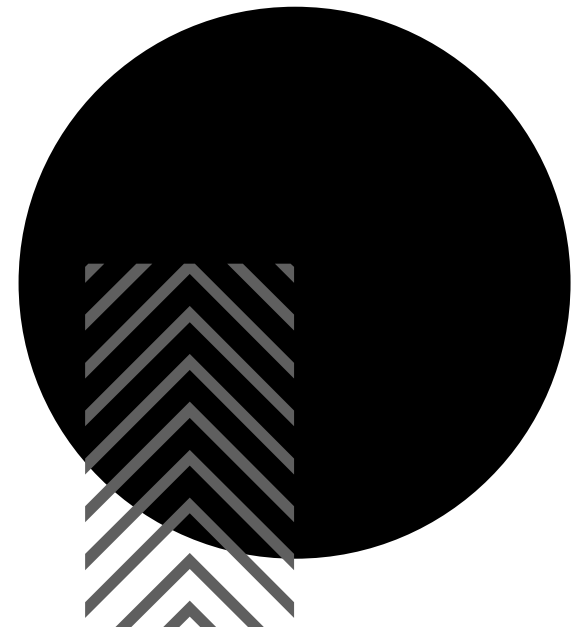




KONSEP PERANGKAT LUNAK



Tim Ajar Rekayasa Perangkat Lunak



Biodata Dosen



Annisa Puspa Kirana, S.Kom, M.Kom
082142919990

puspakirana@polinema.ac.id
<https://s.id/puspakirana>

Rules



Don't be late



Don't call (WA Only)

Monday-Friday(07.00 - 18.00 WIB)



Don't cheat

Discount 50%

DETAIL RPL



2 SKS



4 Jam/Minggu



Mampu membuat perancangan tentang konsep rekayasa perangkat lunak, software development life cycle (SDLC), UML dan dasar pengujian (testing) pada pengembangan perangkat lunak, serta dapat mengoperasikan piranti pendukung untuk pengembangan perangkat lunak



SILABUS

- ❑ Minggu 1: Konsep Perangkat Lunak
- ❑ Minggu 2: Konsep Model Perangkat Lunak
- ❑ Minggu 3: Rekayasa System
- ❑ Minggu 5: Context Diagram Dan Data Flow Diagram
- ❑ Minggu 6: ERD
- ❑ Minggu 7: Prinsip-Prinsip Dasar Desain Dan Implementasi Perangkat Lunak
- ❑ Minggu 10: Candidate Class
- ❑ Minggu 9: Squence Diagram
- ❑ Minggu 11: Class Diagram
- ❑ Minggu 13: Activity Diagram
- ❑ Minggu 14: Component Diagram
- ❑ Minggu 15: Analisis Desain Berbasis Objek
- ❑ Minggu 16: Pengujian perangkat lunak



ASSESSMENT



2X Kuis: 30%

Tugas: 30%

UTS: 20%

UAS: 20%



REFERENSI DAN SOFTWARE

Referensi

1. Roger S. Pressman, Software Engineering A Practioner's Apporach, 2014
2. John Hunt, 2000, The Unified Process for Practioners: Object-Oriented Design, UML and Java, Second Edition, Springer Science & Business Media
3. Doug Rosenberg and Matt Stephens, 2007, Use Case Driven Object Modeling with UML Theory and Practice

Software

1. Star UML
2. Rational Rose

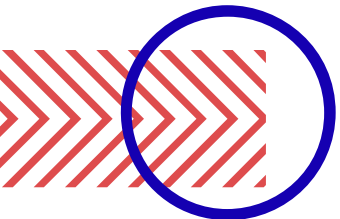
BAHAN KAJIAN

01 Konsep dasar dan karakteristik perangkat lunak

03 Mutu dan daur hidup perangkat lunak

02 Kategori perangkat lunak

04 Model proses perangkat lunak

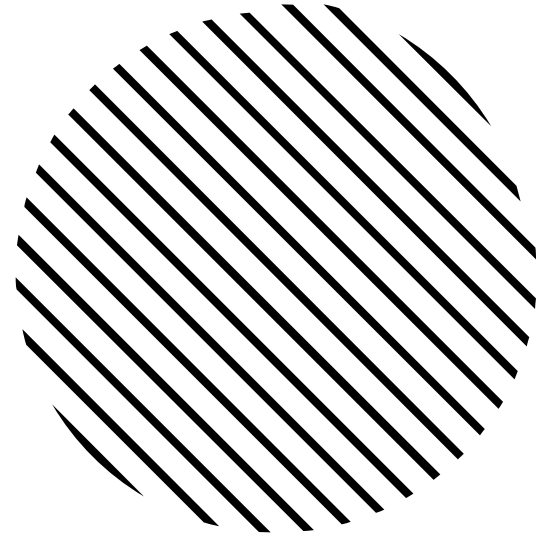




01



Karakteristik dan proses perangkat lunak






KONSEP DASAR – WHAT?

Rekayasa Perangkat Lunak adalah adalah penerapan pendekatan yang sistematis , disiplin, dan dapat diukur, terhadap pengembangan, operasi, dan pemeliharaan perangkat lunak, dan studi tentang pendekatan-pendekatan tersebut, yaitu penerapan teknik/rekayasa terhadap perangkat lunak (IEEE).

RPL adalah disiplin rekayasa yang berkaitan dengan **semua aspek produksi perangkat lunak** dari **tahap awal spesifikasi sistem sampai dengan pemeliharaan sistem** ketika sistem mulai digunakan (Sommerville).





KONSEP DASAR – WHY?

- Pengembangan perangkat lunak skala besar tidak mungkin dapat dikerjakan oleh satu orang saja
- Diperlukan kerja tim yang baik agar pengembangan dapat berjalan sesuai rencana
- Untuk itu, diperlukan teknik-teknik rekayasa untuk pengembangan perangkat lunak
- RPL belum tentu diperlukan untuk pengembangan PL skala kecil, namun sudah mulai diperlukan untuk pengembangan skala medium.





KONSEP DASAR – WHERE & WHEN?

Where ?

RPL diperlukan pada pengembangan PL skala medium dan besar.

When ?

- RPL diperlukan ketika pengembangan PL tidak dapat dikerjakan hanya dengan cara-cara tradisional.
- Ketika pengembangan PL memerlukan teknik-teknik rekayasa agar produk PL yang dihasilkan sesuai target yang direncanakan.





KONSEP DASAR – TUJUAN RPL

Menghasilkan sebuah perangkat lunak yang **berkualitas**

Menghasilkan perangkat lunak dengan **biaya** yang **efisien**





PERANGKAT LUNAK...(I)

Perangkat lunak adalah sekumpulan item atau objek yang membentuk “konfigurasi” yang meliputi :

- program/instruksi
- struktur data
- dokumentasi

Perangkat lunak = program + data + dokumentasi





KARAKTERISTIK PERANGKAT LUNAK

Maintainability

Kemampuan dari perangkat lunak tersebut dalam “kemudahan untuk modifikasi” misalnya untuk memperbaiki kinerja, memperbaiki kesalahan dan memperluas fungsi sari perangkat lunak

Dependability


Dapat mengikuti perkembangan teknologi

Robust

Dapat mengikuti keinginan pengguna

Efektif & efisien

Efektif dan efisien dalam menggunakan energi dan penggunaannya



Usability

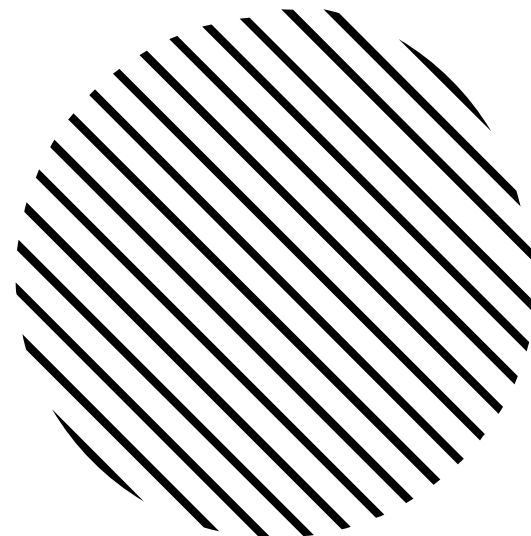
Dapat memenuhi kebutuhan yang diinginkan



02



**Kategori perangkat
lunak**






PRODUK PL

Produk PL generik (umum)

- PL yang dibuat oleh perusahaan pembuat PL untuk dijual kepada publik.
- Semua spesifikasi produk dan perubahannya ditentukan oleh perusahaan pembuatnya.
- Contoh: Microsoft Office, Adobe Acrobat, dsb.

Produk PL pesanan (khusus)

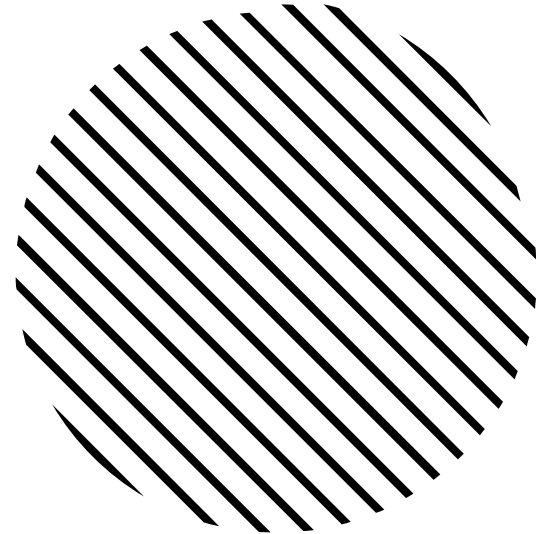
- PL yang dibuat berdasarkan pesanan pelanggan.
 - Semua spesifikasi produk dan perubahannya ditentukan berdasarkan permintaan dari pelanggannya.
 - Populer dengan istilah: Taylor-made Software, Customized Software.
 - Contoh: Sistem Informasi suatu instansi, Sistem Pengendali Lalu Lintas
- 



03



**Mutu dan daur hidup
perangkat lunak**





MUTU PERANGKAT LUNAK... (I)

Terdapat 3 pihak (minimal) yang mempengaruhi mutu PL yaitu:

1. Sponsor


Pengembangan atau perantaraan sistem software dan biasanya mempunyai respon terhadap pengembangan sistem software itu sendiri dengan melibatkan perhitungan biaya yang optimal

2. User

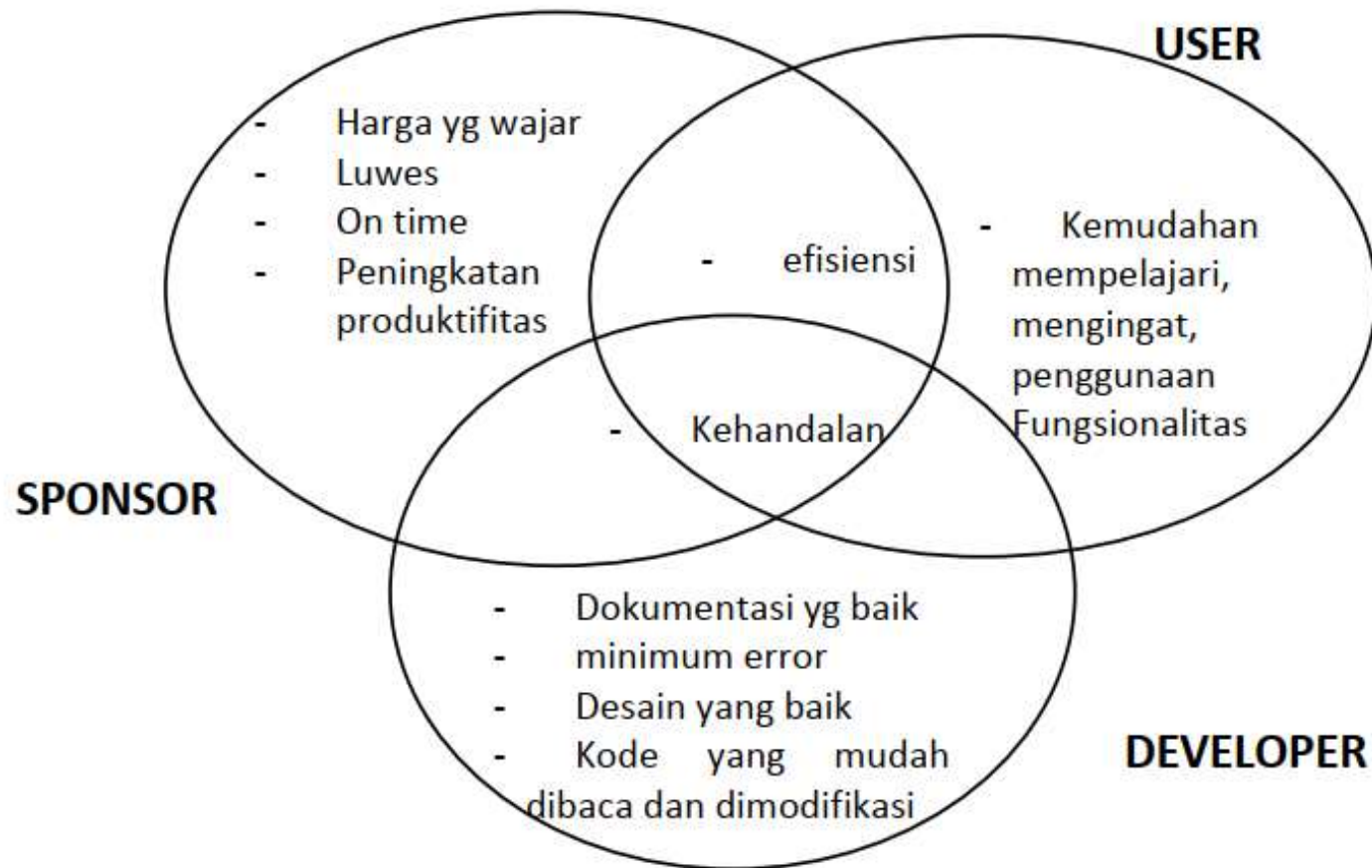
Setiap orang yang secara langsung berinteraksi terhadap eksekusi software, yang secara langsung memberi input ke komputer dan menggunakan/menikmati output dari komputer

3. Developer

Setiap orang yang secara langsung berinteraksi terhadap eksekusi software, yang secara langsung memberi input ke komputer dan menggunakan/menikmati output dari komputer.



MUTU PERANGKAT LUNAK... (2)



Gambar Sisi Pandang dari komponen kategori terhadap Mutu Perangkat Lunak



FAKTOR UTAMA KEGAGALAN PL

- Kebutuhan customer tidak bisa dipahami dan ditangkap dengan tepat
- Kebutuhan customer sering mengalami perubahan
- Customer tidak bisa bekerja sama dengan pengembang
- Pengembang kurang memiliki kecakapan dalam menjalankan tugas
- Sistem yang dikembangkan tidak terlalu banyak memberikan manfaat kepada customer



DAUR HIDUP PL... (I)

- **SDLC (Software Development Life Cycle, Siklus Hidup Pengembangan Sistem)** adalah proses pembuatan dan perubahan sistem serta model dan metodologi yang digunakan untuk mengembangkan sistem-sistem.

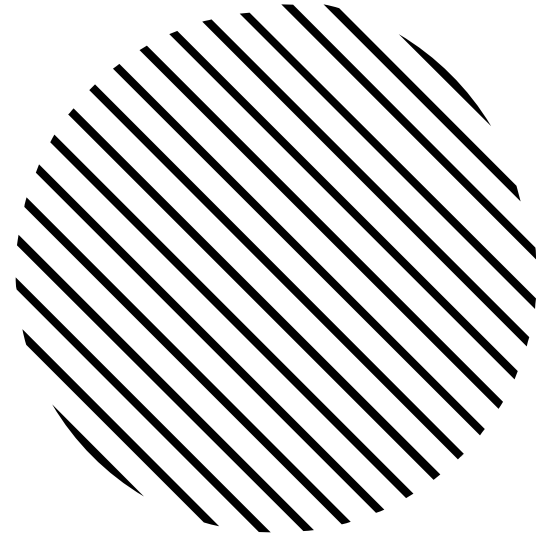




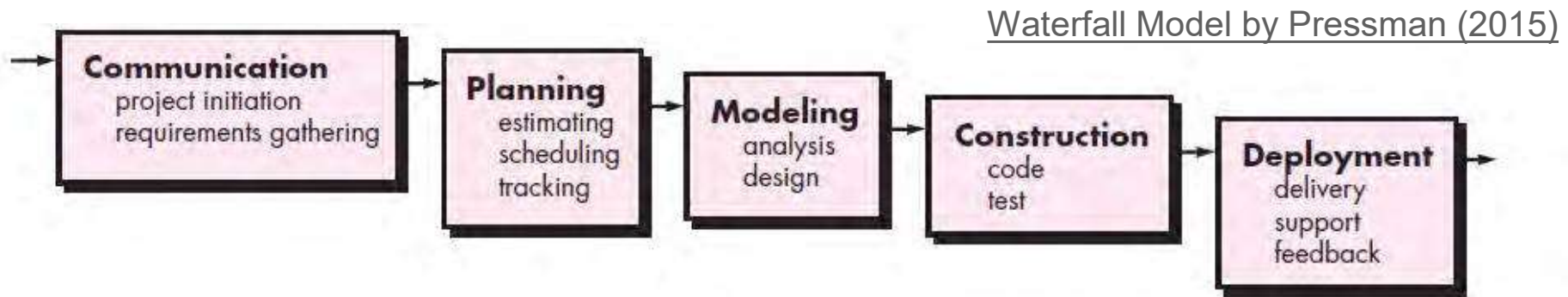
04



Model proses perangkat lunak



MODEL SEKUENSIAL LINIER ATAU WATERFALL

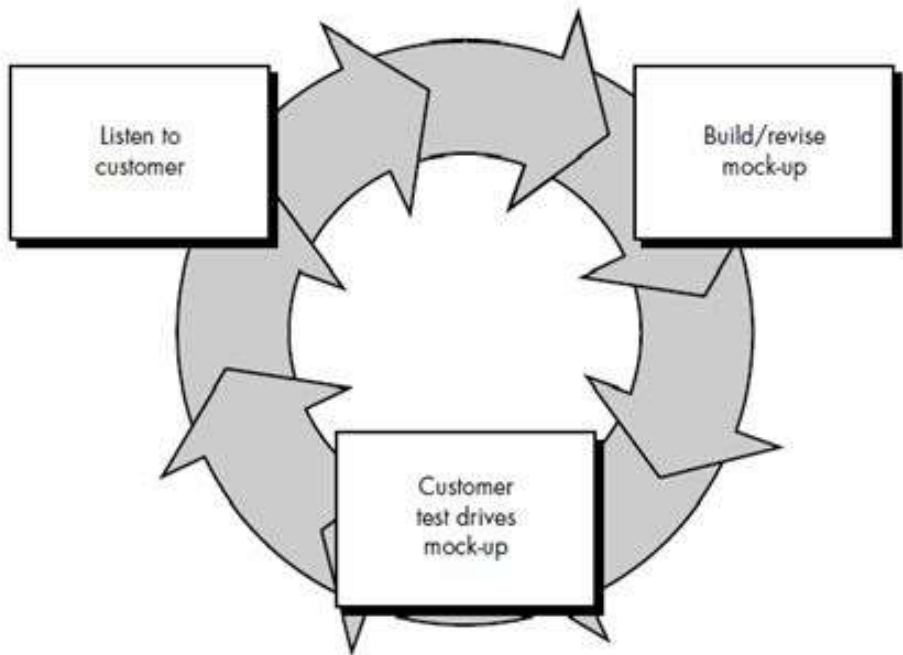


Contoh: Pembuatan program pendaftaran online ke suatu Instansi Pendidikan.

Program ini akan sangat membantu dalam proses pendaftaran, karena dapat meng-efektifkan waktu serta pendaftar tidak perlu repot-repot langsung mendatangi Instansi Pendidikan. Teknisnya adalah sebagai berikut :

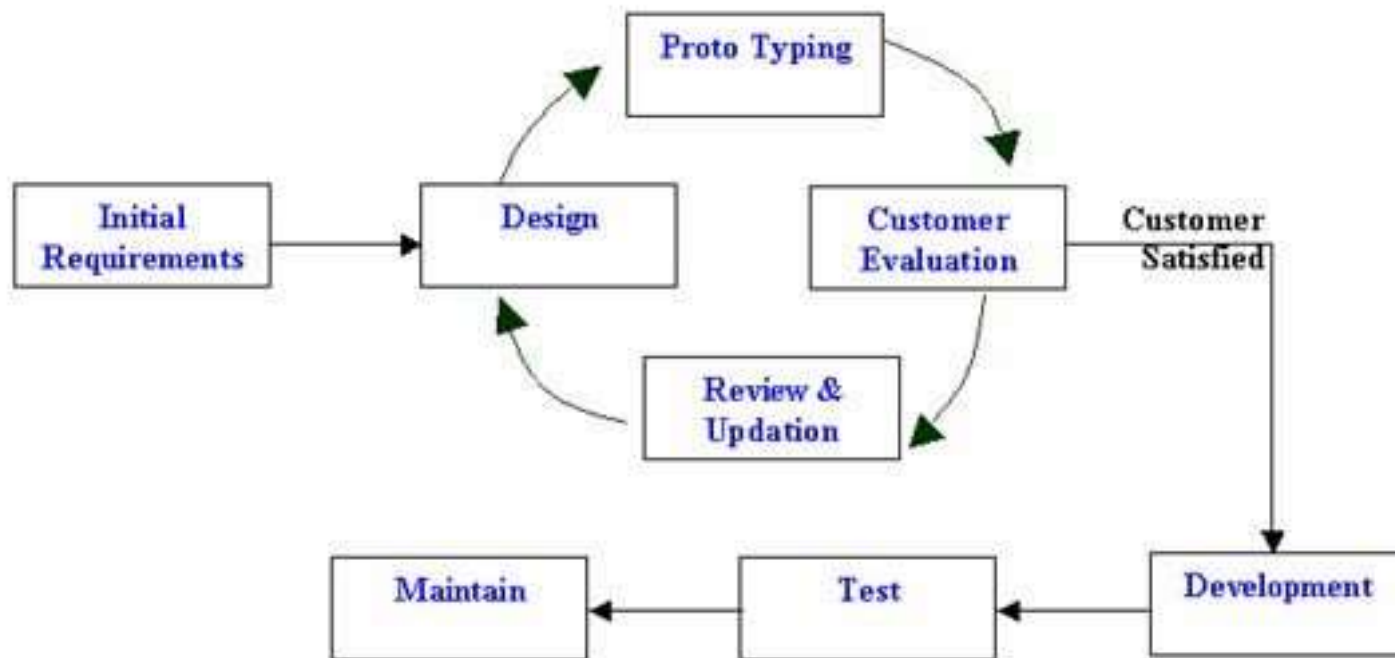
- Sistem program untuk pendaftaran dibuat menggunakan bahasa pemrograman PHP, dengan Sistem Database yang dibuat menggunakan MySQL, dan diterapkan (diaplikasikan) pada PC (personal computer) dengan sistem operasi berbasis Microsoft Windows, Linux, dan sebagainya.
- Setelah program selesai dibuat dan kemudian dipergunakan oleh user, programmer akan memelihara serta menambah atau menyesuaikan program dengan kebutuhan serta kondisi user.

MODEL PROTOTYPE (I)



- Dalam Model Prototype, prototype dari perangkat lunak yang dihasilkan kemudian dipresentasikan kepada pelanggan, dan pelanggan tersebut diberikan kesempatan untuk memberikan masukan sehingga perangkat lunak yang dihasilkan nantinya betul-betul sesuai dengan keinginan dan kebutuhan pelanggan.
- Perubahan dan presentasi prototype dapat dilakukan berkali-kali sampai dicapai kesepakatan bentuk dari perangkat lunak yang akan dikembangkan.

MODEL PROTOTYPE (2)



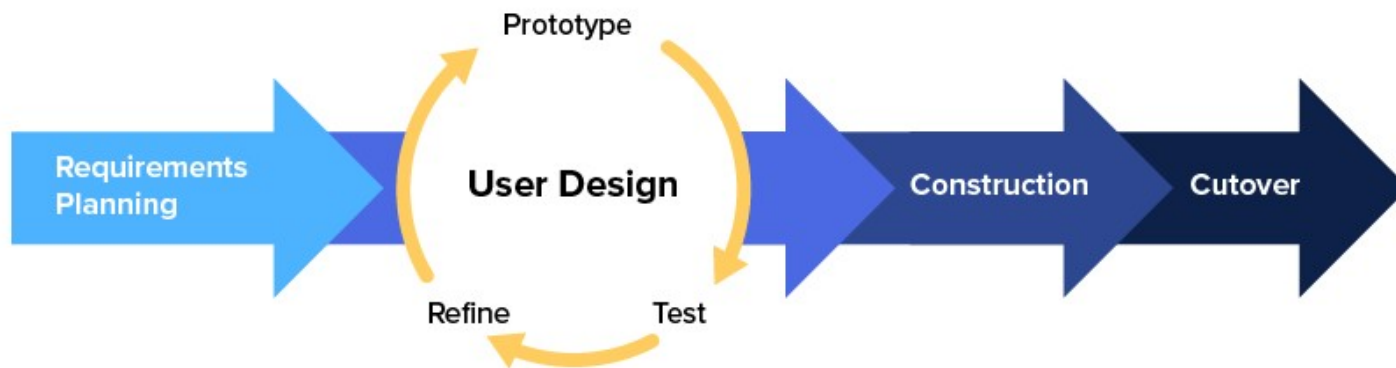
Proto Type Model

MODEL PROTOTYPE (3)

Contoh:

Sebuah rumah sakit ingin membuat aplikasi sistem database untuk pendataan pasiennya. Seorang atau sekelompok programmer akan melakukan identifikasi mengenai apa saja yang dibutuhkan oleh pelanggan, dan bagaimana model kerja program tersebut. Kemudian dilakukan rancangan program yang diujikan kepada pelanggan. Hasil/penilaian dari pelanggan dievaluasi, dan analisis kebutuhan pemakai kembali dilakukan.

RAPID APPLICATION DEVELOPMENT (RAD)



- *Rapid application development* (RAD) adalah metode yang berfokus pada pengembangan aplikasi secara cepat, melalui pengulangan dan *feedback* berulang-ulang
- Membutuhkan tim berisikan *developer* yang benar-benar memiliki *skill* tinggi
- Untuk proyek yang memerlukan waktu yang singkat

MODEL SPIRAL



- Model spiral merupakan penggabungan dari model prototyping dan model waterfall.
- Model prototyping yang fokus pada penyajian atau presentasi kepada user dengan format input dan output kemudian perangkat lunak akan dievaluasi.
- Model waterfall yang fokus kepada proses pengembangan perangkat lunak yang sistematis atau berurutan.
- Model yang sangat berguna untuk melakukan pembangunan proyek-proyek besar dan prosesnya dilakukan dengan memperhatikan resiko proyek sehingga pada akhirnya akan menghasilkan model proses yang tepat sesuai kebutuhan pengguna.



LATIHAN SOAL

1. Jelaskan menurut pendapat anda apa yang dimaksud dengan RPL!
 2. Berikan kelemahan dan kelebihan pada masing-masing model proses perangkat lunak tersebut!
 3. Jelaskan tentang scrum dan agile!
 4. Jelaskan menurut pendapat kalian apakah yang terjadi jika sebuah perangkat lunak tidak memiliki karakteristik dependability dan usability!
- 