**Technical Knowledge & Experience**

1. **Jelaskan bagaimana anda menggunakan GIT untuk proses development sampai deployment production?**

Pertama membuat sebuah repository baru pada GIT, kemudian saya clone memakai SourceTree (yang pernah saya pakai) kemudian saya buka folder tersebut di editor code (saya pakai Visual Studio Code). Kemudian memulai instalasi untuk dependencies yang dibutuhkan dalam project tersebut

Lalu memulai untuk coding Bersama tim, jika pada tim ada update untuk project saya mulai untuk commit pada source tree kemudian push dan pull. Jika ada error terjadi Karena bentrok code salah satu harus mengalah untuk codenya dibackup dan digabung kembali jika code tersebut masih digunakan

1. **Apa yang anda ketahui dengan code refactoring? Jika tahu, bagaimana anda melakukannya?**

Refactoring adalah proses merekonstruksi/memperbaiki struktur internal sebuah sistem perangkat lunak tanpa mengubah sedikitpun fungsionalitas dari sistem. Artinya, dalam proses refactoring dilakukan modifikasi program untuk memperbaiki struktur, mengurangi kompleksitas, atau untuk membuatnya lebih mudah dimengerti. Code refactoring juga dapat diartikan sebagai code refactoring adalah proses restrukturisasi kode komputer yang ada-mengubah anjak piutang-tanpa mengubah tingkah laku eksternalnya. Refactoring meningkatkan atribut perangkat lunak nonfungsional. Keuntungan meliputi pembacaan kode yang lebih baik dan kompleksitas yang berkurang; Ini dapat memperbaiki kode sumber rawatan dan menciptakan arsitektur internal yang lebih ekspresif atau model objek untuk meningkatkan kemampuan diperpanjang. Biasanya, refactoring menerapkan serangkaian standar dasar mikro-refactorings, yang masing-masing (biasanya) merupakan perubahan kecil dalam kode sumber program komputer yang dapat mempertahankan perilaku perangkat lunak, atau setidaknya tidak mengubah kesesuaiannya dengan persyaratan fungsional. Banyak lingkungan pengembangan memberikan dukungan otomatis untuk melakukan aspek mekanis dari refactorings dasar ini. Jika dilakukan dengan sangat baik, refactoring kode juga dapat mengatasi bug komputer atau kerentanan tersembunyi, tidak aktif, atau belum ditemukan di sistem dengan menyederhanakan logika yang mendasarinya dan menghilangkan tingkat kerumitan yang tidak perlu. Jika dilakukan dengan buruk mungkin gagal memenuhi persyaratan bahwa fungsi eksternal tidak diubah, perkenalkan bug baru, atau keduanya. Dengan terus menyempurnakan desain kode, perusahaan dapat mempermudah kerja sama. Hal ini sangat berbeda dengan apa yang biasanya terjadi: refactoring kecil dan banyak perhatian diberikan untuk menambahkan fitur baru dengan mudah. Jika Anda memasuki kebiasaan higienis refactoring secara terus menerus, Anda akan merasa lebih mudah untuk memperpanjang dan mempertahankan kode.Ini menurut Joshua Kerievsky, Refactoring to Patterns

1. **Apa yang anda ketahui dari technical debt? Jika tahu bagaimana anda mengatasinya?**

Technical debt mencakup hal-hal internal yang di pilih untuk tidak dilakukan sekarang, namun akan menghambat perkembangan masa depan jika dibiarkan dibatalkan. Ini termasuk refactoring yang ditangguhkan.

Hutang Teknis tidak termasuk fungsi yang ditangguhkan, kecuali mungkin dalam kasus tepi dimana fungsionalitas yang disampaikan "cukup baik" untuk pelanggan, namun tidak memenuhi beberapa standar (misalnya, elemen UI yang tidak sesuai dengan beberapa standar UI).

Saya belum mengerti tentang technical debt, akan tetapi suatu project akan diketahui fitur mana yang harus diprioritaskan untuk dikerjakan dengan menggunakan scrum. Dengan begitu tidak ada refactoring pada pertengahan untuk mengubah struktur project tersebut

1. **Bagaimana anda mengukur kualitas suatu aplikasi?**

Saya mengukur kualitas aplikasi dari seberapa mampu aplikasi mampu untuk mencakup berbagai hal yang dibutuhkan untuk pelanggan dan bagaimana aplikasi tersebut bias memudahkan bagi para programmer. Missal sudah ada aplikasi tentang CRUD dan ada client ingin aplikasi yg serupa tentang CRUD. Bagaimana code dari aplikasi tersebut bisa mempermudah bagi yang akan meneruskan atau memodifikasi aplikasi tersebut ke bentuk yang lebih baru