1. Version Control System (Sistem Kontrol Versi)

Apa itu?

Version Control System (VCS) atau Sistem Kontrol Versi adalah sebuah sistem yang merekam setiap perubahan yang terjadi pada sebuah file atau sekumpulan file dari waktu ke waktu. Dengan VCS, kita bisa kembali ke versi sebelumnya, membandingkan perubahan, melihat siapa yang terakhir mengubah sesuatu, dan banyak lagi. Ini sangat penting dalam pengembangan perangkat lunak agar setiap perubahan kode dapat dilacak dan dikelola dengan baik.

Mengapa Penting?

Kolaborasi Tim: Memungkinkan banyak developer bekerja pada proyek yang sama secara bersamaan tanpa menimbulkan konflik. Setiap developer bisa bekerja di "cabang" (branch) mereka sendiri dan kemudian menggabungkan pekerjaan mereka ke proyek utama.

Mencegah Kehilangan Data: Jika terjadi kesalahan atau bug, kita bisa dengan mudah kembali ke versi kode sebelumnya yang masih berfungsi dengan baik.

Pelacakan Perubahan: Setiap perubahan tercatat dengan jelas, termasuk siapa yang mengubah, kapan, dan apa yang diubah. Ini membuat proses pengembangan menjadi lebih transparan.

Contoh Populer

Git: Sistem kontrol versi terdistribusi yang paling populer saat ini.

Subversion (SVN): Sistem kontrol versi terpusat.

Mercurial: Sistem kontrol versi terdistribusi lainnya.

2. Scrum vs Kanban

Pengantar

Scrum dan Kanban adalah dua kerangka kerja (framework) populer dalam metodologi Agile yang membantu tim mengelola pekerjaan mereka secara efisien. Keduanya bertujuan untuk meningkatkan alur kerja dan menghasilkan produk yang lebih baik, namun dengan pendekatan yang berbeda.

| Fitur | Scrum | Kanban |
| --- | --- | --- |
| **Irama Kerja** | Berbasis **Sprint**, yaitu siklus kerja dengan durasi tetap (biasanya 2-4 minggu). | Berbasis **Alur Berkelanjutan** (*continuous flow*). Tugas dikerjakan secara terus-menerus tanpa ada batasan waktu yang kaku. |
| **Peran** | Memiliki peran yang terdefinisi dengan jelas: **Product Owner**, **Scrum Master**, dan **Development Team**. | Tidak memiliki peran yang wajib. Tim bisa mempertahankan struktur yang sudah ada. |
| **Rapat** | Memiliki serangkaian rapat yang wajib: *Sprint Planning*, *Daily Scrum*, *Sprint Review*, dan *Sprint Retrospective*. | Rapat bersifat opsional dan disesuaikan dengan kebutuhan tim. |
| **Pengukuran** | Fokus pada kecepatan tim dalam menyelesaikan pekerjaan dalam satu Sprint (*velocity*). | Fokus pada waktu yang dibutuhkan untuk menyelesaikan satu tugas (*cycle time*). |
| **Perubahan** | Perubahan tidak diizinkan selama Sprint sedang berjalan untuk menjaga fokus tim. | Perubahan bisa dilakukan kapan saja selama kapasitas tim memungkinkan. |
| **Visualisasi** | Menggunakan **Scrum Board** yang biasanya di-reset setiap akhir Sprint. | Menggunakan **Kanban Board** yang terus-menerus diperbarui. |

Kapan Menggunakannya?

Gunakan Scrum jika:

Proyek memiliki prioritas yang jelas dan bisa dipecah menjadi bagian-bagian kecil (Sprint).

Tim membutuhkan struktur dan ritme kerja yang teratur.

Ada kebutuhan untuk perencanaan dan estimasi yang terstruktur.

Gunakan Kanban jika:

Prioritas pekerjaan sering berubah.

Tim perlu merespons permintaan dengan cepat.

Fokus utama adalah visualisasi alur kerja dan efisiensi.