
- Membutuhkan waktu tambahan untuk iterasi berulang.
- Dokumentasi bisa kurang lengkap.
- Pengguna mungkin salah paham bahwa prototype sudah produk akhir.
Kesesuaian untuk FoodFast :
Sangat cocok, karena perubahan sering terjadi dan butuh hasil cepat yang bisa diuji pengguna.

Hide sidebars

Model SDLC Kelebihan Kekurangan Kesesuaian untuk FoodFast
Prototipe - Dapat menyesuaikan kebutuhan pengguna dengan cepat.
- Mengurangi risiko kesalahan kebutuhan (requirement error).
- Mempercepat waktu pengembangan produk awal.
- Pengguna terlibat langsung dalam setiap tahap pengembangan. - Membutuhkan waktu tambahan untuk iterasi berulang.
- Dokumentasi bisa kurang lengkap.
- Pengguna mungkin salah paham bahwa prototype sudah produk akhir. Sangat cocok, karena perubahan sering terjadi dan butuh hasil cepat yang bisa diuji pengguna.

Kelebihan Waterfall:
- Proses terstruktur dan terdokumentasi dengan baik.
- Cocok untuk proyek dengan kebutuhan yang jelas dan stabil.
Kekurangan Waterfall :
- Tidak fleksibel terhadap perubahan.
- Umpan balik pengguna datang terlambat (setelah tahap akhir).
Kesesuaian untuk FoodFast :
Kurang cocok, karena kebutuhan FoodFast dinamis dan sering berubah.

Kelebihan RAD (Rapid Application Development) :
- Cepat menghasilkan sistem melalui komponen modular.
- Fokus pada pengembangan cepat dengan keterlibatan pengguna

Kekurangan Kelebihan RAD (Rapid Application Development) :
- Membutuhkan tim berpengalaman dan pengguna aktif terlibat.
- Kurang cocok untuk sistem kompleks yang butuh integrasi tinggi.

Kesesuaian untuk FoodFast
Cukup cocok, namun FoodFast membutuhkan fleksibilitas lebih dalam pengujian ide dan fitur.

Kelebihan Spiral :
- Menggabungkan kelebihan prototipe dan waterfall.
- Cocok untuk proyek besar dengan risiko tinggi
Kekurangan Spiral :
- Kompleks dan mahal dalam pengelolaan.
- Membutuhkan perencanaan dan evaluasi risiko mendalam di tiap siklus.
Kesesuaian untuk FoodFast :
Terlalu kompleks untuk startup kecil seperti FoodFast

Sumber Referensi :
Universitas Terbuka. (2025). Buku Materi Pokok STSI4202 – Rekayasa Perangkat Lunak (Edisi 1).
Modul 3: Model Pengembangan Perangkat Lunak
Pressman, R. S. (2015). Software Engineering: A Practitioner's Approach (8th Edition). McGraw-Hill Education.
Sommerville, I. (2016). Software Engineering (10th Edition). Pearson Education Limited.

Tautan permanen Tampilkan induk

Re: Diskusi.3
oleh AJAY SUPRIADI 04001670 - Selasa, 28 Oktober 2025, 09:32

Terima kasih atas penjelasan Anda, analisis terhadap model Prototipe sudah tepat mengingat kebutuhan FoodFast yang dinamis

Tautan permanen Tampilkan induk

Hide sidebars

Re: Diskusi.3

oleh HAMDI 053837115 - Senin, 27 Oktober 2025, 13:45

1. Model SDLC yang paling cocok digunakan untuk proyek FoodFast

Model yang paling cocok untuk proyek ini adalah model Prototipe.

Alasan pemilihan:

Startup ingin mengembangkan aplikasi dengan cepat agar segera digunakan pelanggan. Model Prototipe memungkinkan pembuatan versi awal aplikasi dalam waktu singkat.

Beberapa fitur perlu diuji oleh pengguna sebelum dirilis penuh. Dalam model Prototipe, pengguna dapat langsung memberikan umpan balik terhadap versi awal (mockup atau versi beta).

Perubahan fitur yang sering terjadi dapat diakomodasi dengan mudah karena model ini bersifat iteratif — setiap versi prototipe dapat diperbaiki dan disesuaikan dengan masukan pengguna.

Risiko kegagalan proyek bisa diminimalkan, karena kesalahan dapat ditemukan lebih awal melalui uji coba prototipe sebelum implementasi akhir dilakukan.

2. Cara model Prototipe mengatasi kemungkinan perubahan fitur yang sering terjadi

Model Prototipe menggunakan pendekatan iteratif dan umpan balik berulang.
Langkah-langkahnya:

Tim pengembang membuat versi awal (prototipe) dari aplikasi FoodFast.

Pengguna (baik restoran maupun pelanggan) mencoba aplikasi tersebut.

Mereka memberikan umpan balik mengenai tampilan, alur pemesanan, sistem pembayaran, hingga pelacakan real-time.

Tim pengembang memperbaiki dan menyesuaikan fitur berdasarkan masukan tersebut.

Siklus ini diulang hingga pengguna merasa puas, kemudian barulah sistem final dikembangkan dan diimplementasikan.

Dengan cara ini, perubahan fitur dapat dilakukan secara cepat dan fleksibel tanpa harus mengulang seluruh proses pengembangan seperti pada model Waterfall.

3. Kelebihan dan kekurangan model Prototipe dalam konteks proyek FoodFast
Kelebihan:

Pengembangan cepat dan pengguna bisa segera mencoba versi awal aplikasi.

Umpan balik langsung dari pengguna membantu tim memahami kebutuhan nyata di lapangan.

Mengurangi risiko kegagalan proyek, karena kesalahan dapat diperbaiki sejak tahap awal.

Fleksibel terhadap perubahan fitur, sesuai kebutuhan startup dan respon pasar.

Kekurangan:

Membutuhkan komitmen waktu dan komunikasi intensif antara tim pengembang dan pengguna.

Kadang pengguna menganggap prototipe sebagai produk akhir, padahal masih perlu penyempurnaan.

Dokumentasi teknis sering diabaikan karena fokus pada iterasi cepat.

Jika tidak dikontrol dengan baik, biaya dan waktu bisa meningkat akibat banyak revisi.

Sumber: BMP STSI4202 Rekayasa Perangkat Lunak

Tautan permanen Tampilkan induk

Hide sidebars

Re: Diskusi.3

oleh AJAY SUPRIADI 04001670 - Selasa, 28 Oktober 2025, 09:33

Anda telah memahami inti persoalan dengan baik. Model Prototipe memang fleksibel untuk perubahan fitur

Tautan permanen Tampilkan induk

M

Re: Diskusi.3

oleh MUHAMMAD FARHAN 051501474 - Senin, 27 Oktober 2025, 16:23

Ijin menanggapi diskusi.
1. Model SDLC yang Paling Cocok
Model Prototipe (Prototyping Model) merupakan model yang paling cocok digunakan untuk proyek pengembangan aplikasi FoodFast. Model ini menekankan pada pembuatan versi awal atau model sederhana dari perangkat lunak yang akan dikembangkan. Prototipe berfungsi untuk menampilkan sebagian fungsionalitas utama sistem agar pengguna dapat mencoba dan mengevaluasi produk sebelum versi final selesai dibangun. Dengan cara ini, pengembang dan pengguna dapat memahami kebutuhan secara lebih jelas sejak tahap awal pengembangan.
2. Alasan Pemilihan Model Prototipe
Ada beberapa alasan mengapa model Prototipe sangat sesuai untuk proyek FoodFast:
a. Pengembangan yang cepat. Startup memerlukan aplikasi yang segera dapat digunakan oleh pelanggan. Model Prototipe memungkinkan pembuatan versi awal (MVP) dengan fitur utama seperti pendaftaran pengguna, pencarian restoran, dan pemesanan makanan dalam waktu singkat.
b. Uji coba langsung oleh pengguna. Prototipe memungkinkan pelanggan dan restoran mencoba sistem lebih awal sehingga pengembang mendapatkan umpan balik nyata.
c. Fleksibilitas terhadap perubahan. Karena pengembangan dilakukan secara bertahap, model ini dapat menyesuaikan perubahan kebutuhan dan fitur baru tanpa mengulang seluruh proses dari awal.
d. Mengurangi risiko kegagalan. Melalui siklus umpan balik berkelanjutan, sistem dapat diperbaiki sebelum diluncurkan ke pasar, sehingga kemungkinan kegagalan proyek berkurang.
3. Cara Model Prototipe Mengatasi Perubahan Fitur yang Sering Terjadi
Model Prototipe bersifat iteratif, artinya proses pengembangannya dilakukan berulang kali. Setiap kali pengguna memberikan umpan balik, pengembang dapat segera memperbarui prototipe yang sudah ada tanpa harus membangun ulang seluruh sistem. Tahapan umumnya meliputi:
Identifikasi kebutuhan dasar untuk menentukan fitur utama yang harus ada.
Pembuatan prototipe awal untuk menampilkan tampilan dan fungsi sederhana.
Uji coba dan evaluasi pengguna, di mana pengguna memberikan saran dan masukan.
Revisi dan penyempurnaan berdasarkan umpan balik tersebut.
Siklus ini terus berulang hingga sistem memenuhi harapan pengguna. Dengan demikian, perubahan fitur yang sering terjadi justru menjadi bagian alami dari proses pengembangan, bukan hambatan.
4. Kelebihan Model Prototipe dalam Konteks Proyek FoodFast
Model Prototipe memiliki sejumlah kelebihan penting yang relevan dengan kebutuhan startup:
a. Keterlibatan pengguna tinggi. Pengguna dapat berpartisipasi sejak awal, sehingga sistem yang dibangun lebih sesuai dengan kebutuhan mereka.
b. Pemahaman kebutuhan lebih baik. Karena sistem divisualisasikan melalui prototipe, pengembang dan pengguna memiliki pemahaman yang sama tentang arah pengembangan.

Hide sidebars

c. Waktu dan biaya efisien. Kesalahan dapat ditemukan lebih awal sehingga perbaikan dapat dilakukan sebelum masuk ke tahap implementasi besar.
d. Umpan balik cepat. Setiap iterasi menghasilkan masukan langsung yang dapat meningkatkan kualitas produk.
e. Mengurangi risiko kegagalan. Sistem diuji berulang kali sebelum dirilis, sehingga lebih stabil dan sesuai harapan.
5. Kekurangan Model Prototipe dalam Konteks Proyek FoodFast
Meskipun efektif, model ini juga memiliki beberapa kelemahan:
a. Analisis kebutuhan bisa kurang mendalam, karena fokus terlalu besar pada tampilan prototipe.
b. Potensi kesalahpahaman pengguna, yang mungkin mengira prototipe adalah produk final padahal masih versi uji coba.
c. Risiko perluasan proyek (scope creep), yaitu terlalu banyak revisi dan tambahan fitur tanpa batas waktu yang jelas.
d. Kemungkinan penyalahgunaan prototipe, misalnya pengembang tergoda menggunakan kode sementara dari prototipe untuk sistem akhir yang seharusnya dibangun ulang secara terstruktur.
e. Waktu dan biaya bisa meningkat bila revisi dilakukan terlalu sering tanpa pengendalian.
6. Perbandingan dengan Model SDLC Lainnya
a. Model Waterfall tidak cocok karena prosesnya bersifat kaku dan tidak mendukung perubahan kebutuhan di tengah jalan. Sekali tahap selesai, sulit untuk kembali dan memperbaikinya.
b. Model RAD (Rapid Application Development) memang cepat, tetapi memerlukan sumber daya besar serta keterlibatan pengguna yang intens, yang mungkin sulit bagi startup kecil.
c. Model Spiral cocok untuk proyek besar dengan tingkat risiko tinggi, namun terlalu kompleks dan mahal untuk startup seperti FoodFast.
Dengan demikian, dibandingkan model-model lain, Prototipe menawarkan keseimbangan terbaik antara kecepatan, fleksibilitas, dan efisiensi sumber daya.

Sumber referensi: https://www.tutorialspoint.com/sdlc/sdlc_software_prototyping

Tautan permanen Tampilkan induk

Re: Diskusi.3
oleh AJAY SUPRIADI 04001670 - Selasa, 28 Oktober 2025, 09:34

Gunakan BMP STSI4202 Rekayasa Perangkat Lunak sebagai acuan dalam menjawab forum diskusi ini.

Tautan permanen Tampilkan induk

◀ Materi Inisiasi

Lompat ke...

Tugas.1 ▶

Navigasi

- Dasbor
  - Beranda situs
  - Laman situs
  - Kelasku
    - STSI4203.108
    - STSI4202.42
      - Peserta
      - Nilai
      - Pendahuluan
      - Sesi 1
      - Sesi 2
      - Sesi 3

Kehadiran Sesi ke-3
Materi Inisiasi
Diskusi.3
Tugas.1
Sesi 4
STSI4103.119
MKKI4201.278
STSI4201.161
STSI4205.331
STSI4104.284
MKDI4202.1514
Kelas

Hide sidebars

Administrasi

Forum administrasi
Berlangganan dinonaktifkan

Follow Us:

Anda masuk sebagai INDRAWAN LISANTO 053724113 (Keluar)
Dapatkan aplikasi seluler