Nama : Alan Aditya

NIM : 2304130087

Rombel : 2 Teknik Informatika

**Mitos-mitos dalam pengembangan perangkat lunak**

1. **"Jika ketinggalan jadwal, kita bisa menambah programmer dan mengejar ketertinggalan."** Mitos ini sangat berbahaya. Menambahkan lebih banyak orang ke proyek yang terlambat seringkali justru memperlambatnya. Tim yang baru perlu waktu untuk beradaptasi, memahami kode yang sudah ada, dan terintegrasi dengan tim yang sudah ada. Analogi yang sering digunakan adalah "sembilan wanita tidak bisa melahirkan bayi dalam satu bulan". Masalah keterlambatan jadwal seringkali lebih dalam dari sekadar kurangnya tenaga kerja, dan membutuhkan evaluasi ulang perencanaan, persyaratan, atau bahkan arsitektur perangkat lunak.
2. **"Persyaratan yang terlalu umum tentang tujuan akhir untuk mulai membangun program."** Mitos ini mengabaikan pentingnya spesifikasi yang jelas dan detail. Tujuan yang terlalu umum akan menghasilkan interpretasi yang berbeda-beda dari tim pengembang, yang berujung pada perangkat lunak yang tidak sesuai dengan kebutuhan pengguna. Sebelum mulai *coding*, persyaratan perangkat lunak harus didefinisikan dengan baik, termasuk fungsionalitas, kinerja, antarmuka pengguna, dan batasan-batasan lainnya.
3. **"Perubahan persyaratan proyek mudah diakomodasi karena perangkat lunak fleksibel."** Meskipun perangkat lunak lebih fleksibel daripada konstruksi bangunan, misalnya, perubahan persyaratan tetap memiliki biaya. Perubahan di tengah proyek bisa sangat mahal dan memakan waktu karena perlu analisis dampak, desain ulang, pengujian ulang, dan dokumentasi ulang. Mengelola persyaratan perubahan dengan baik adalah bagian penting dari rekayasa perangkat lunak.
4. **"Setelah kita menulis program yang berfungsi, kita selesai."** Mitos ini mengabaikan siklus hidup perangkat lunak yang panjang. Program yang berfungsi hanyalah fase awal. Perangkat lunak perlu diuji, diintegrasikan, di-deploy, dipelihara, dan mungkin ditingkatkan di masa depan. Selain itu, dokumentasi yang baik juga penting agar program dapat dipahami dan dimodifikasi oleh orang lain di kemudian hari.
5. **"Sampai saya menjalankan program, saya tidak punya cara untuk menilai kualitasnya."** Kualitas perangkat lunak tidak hanya diukur dari apakah program itu berjalan atau tidak. Ada banyak aspek kualitas lain yang perlu dipertimbangkan, seperti kinerja, keamanan, keandalan, *usability*, dan *maintainability*. Penilaian kualitas harus dilakukan sepanjang siklus pengembangan, bukan hanya setelah program selesai. Teknik seperti *code review*, pengujian unit, dan pengujian integrasi dapat membantu mengidentifikasi masalah kualitas sejak dini.
6. **"Satu-satunya produk kerja yang dapat dikirim untuk proyek yang sukses adalah program yang berfungsi."** Mitos ini meremehkan pentingnya dokumentasi, desain, rencana pengujian, dan artefak lainnya. Dokumentasi yang baik sangat penting untuk pemahaman, pemeliharaan, dan pengembangan lebih lanjut. Desain yang baik membantu memastikan kualitas dan skalabilitas perangkat lunak. Rencana pengujian membantu memastikan bahwa perangkat lunak memenuhi persyaratan. Semua artefak ini merupakan bagian penting dari proyek perangkat lunak yang sukses.
7. **"Rekayasa perangkat lunak akan membuat kita membuat terlalu banyak dokumentasi dan akan memperlambat kita."** Mitos ini salah kaprah. Meskipun dokumentasi yang berlebihan memang bisa memperlambat, dokumentasi yang tepat justru mempermudah pemahaman, pemeliharaan, dan pengembangan perangkat lunak dalam jangka panjang. Rekayasa perangkat lunak yang baik membantu tim bekerja lebih efisien dengan proses yang terstruktur dan terukur. Dokumentasi yang baik juga membantu mengurangi risiko dan meningkatkan kualitas perangkat lunak.