

RPL

(Rekayasa Perangkat Lunak)

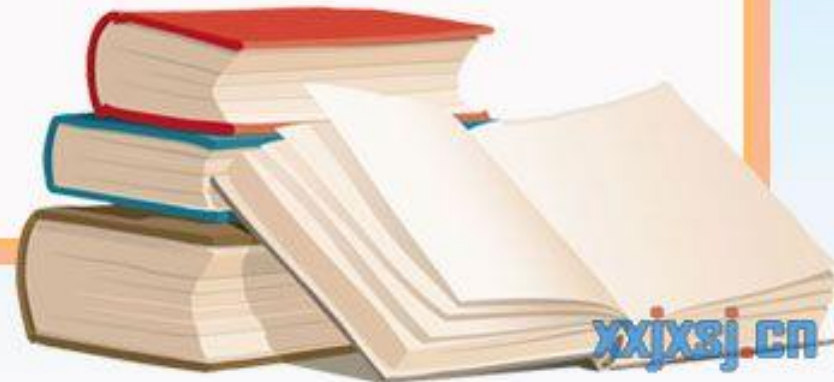
“ SOFTWARE PROCESS MODEL”

Nama : Dewi Puspitasari

Kelas : MI7

NIM : 1421024181

Dosen : Teguh Peribadi S.Pd.



a. Software process model yang paling disukai

Dari beberapa Prespective model yang telah di jabarkan, model software proses yang paling saya sukai adalah **spiral model**

Alasan : karena mengingat banyaknya kelebihan pada model ini terutama dapat disesuaikan agar perangkat lunak bisa dipakai selama hidup perangkat lunak komputer dan cocok untuk perangkat lunak skala besar, meskipun model ini memiliki beberapa kekurangan tapi semua kekurangan tersebut dapat tertutupi dengan besarnya kelebihan yang dimilikinya.

Kelebihan Model Spiral

- Setiap tahap pengerjaan dibuat prototyping sehingga kekurangan dan apa yang diharapkan oleh client dapat diperjelas dan juga dapat menjadi acuan untuk client dalam mencari kekurangan kebutuhan.
- Lebih cocok untuk pengembangan sistem dan perangkat lunak skala besar.
- Dapat disesuaikan agar perangkat lunak bisa dipakai selama hidup perangkat lunak komputer.
- Pengembang dan pemakai dapat lebih mudah memahami dan bereaksi terhadap resiko setiap tingkat evolusi karena perangkat lunak terus bekerja selama proses.
- Menggunakan prototipe sebagai mekanisme pengurangan resiko dan pada setiap keadaan di dalam evolusi produk.
- Tetap mengikuti langkah-langkah dalam siklus kehidupan klasik dan memasukkannya ke dalam kerangka kerja iteratif.
- Membutuhkan pertimbangan langsung terhadap resiko teknis sehingga mengurangi resiko sebelum menjadi permasalahan yang serius.

b. Software process model yang paling tidak disukai

Dari beberapa Prespective model yang telah di jabarkan, model software proses yang saya tidak sukai adalah **Prototyping Model**.

Alasan : karena mengingat banyaknya kekurangan pada model ini, meskipun model ini dapat diterapkan pada model apapun dan tepat digunakan jika pihak client menginginkan prototype dari software dalam waktu yang singkat.

Kekurangan **Prototyping Model**.

- Pada prototype tentu saja banyak kebutuhan yang tidak di tampilkan seluruhnya karena data yang dikumpulkan hanya sebagian.
- Prototype yang di setuju oleh client harus dikembangkan oleh developer tanpa ada data tambahan dari client dan software dari prototype harus memiliki fungsi yang lengkap.
- Banyak ketidak sesuaian pada bentuk prototype.
- Proses analisis dan perancangan terlalu singkat.

Kekurangan **Prototyping Model**.

- Pengembang kadang-kadang membuat kompromi implementasi dengan menggunakan sistem operasi yang tidak relevan dan algoritma yang tidak efisien.
- Mengesampingkan alternatif pemecahan masalah.
- Biasanya kurang fleksible dalam menghadapi perubahan.
- Prototype yang dihasilkan tidak selamanya mudah dirubah.
- Walaupun pemakai melihat berbagai perbaikan dari setiap versi prototype, tetapi pemakai mungkin tidak menyadari bahwa versi tersebut dibuat tanpa memperhatikan kualitas dan pemeliharaan jangka panjang.

Kekurangan **Prototyping Model.**

- Pelanggan kadang tidak melihat atau menyadari bahwa perangkat lunak yang ada belum mencantumkan kualitas perangkat lunak secara keseluruhan dan juga belum memikirkan kemampuan pemeliharaan untuk jangka waktu lama.
- Pengembang biasanya ingin cepat menyelesaikan proyek. Sehingga menggunakan algoritma dan bahasa pemrograman yang sederhana untuk membuat prototyping lebih cepat selesai tanpa memikirkan lebih lanjut bahwa program tersebut hanya merupakan cetak biru sistem.
- Hubungan pelanggan dengan komputer yang disediakan mungkin tidak mencerminkan teknik perancangan yang baik.

Thank you!

