

Rekayasa Perangkat Lunak

PROTOTYPE & SPIRAL

Kelompok 1



ANDRE PUTRA DEWANSYAH
2311523001



VANIA ZHAFIRA ZAHRA
2311523007



IKHWAN HAMIDI
2311521003



RIDHO DWI SYAHPUTRA
2311522033



GHINA TSARY ANANTA
2311523037



FATHIYA AL ZHAFIRA
2311523009



NAISYA HARJIMEN
2311523025



VELISA DWI SONIA
2311522039



PROTOTYPE

Konsep PROTOTYPE

Prototype dibuat untuk kebutuhan awal *development software* dan untuk mengetahui apakah fitur dan fungsi dalam program berjalan sesuai dengan kebutuhan yang telah direncanakan. Sehingga pengembang produk dapat mengetahui kekurangan dan kesalahan lebih awal sebelum mengimplementasikan fitur lain ke dalam produk dan merilis produk.

Prototipe bertujuan untuk mengembangkan rancangan awal menjadi produk akhir yang sesuai dengan kebutuhan pengguna. Dalam prosesnya, pengguna berperan dengan memberikan evaluasi dan umpan balik, yang kemudian dijadikan acuan untuk penyempurnaan.

Selain itu, prototipe juga dapat memunculkan ide baru yang dapat dikembangkan menjadi fitur tambahan.



Konsep **PROTOTYPE**

Dalam desain UI/UX, terdapat beberapa jenis prototipe yang digunakan untuk menguji pengalaman pengguna:

01 **Low-fidelity Prototype**

Prototipe sederhana dengan detail rendah, biasanya berupa sketsa atau wireframe, digunakan untuk menyampaikan konsep awal dan mengumpulkan umpan balik dengan cepat.

02 **High-fidelity Prototype**

Prototipe dengan elemen visual lebih realistis, dibuat menggunakan alat seperti Figma atau Adobe XD, memungkinkan pengujian mendalam terhadap desain dan interaksi produk.



Konsep **PROTOTYPE**

Dalam desain UI/UX, terdapat beberapa jenis prototipe yang digunakan untuk menguji pengalaman pengguna:

03 Interactive Prototype

Prototipe yang memungkinkan pengguna berinteraksi dengan elemen UI, seperti navigasi dan animasi, untuk menguji alur pengguna dan pengalaman interaktif.

04 Responsive Prototype

Prototipe yang menyesuaikan tampilan dan fungsi di berbagai perangkat, memastikan pengalaman yang konsisten pada mobile, tablet, dan desktop.



Konsep **PROTOTYPE**

Dalam desain UI/UX, terdapat beberapa jenis prototipe yang digunakan untuk menguji pengalaman pengguna:

05 **Simulative Prototype**

Prototipe yang mensimulasikan interaksi khusus menggunakan perangkat lunak, berguna untuk menguji fitur unik atau pengalaman pengguna tertentu.



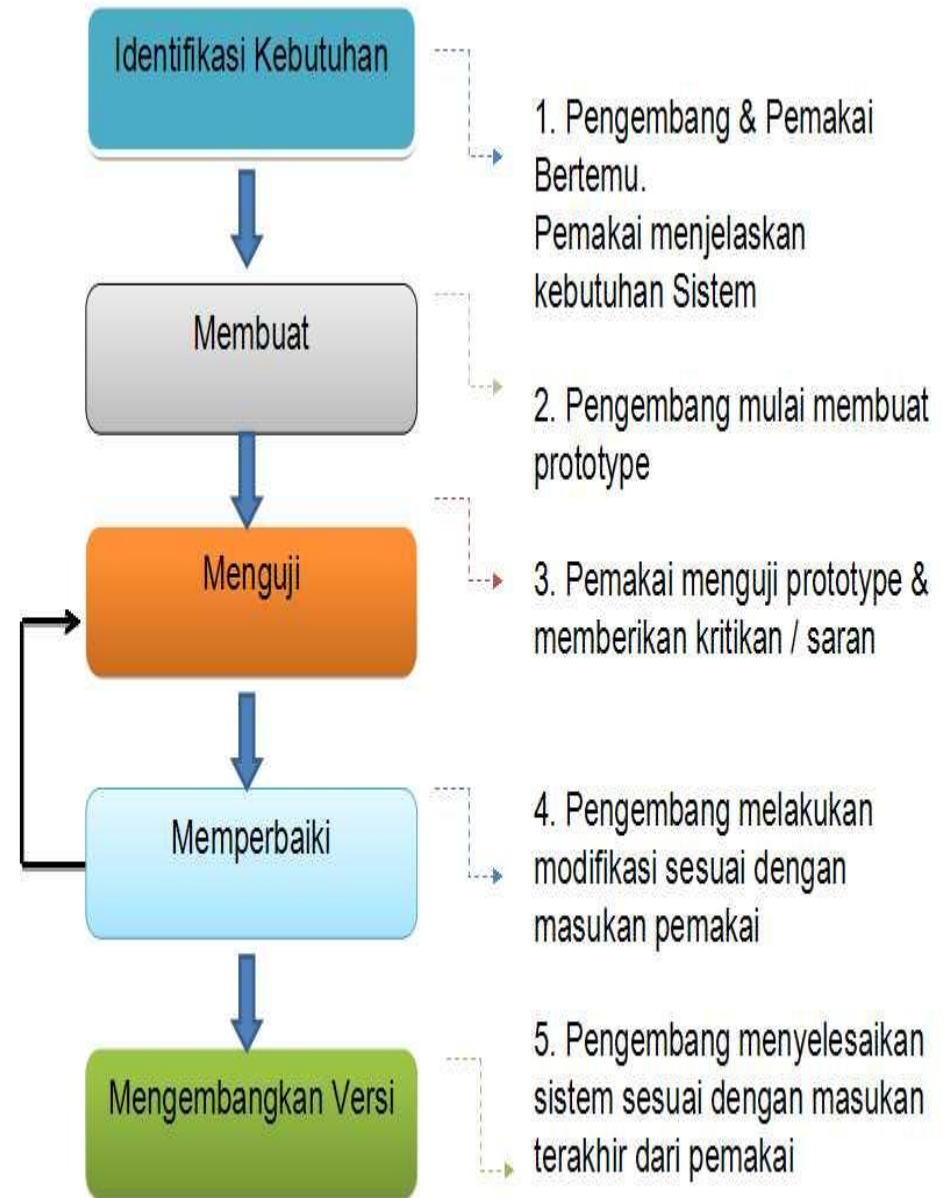
Tahapan Prototype

01 IDENTIFIKASI KEBUTUHAN PEMAKAI

Pengembang dan pelanggan bersama-sama mendefinisikan format perangkat lunak, mengidentifikasi kebutuhan, serta merancang garis besar sistem yang akan dibuat.

02 MEMBANGUN PROTOTIPE

Membuat rancangan awal yang berfokus pada penyajian kepada pelanggan, seperti desain input dan format output.



Evaluasi Prototipe

03

Pelanggan menilai apakah prototipe yang dibuat sudah sesuai dengan kebutuhan mereka. Pada tahap ini, kritik dan saran dikumpulkan untuk perbaikan.

Prototipe yang telah disepakati diterjemahkan ke dalam bahasa pemrograman yang sesuai untuk dikembangkan menjadi sistem yang fungsional.

04

Mengkodekan Sistem

Pengujian Sistem

05

Sistem yang telah dikembangkan diuji menggunakan berbagai metode, seperti white-box testing, black-box testing, dan pengujian arsitektur, untuk memastikan kualitasnya sebelum digunakan.

Pelanggan menilai apakah sistem yang telah selesai sesuai dengan harapan. Jika belum, maka dilakukan perbaikan dengan mengulangi proses pengkodean dan pengujian.

06

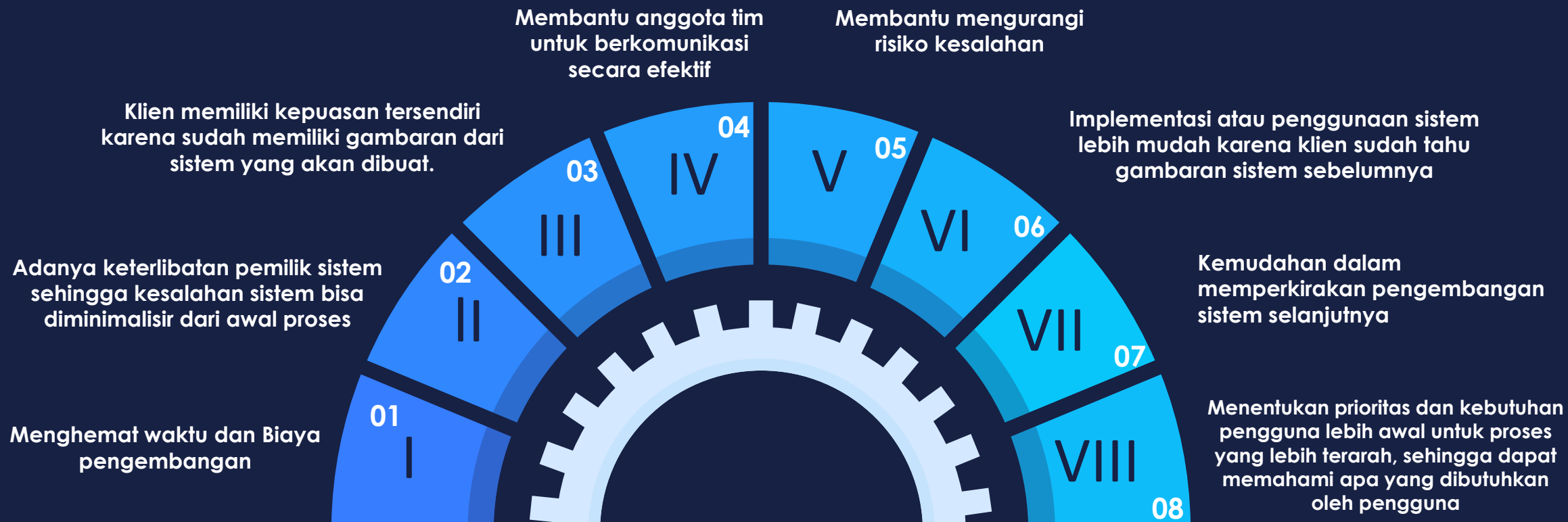
Evaluasi Sistem

Evaluasi Prototipe

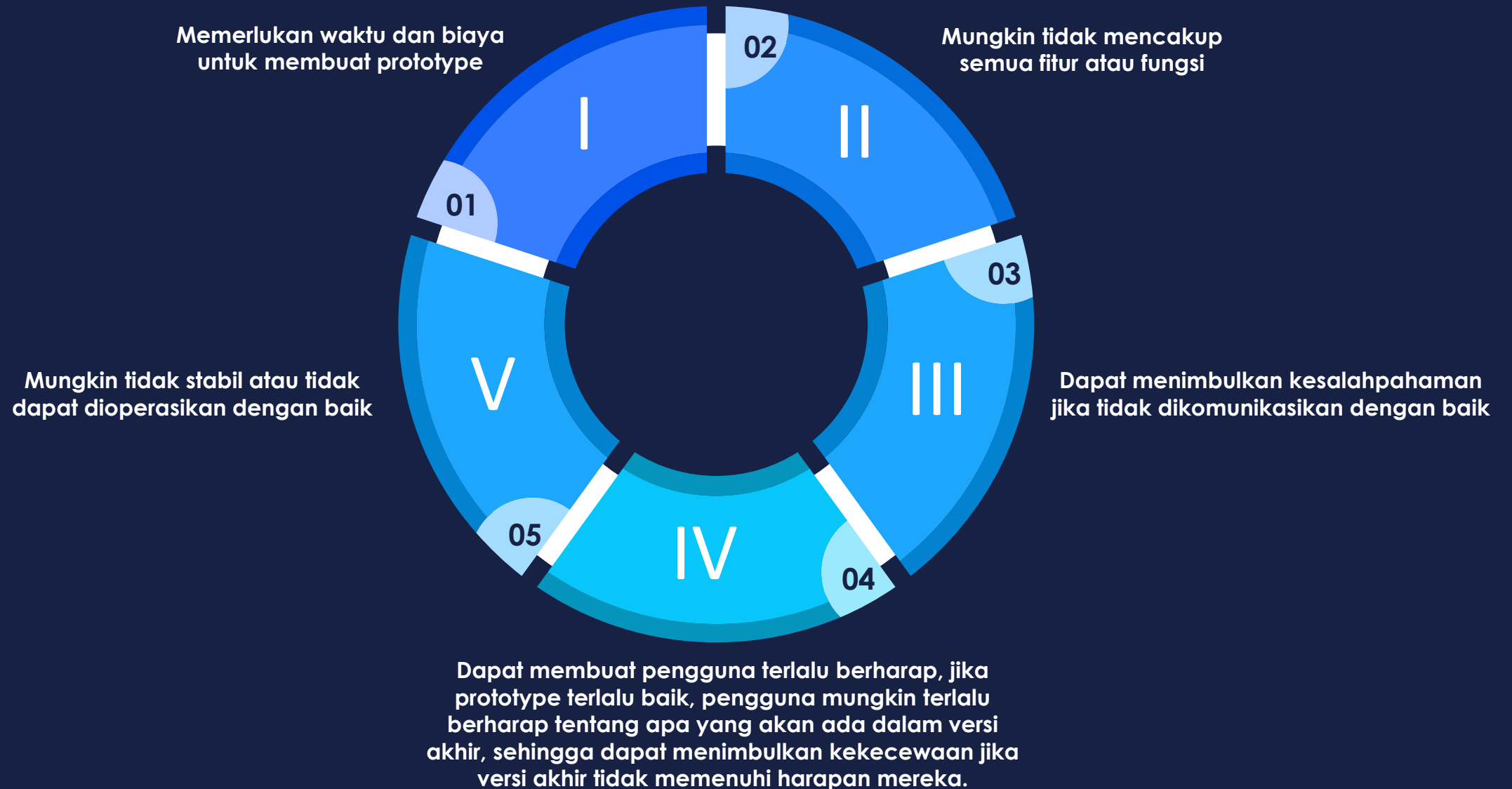
07

Setelah diuji dan disetujui oleh pelanggan, perangkat lunak siap digunakan.

Kelebihan Prototype



REKAYASA PERANGKAT LUNAK



The logo features a dark blue circular background. In the center, there is a stylized, thick blue swirl that resembles a spiral. Overlaid on this swirl is the word "SPIRAL" in a bold, white, sans-serif font. Surrounding the central swirl and text are several thin, light blue curved lines that suggest a path or orbit, with small light blue dots at their endpoints.

SPIRAL

konsep SPIRAL



Model Spiral diperkenalkan pertama kali oleh **Barry Boehm** pada makalahnya yang berjudul **Spiral Model of Software Development and Enhancement**.

Model spiral merupakan **penggabungan** dari model **prototyping** dan model **waterfall**.

Model spiral menekankan pada Analisa resiko setiap tahapannya.

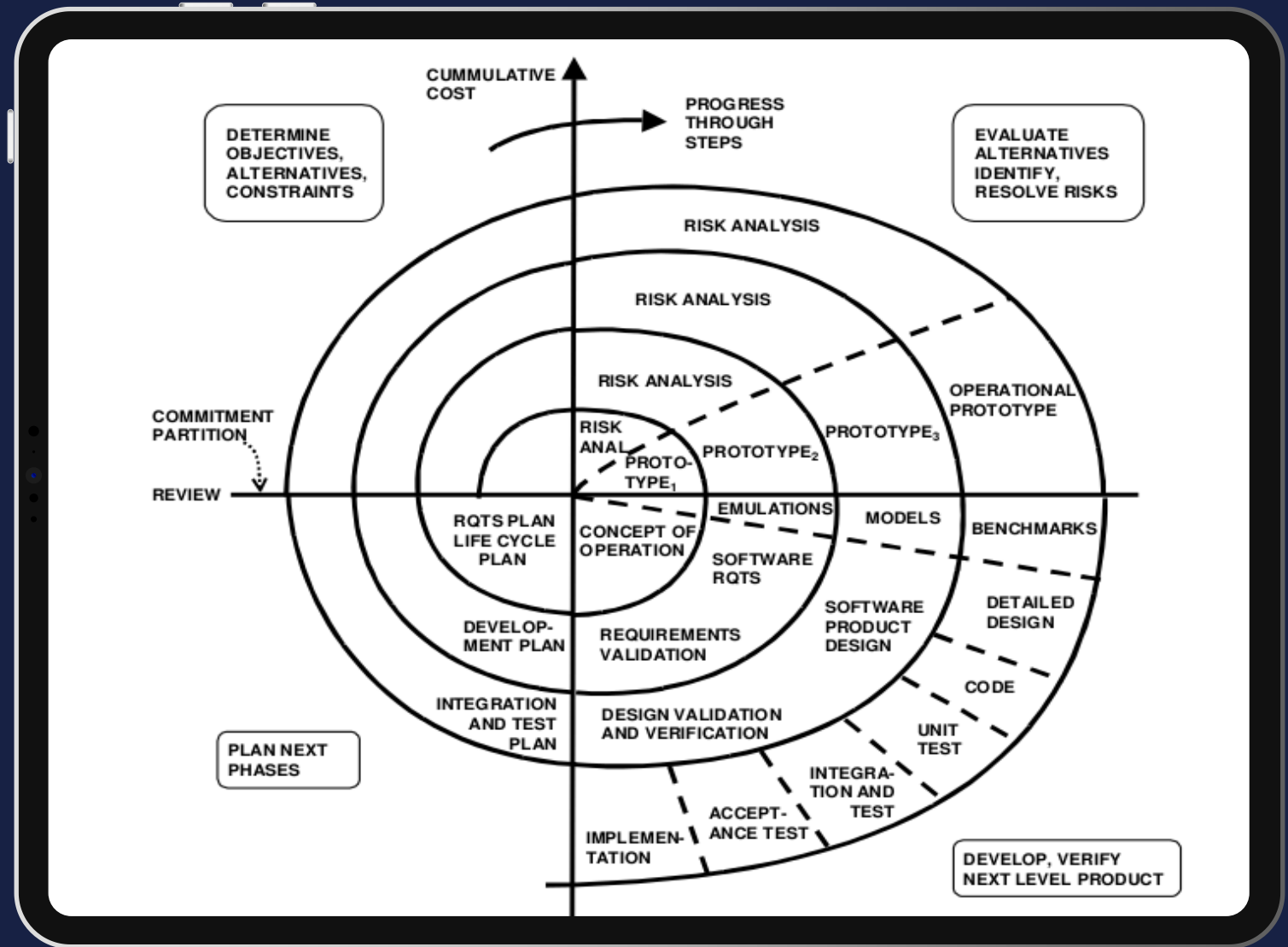
Fungsi model spiral adalah untuk melakukan perubahan, penambahan dan pengembangan perangkat lunak dengan memaksimalkan aspek kecepatan dan ketepatan berdasarkan keinginan dan kebutuhan penggunaanya.

Tahapan Model **SPIRAL**

01 Tahap Liason

Tahap ini berhubungan dengan **komunikasi** antara **pengembang software** dengan **pelanggan**.

Tujuannya adalah **memperbaiki** dan **mengembangkan** software sesuai **kebutuhan** dan keinginan hingga memuaskan pelanggan.



SPIRAL

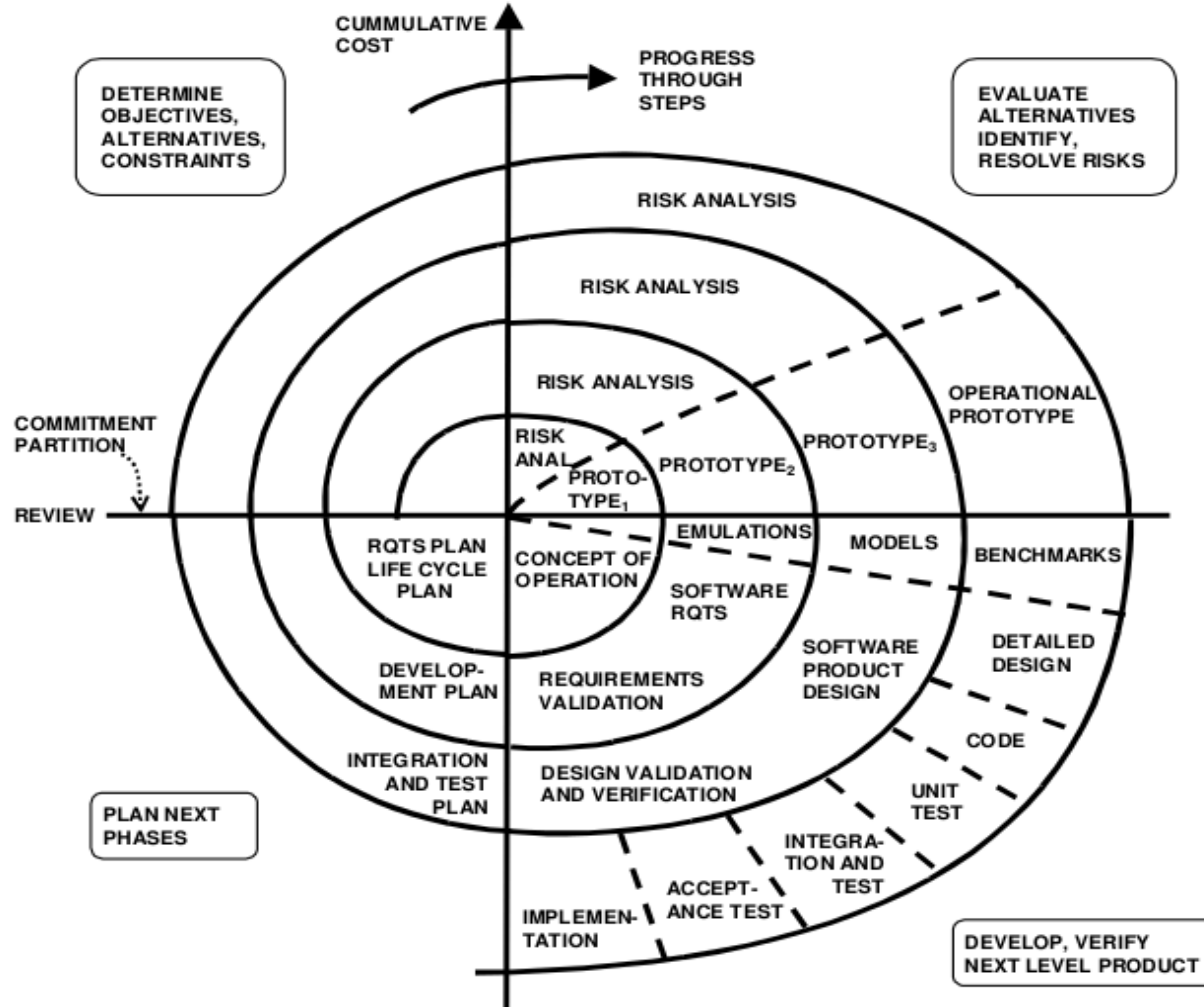
02

Tahap perencanaan meliputi estimasi biaya yang digunakan, batas waktu, pengaturan jadwal, dan lainnya.

Hasil dari tahapan ini adalah **dokumen spesifikasi kebutuhan sistem dan bisnis.**

03

Tahap analisis resiko berfungsi untuk **mengidentifikasi resiko** yang berpotensi akan terjadi dan **menghasilkan solusi alternatif** secara teknis dan manajemen saat strategi mitigasi direncanakan dan diselesaikan.



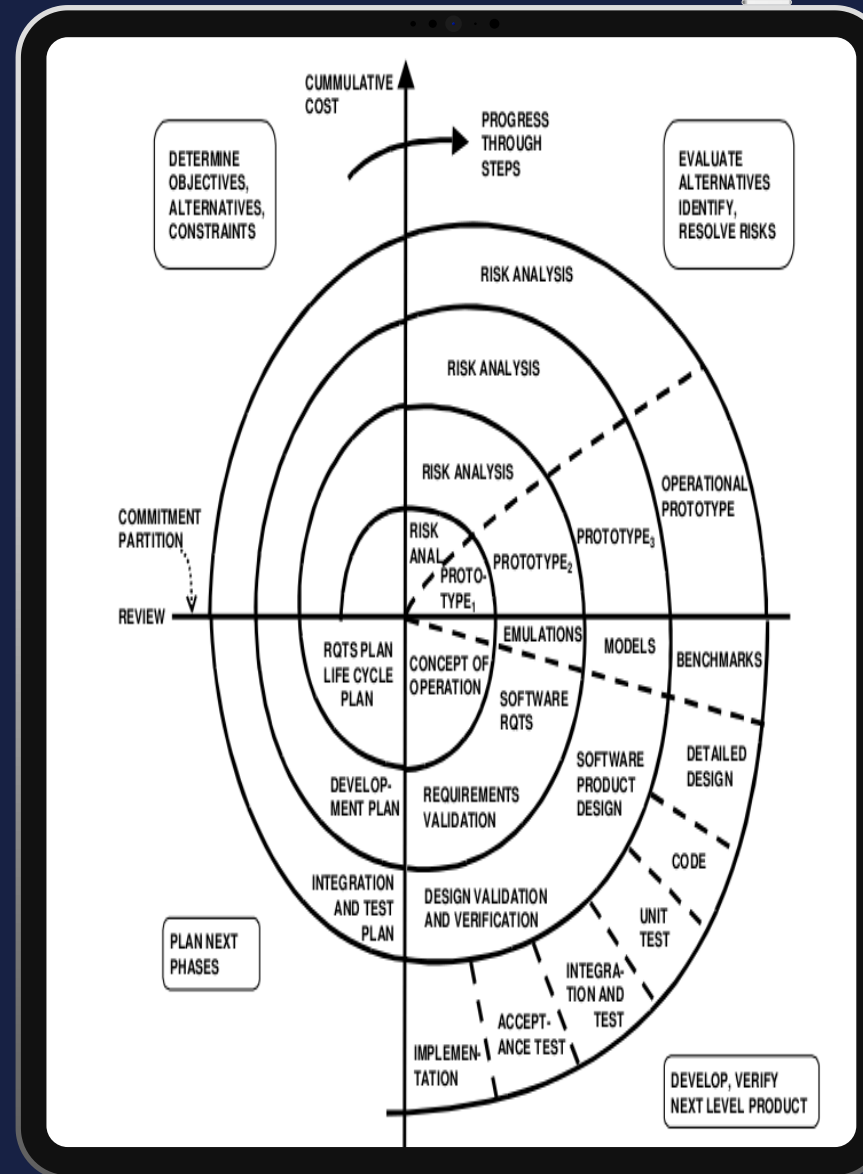
Tahapan Model SPIRAL

04

Tahap Rekayasa (Engineering)

Pada tahap rekayasa, beberapa kegiatan ini yang akan dilakukan, yaitu:

- Menguji, coding dan mengembangkan software
- Menginstal software
- Membuat prototype
- Mendesain dokumen
- Meringkas suatu pengujian software
- Membuat laporan atas kekurangan dari software agar segera diperbaiki



05

Tahap Evaluasi

Pada tahap evaluasi, system analyst membutuhkan masukan dan tanggapan dari para user dalam mengevaluasi perangkat/produk yang diuji dan memastikan bahwa produk dibutuhkan sesuai ketentuan yang telah dibicarakan diawal dengan user.

Selain itu, system analyst harus tetap memantau resiko yang akan terjadi seperti faktor-faktor yang dapat menyebabkan cost overrun (pembengkakan biaya).



Kelebihan Metode SPIRAL

- 01.** Pembangunan dan perubahan perangkat lunak yang terjadi dapat diselesaikan secara sistematis

Model Spiral menggunakan pendekatan bertahap dan berulang sehingga pengembang dapat menyelesaikan proyek dengan lebih terstruktur. Setiap iterasi memastikan semua aspek sistem dikembangkan dengan baik sebelum ke tahap berikutnya.
- 02.** Mudah dalam mengestimasi biaya karena proses pembuatan prototype yang jelas dan terencana dalam tahapan yang sistematis

Model Spiral membagi pengembangan menjadi fase-fase yang terencana, sehingga biaya dapat dihitung lebih akurat dan tim pengembang dapat mengidentifikasi kebutuhan tambahan lebih awal.
- 03.** Manajemen dan analisa risiko yang lebih cepat dan mudah

Setiap iterasi mencakup analisis risiko yang membantu dalam mendeteksi dan menangani potensi masalah. Proyek menjadi lebih aman dan memiliki peluang keberhasilan yang lebih tinggi.



Kelebihan Metode SPIRAL

04.

Mudah dalam melakukan perubahan kebutuhan dan dokumentasi

Model ini berbasis iterasi, sehingga perubahan kebutuhan dapat dilakukan dengan lebih fleksibel tanpa mengganggu seluruh proses pengembangan. Setiap tahap ada dokumentasi yang jelas, yang berguna bagi pengembang dan pemangku kepentingan dalam memahami perkembangan proyek.

05.

Produksi software bisa terjadi lebih cepat

Adanya iterasi memungkinkan pengujian dan evaluasi di setiap tahap, produk dapat dikembangkan dan dirilis lebih cepat yang memungkinkan pengguna mendapatkan manfaat dari perangkat lunak sebelum seluruh sistem selesai.



Kekurangan Metode **SPIRAL**

Resiko dalam tahap planning cukup besar.

Hal ini dikarenakan setiap iterasi dapat menghasilkan perubahan dalam kebutuhan proyek, ada kemungkinan jadwal pengembangan dan anggaran belanja mengalami penyimpangan.

Tidak cocok dan sulit diimplementasikan dalam proyek kecil

Proyek kecil dengan sumber daya terbatas akan kesulitan menerapkan model ini karena membutuhkan banyak iterasi dan tahapan evaluasi risiko. Model ini cocok untuk proyek besar yang punya banyak risiko dan kompleksitas.

Membutuhkan best practice sebelumnya karena proses yang sangat kompleks

Model ini tidak mudah diterapkan jika belum berpengalaman dalam manajemen risiko dan pengembangan perangkat lunak berbasis iterasi karena membutuhkan keahlian dalam mengidentifikasi dan menangani risiko secara efektif agar model ini berjalan dengan baik.


Projects Memakan waktu yang cukup lama

Setiap siklus dalam Model Spiral melibatkan analisis, perencanaan, pengembangan, dan evaluasi, yang membuat prosesnya lebih panjang dibandingkan model lain.



Soal Latihan 1

Sarankan model proses perangkat lunak generik yang paling tepat yang dapat digunakan sebagai dasar untuk mengelola pengembangan Sistem Informasi Akademik Universitas Andalas (sertakan penjelasannya dan asumsi yang dibutuhkan).




Dalam pengembangan Sistem Informasi Akademik (SIA) di sebuah universitas, tim pengembang menghadapi tantangan besar karena kebutuhan sistem yang terus berubah. Regulasi akademik dan kebijakan universitas sering diperbarui, serta masukan dari berbagai pengguna, seperti mahasiswa, dosen, dan tenaga administrasi, harus dipertimbangkan agar sistem dapat benar-benar memenuhi kebutuhan mereka.

Untuk mengatasi kompleksitas ini, tim pengembang memilih menggunakan Model Spiral dalam pengembangan perangkat lunak. Model ini memungkinkan pengembangan sistem secara bertahap, dengan setiap iterasi mencakup perencanaan, analisis risiko, pengembangan, pengujian, dan evaluasi oleh pengguna. Dengan pendekatan ini, tim dapat mengidentifikasi dan menyesuaikan fitur berdasarkan perubahan kebijakan dan kebutuhan akademik sebelum implementasi penuh dilakukan.

Pada tahap awal, tim melakukan komunikasi intensif dengan pihak universitas untuk memahami kebutuhan dasar sistem. Setelah itu, mereka membuat prototipe awal yang mencakup fitur dasar seperti pengelolaan data mahasiswa, pendaftaran mata kuliah, dan sistem penjadwalan. Prototipe ini kemudian diuji oleh pengguna untuk mendapatkan umpan balik yang berguna bagi iterasi berikutnya.

Seiring waktu, dengan adanya perubahan kebijakan akademik atau kebutuhan baru dari pengguna, sistem dapat diperbarui tanpa harus membangun ulang dari awal. Analisis risiko dalam setiap iterasi membantu mengidentifikasi potensi masalah, seperti keamanan data akademik dan keandalan sistem, sehingga tindakan pencegahan dapat diterapkan sebelum masalah menjadi lebih besar.



Pendekatan ini juga meminimalkan pemborosan anggaran dan waktu karena setiap pengembangan diuji dan divalidasi secara bertahap. Selain itu, dengan melibatkan pengguna dalam evaluasi sistem di setiap iterasi, kepuasan pengguna dapat meningkat, karena mereka merasa sistem benar-benar dirancang sesuai dengan kebutuhan mereka.

Dibandingkan dengan model pengembangan lainnya, Model Spiral lebih cocok untuk proyek ini karena sifatnya yang fleksibel dan berbasis evaluasi berulang. Dengan model ini, tim dapat lebih mudah menyesuaikan sistem dengan dinamika kebijakan akademik dan memastikan bahwa SIA yang dikembangkan benar-benar andal, aman, dan sesuai dengan kebutuhan stakeholder.



Analisis Kebutuhan dan Karakteristik Proyek :

- Sistem yang dikembangkan harus fleksibel untuk beradaptasi dengan perubahan regulasi akademik dan kebijakan universitas. Dengan pengguna yang beragam mencakup mahasiswa, dosen, dan staf administrasi, diperlukan pendekatan iteratif agar fitur yang dikembangkan benar-benar sesuai kebutuhan.
- Keamanan dan keandalan sangat krusial karena sistem menangani data akademik dan administrasi yang sensitif. Pengujian keamanan harus dilakukan di setiap tahap pengembangan untuk mencegah kebocoran data.
- Pengembangan bertahap diperlukan agar fitur dapat diuji dan diperbaiki sebelum implementasi penuh, mengurangi risiko kesalahan besar. Manajemen risiko juga penting untuk mengantisipasi pembengkakan biaya, perubahan regulasi, dan ancaman keamanan.
- Sistem harus skalabel agar dapat berkembang dan menangani peningkatan jumlah pengguna serta data di masa depan. Dengan kompleksitas proyek ini, Model Spiral menjadi pilihan tepat karena memungkinkan evaluasi risiko berkala, penyesuaian fleksibel, dan pengembangan yang lebih terstruktur.



Asumsi yang digunakan dalam pemilihan model :

- **Fleksibilitas.** Dalam menghadapi perubahan-perubahan regulasi akademik dan kebijakan universitas, sistem yang dikembangkan harus fleksibel. Dengan Model Spiral, memungkinkan sistem dapat berkembang tanpa perlu merombak seluruh pengembangan dari awal dengan penyesuaian di setiap iterasi.
- **Pengujian keamanan di setiap tahap.** Sistem ini menangani data akademik dan administrasi yang bersifat rahasia, seperti nilai, data pribadi mahasiswa, dan transaksi keuangan. Model Spiral menekankan identifikasi risiko di setiap siklus, termasuk risiko keamanan, sehingga pengujian keamanan dapat dilakukan secara berkala.
- **Pengembangan bertahap.** Pengembangan sistem perlu dilakukan secara bertahap agar setiap fitur dapat diuji dan diperbaiki sebelum implementasi penuh. Dengan iterasi yang berulang dan evaluasi berkala dari Model Spiral, dampak dari kesalahan dapat diminimalkan karena hanya mempengaruhi bagian tertentu dari sistem, bukan keseluruhan proyek.
- **Manajemen risiko.** Risiko utama dalam proyek ini mencakup perubahan regulasi, pembengkakan biaya, dan ancaman keamanan. Model Spiral memiliki fase eksplisit untuk analisis risiko di setiap iterasi, sehingga potensi masalah dapat diidentifikasi dan diatasi sebelum berkembang menjadi kegagalan besar.
- **Skalabilitas.** Pertumbuhan jumlah mahasiswa, dosen, dan data akademik sangat memungkinkan terjadi di masa depan, sehingga sistem harus mampu menangani hal tersebut. Model Spiral memungkinkan pengembangan modular, sehingga sistem dapat diperluas tanpa mengganggu fungsi yang sudah berjalan.

Kesimpulan




Dengan demikian, pemilihan Model Spiral sebagai pendekatan dalam pengembangan Sistem Informasi Akademik ini didasarkan pada karakteristik proyek yang membutuhkan iterasi berulang, fleksibilitas terhadap perubahan kebutuhan, serta manajemen risiko yang ketat demi memastikan sistem yang dihasilkan benar-benar optimal.



Soal Latihan 2

Bayangkan bahwa pemerintah menginginkan program perangkat lunak yang membantu melacak pemanfaatan sumber daya mineral negara yang sangat besar. Meskipun persyaratan yang diajukan oleh pemerintah tidak terlalu jelas, sebuah perusahaan perangkat lunak ditugaskan untuk mengembangkan software melalui model prototipe. Pemerintah menganggap prototipe yang dihasilkan mengesankan, dan memintanya untuk dikembangkan menjadi sistem aktual yang akan digunakan. Diskusikan pro dan kontra dari mengambil pendekatan ini.



Pemerintah ingin membangun sistem untuk melacak pemanfaatan sumber daya mineral negara, tetapi mereka sendiri belum punya gambaran jelas tentang seperti apa sistem itu harusnya bekerja.

Untuk mengatasi ketidakjelasan ini, tim pengembang perangkat lunak membuat prototipe terlebih dahulu. Prototipe ini berfungsi sebagai versi awal yang memungkinkan pemerintah melihat dan mencoba fitur-fitur seperti pemantauan tambang, pelacakan distribusi, dan pembuatan laporan otomatis.

Setelah melihat prototipe, pemerintah merasa sistem ini menjanjikan dan meminta agar dikembangkan menjadi sistem penuh. Dengan pendekatan ini, mereka bisa memberikan umpan balik sejak awal, meminta penyesuaian jika diperlukan, dan memastikan bahwa sistem yang dibangun benar-benar bermanfaat.

Dibandingkan langsung membangun sistem tanpa kejelasan, pendekatan prototipe ini jauh lebih aman dan efisien. Kesalahan bisa diperbaiki lebih awal, waktu pengembangan lebih cepat, dan dana yang dikeluarkan tidak terbuang sia-sia. Inilah alasan mengapa pendekatan ini sering digunakan dalam proyek besar dengan kebutuhan yang kompleks.

Pendekatan prototipe dalam pengembangan perangkat lunak berarti membuat model awal dari sistem sebelum benar-benar membangunnya secara penuh. Cara ini sangat berguna ketika kebutuhan proyek belum jelas karena memungkinkan pengguna mencoba versi awal dan memberikan masukan sebelum sistem yang sebenarnya dikembangkan. Beberapa keuntungan dari metode ini antara lain:



1. Memperjelas Kebutuhan yang Masih Kabur

Dengan prototipe, pengguna bisa melihat gambaran awal sistem dan memberikan umpan balik, sehingga tim pengembang dapat memahami dengan lebih baik apa yang benar-benar dibutuhkan.

2. Mengurangi Risiko Kesalahan Besar

Kesalahan desain bisa dideteksi sejak awal, sehingga lebih mudah dan murah untuk diperbaiki dibandingkan jika ditemukan di tahap akhir.

3. Meningkatkan Kepuasan Pengguna

Karena pengguna dilibatkan sejak awal, mereka merasa lebih memiliki sistem tersebut, dan hasil akhirnya pun lebih sesuai dengan kebutuhan mereka.

4. Mempercepat Proses Pengembangan

Daripada hanya mengandalkan dokumen perencanaan yang panjang, pengembang bisa langsung membuat sesuatu yang bisa diuji dan diperbaiki secara bertahap.

5. Mengoptimalkan Anggaran

Dengan fokus hanya pada fitur yang benar-benar penting, biaya pengembangan dapat dikelola dengan lebih efisien.



Analisis Keuntungan pendekatan prototipe dalam proyek ini

- Karena pemerintah belum memiliki gambaran yang jelas tentang sistem yang diinginkan, penggunaan prototipe memungkinkan pengembang untuk membuat versi awal yang bisa diuji. Dimulai dari sini, pemerintah bisa memberikan masukan untuk menyempurnakan sistem sesuai kebutuhan mereka.
- Dengan adanya prototipe, pemerintah dapat langsung mencoba sistem dalam tahap awal. Hal ini membantu memahami cara kerja perangkat lunak dan memberikan saran perbaikan sebelum sistem benar-benar dikembangkan secara keseluruhan. Kemudian hasil akhirnya akan lebih sesuai dengan harapan pengguna.
- Pendekatan prototipe memungkinkan pengembang untuk menyesuaikan sistem berdasarkan umpan balik dari pemerintah. Jika ada perubahan kebutuhan, pengembang bisa melakukan penyesuaian lebih awal tanpa harus mengubah sistem yang sudah jadi sepenuhnya.
- Prototipe membantu tim pengembang menemukan kesalahan dalam bidang desain ataupun fitur sebelum mengeluarkan biaya besar untuk pengembangan akhir. Hal ini membuat perbaikan lebih cepat dan lebih hemat biaya dibandingkan jika kesalahan baru ditemukan di tahap akhir.



Analisis Kekurangan pendekatan prototipe dalam proyek ini

- Jika pemerintah terus meminta perubahan dan revisi sebelum sistem benar-benar selesai, proyek bisa memakan lebih banyak waktu dan biaya dari yang direncanakan. Perusahaan juga bisa kesulitan menentukan kapan prototipe sudah cukup dan siap dikembangkan menjadi sistem yang sebenarnya.
- Prototipe biasanya hanya gambaran awal dan belum dirancang untuk langsung digunakan sebagai sistem utama. Jika langsung dijadikan produk final tanpa perbaikan lebih lanjut, bisa muncul masalah seperti kode program yang kurang efisien atau sulit dikembangkan lebih lanjut. Bahkan, bisa jadi prototipe harus dibangun ulang dari nol, yang tentunya butuh lebih banyak waktu dan biaya.



Any Question ?

The background is a solid dark blue. In the center is a large sphere with a pixelated, dithered texture. The sphere is partially covered by three overlapping, semi-transparent bands of color: a blue band on the left, a pink band on the right, and a purple band at the top. Scattered around the central sphere are several small, stylized geometric symbols: two plus signs (+) in a light purple color, two 'x' marks in a light orange color, and four circles in various shades of blue and purple. The text 'Thank You' is centered over the sphere in a bold, white, sans-serif font.

Thank You