

## Kemampuan-Dasar -2

### Ruang lingkup software engineer

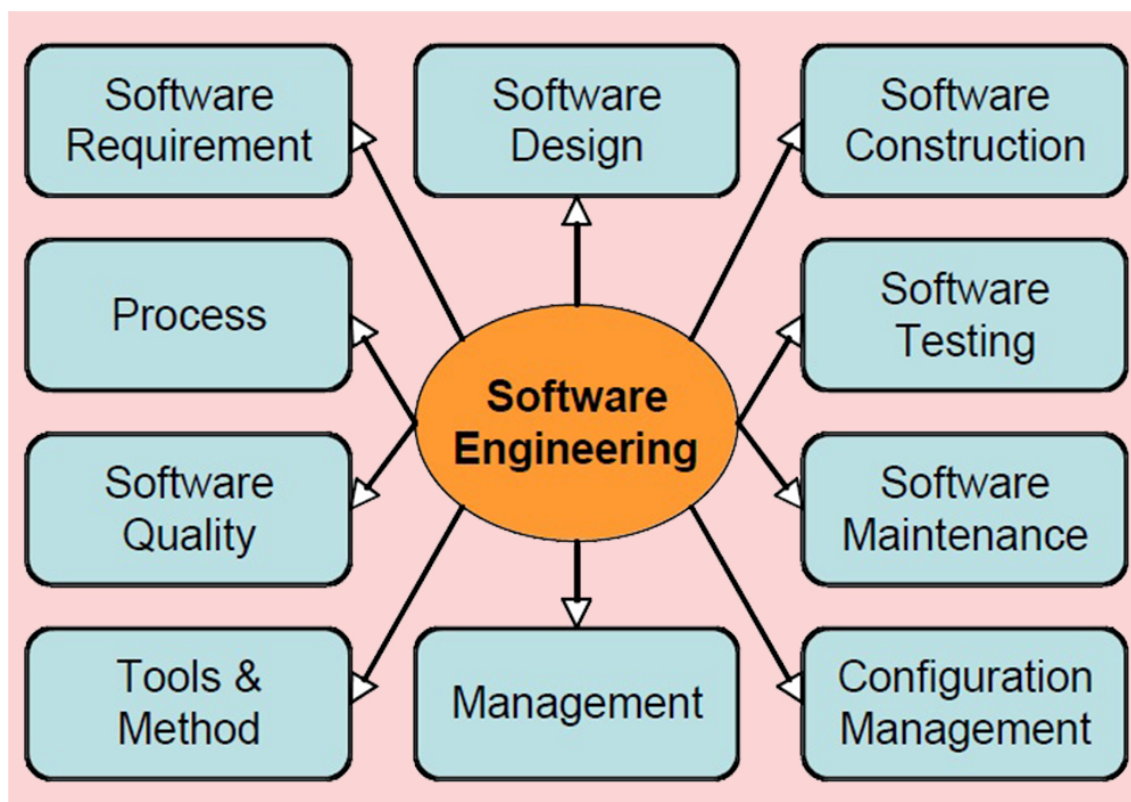
Ruang Lingkup Rekayasa Perangkat Lunak Rekayasa Perangkat sendiri mempunyai ruang lingkup kerjanya Sebagai berikut: Pengertian, Tujuan dan Ruang Lingkup Rekayasa Perangkat Lunak Ruang Lingkup Rekayasa Perangkat Lunak

Software Requirements berhubungan dengan spesifikasi kebutuhan dan persyaratan perangkat lunak.

Software Design meliputi proses penampilan arsitektur, komponen, antar muka, dan karakteristik lain dari perangkat lunak

Software Construction berhubungan dengan detail pengembangan perangkat lunak, termasuk algoritma, pengkodean, pencarian kesalahan dan pengujian.

Software Testing meliputi pengujian pada kinerja perangkat lunak secara keseluruhan



### Keterkaitan software engineering dengan SCM - Software Configuration Management.

Software Configuration Management (SCM) Software Configuration Management (SCM) atau sering dikenal dengan manajemen konfigurasi software merupakan seni untuk mengidentifikasi, mengorganisasi, kontrol modifikasi pada software yang sedang di bangun oleh tim programming.

Tujuan SCM Tujuan dari SCM adalah untuk mengidentifikasi, mengontrol, mengimplementasikan, serta melaporkan perubahan kepada pihak-pihak yang mempunyai kepentingan.

Software Maintenance vs Software Configuration Management Software Maintenance adalah serangkaian aktivitas rekayasa perangkat lunak yang terjadi setelah perangkat lunak diserahkan ke pelanggan setelah dioperasikan.

Sedangkan software configuration management merupakan serangkaian kegiatan tracking & control yang dimulai ketika suatu proyek perangkat lunak dimulai dan berakhir ketika perangkat lunak sudah tidak beroperasi lagi.

#### **Keterkaitan SCM dengan Version Control dan Distributed Version Control.**

Memulai - Tentang Version Control Bab ini akan membahas tentang memulai dengan Git. Kita akan mulai dengan menjelaskan beberapa latar belakang pada peralatan version control, kemudian beralih ke bagaimana cara agar Git dapat berjalan pada sistem Anda, dan terakhir, bagaimana cara mengaturnya agar dapat mulai bekerja dengan Git. Pada akhir bab ini Anda seharusnya telah paham mengapa Git ada, mengapa sebaiknya Anda menggunakannya dan sebaiknya Anda sudah siap untuk melakukannya.

Tentang Version Control Apa itu version control dan mengapa sebaiknya Anda peduli? Version control adalah sebuah sistem yang merekam perubahan-perubahan dari sebuah berkas atau sekumpulan berkas dari waktu ke waktu sehingga Anda dapat menilik kembali versi khusus suatu saat nanti. Sebagai contoh, pada buku ini Anda akan menggunakan sumber kode perangkat lunak sebagai berkas-berkas yang direkam dengan version control, walau pada kenyataannya Anda dapat melakukan ini dengan hampir semua jenis berkas pada komputer.

Jika Anda adalah seorang perancang grafis atau web dan ingin menyimpan setiap versi dari sebuah gambar atau layout (yang tentunya Anda ingin melakukannya), sebuah Version Control System (VCS) adalah hal yang bijak untuk digunakan. VCS memperbolehkan Anda untuk mengembalikan berkas-berkas ke keadaan sebelumnya, mengembalikan seluruh proyek kembali ke keadaan sebelumnya, membandingkan perubahan-perubahan di setiap waktu, melihat siapa yang terakhir mengubah sesuatu yang mungkin menimbulkan masalah, siapa dan kapan yang mengenalkan sebuah isu dan banyak lagi. Menggunakan VCS secara umum juga berarti bahwa jika Anda melakukan kesalahan atau menghilangkan berkas, Anda dapat dengan mudah memulihkannya. Sebagai tambahan, Anda mendapatkan semua ini dengan biaya yang sangat sedikit.

Sistem Version Control Lokal Metode version control yang banyak dipilih oleh orang-orang adalah dengan menyalin berkas-berkas ke direktori lain (mungkin direktori yang diberi catatan waktu, jika mereka cerdas). Pendekatan ini sangat umum karena ini sangat sederhana, namun ini juga sangat rentan terkena galat. Mudah sekali untuk lupa pada direktori mana Anda sedang berada dan menulis ke berkas yang salah atau menyalin setiap berkas yang bukan Anda maksud secara tidak sengaja.

Untuk menghadapi hal ini, dahulu para programmer mengembangkan VCS lokal yang memiliki database sederhana yang menyimpan semua perubahan pada berkas pada revision control. Figure 1. Diagram version control lokal. Salah satu alat VCS yang lebih terkenal adalah sistem yang disebut dengan RCS, yang masih disebarkan dengan banyak komputer saat ini. Bahkan sistem operasi Mac OS X yang terkenal menyertakan perintah rcs ketika Anda memasang Developer Tools. RCS bekerja dengan cara menyimpan sekumpulan patch (itulah, perbedaan antara berkas-berkas) dalam sebuah format dalam disk; itu kemudian dapat membuat ulang sebarang berkas yang terlihat sama pada satu waktu dengan menambahkan semua patch.

Sistem Version Control Terpusat Masalah besar selanjutnya yang dihadapi orang-orang adalah bahwa mereka butuh bekerja bersama dengan para pengembang pada sistem lain.

Untuk menangani masalah ini, Centralized Version Control System (CVCS) dikembangkan. Sistem-sistem ini, seperti CVS, Subversion, dan Perforce, memiliki sebuah server tunggal yang berisi semua berkas-berkas yang telah diberi versi, dan beberapa klien yang melakukan check out pada berkas-berkas dari pusat tersebut. Selama bertahun-tahun, hal ini telah menjadi standar untuk version control. Figure 2. Diagram version control terpusat. Pengaturan ini menawarkan banyak keuntungan, terutama dibandingkan dengan VCS lokal. Contohnya, setiap orang tahu hingga pada tahapan apa yang orang lain sedang kerjakan di dalam proyek. Para administrator memiliki kendali yang baik mengenai siapa dapat melakukan apa; dan itu jauh lebih mudah untuk mengelola sebuah CVCS daripada menangani database lokal pada setiap klien.

Akan tetapi, pengaturan ini juga memiliki beberapa kekurangan. Yang paling jelas adalah satu titik kegagalan yang diwakili oleh server terpusat. Jika server tersebut sedang down selama satu jam, maka selama itu tidak ada orang yang dapat bekerja bersama atau menyimpan perubahan yang telah diberi versi terhadap apapun yang sedang mereka kerjakan. Jika hard disk dari database pusat menjadi corrupted, dan cadangan yang memadai belum tersimpan, Anda akan kehilangan segalanya – seluruh riwayat dari proyek kecuali setiap snapshot yang dimiliki oleh orang-orang pada mesin lokal mereka. Sistem VCS lokal menderita dari hal yang sama pula – ketika Anda memiliki semua riwayat dari proyek pada satu tempat, Anda memiliki resiko untuk kehilangan semuanya.