



Kursus Persiapan Sertifikasi Kompetensi Skema Pemrograman Objek (Object Programmer)






Pertemuan 2


UNIT KOMPETENSI

1. Melakukan identifikasi *library*, komponen atau *framework* yang diperlukan - J.620100.003.01
2. Mengimplementasikan *user interface* - J.620100.005.02
3. Menuliskan kode dengan prinsip sesuai *guidelines* dan *best practices* - J.620100.016.01



Melakukan identifikasi *library*, komponen atau *framework* yang diperlukan

1. Menganalisis *library*, komponen, atau *framework* yang sesuai dengan konteks
 - 1.1. Ruang lingkup kebutuhan akan *library*, komponen atau *framework* diidentifikasi sesuai lingkungan pengembangan.
 - 1.2. Keuntungan penggunaan dibandingkan dengan mengembangkan sendiri diidentifikasi.
2. Membuat ***proof of concept*** *library*, komponen atau *framework* berdasarkan konteks kebutuhan
 - 2.1. Fitur-fitur terkait penggunaan *library*, komponen atau *framework* versi sederhana dibuat.
 - 2.2. Manfaat penggunaan didemostrasikan.




Melakukan identifikasi library, komponen atau framework yang diperlukan (lanjutan)

3. Merancang integrasi dan batasan penggunaan *library*, komponen atau *framework*
 - 3.1. Rencana integrasi ditentukan.
 - 3.2. Limitasi diidentifikasi.



Mengimplementasikan *user interface*

1. Mengidentifikasi rancangan *user interface*
 - 1.1. Rancangan *user interface* diidentifikasi sesuai kebutuhan.
 - 1.2. Komponen *user interface* dialog diidentifikasi sesuai konteks rancangan proses.
 - 1.3. Urutan dari akses komponen user interface dialog dijelaskan.
 - 1.4. Simulasi (*mock-up*) dari aplikasi yang akan dikembangkan dibuat.
2. Melakukan implementasi rancangan *user interface*
 - 2.1. Menu program sesuai dengan rancangan program diterapkan.
 - 2.2. Penempatan *user interface* dialog diatur secara sekuensial.
 - 2.3. Setting aktif-pasif komponen user interface dialog disesuaikan dengan urutan alur proses.
 - 2.4. Bentuk *style* dari komponen user interface ditentukan.
 - 2.5. Penerapan simulasi dijadikan suatu proses yang sesungguhnya.



Menuliskan kode dengan prinsip sesuai *guidelines* dan *best practices*

1. Menerapkan *coding-guidelines* dan *best practices* dalam penulisan program (kode sumber)
 - 1.1. Kode sumber dituliskan mengikuti *coding-guidelines* dan *best practices*.
 - 1.2. Struktur program yang sesuai dengan konsep paradigmanya dibuat.
 - 1.3. Galat/*error* ditangani.
2. Menggunakan ukuran performansi dalam menuliskan kode sumber
 - 2.1. Efisiensi penggunaan *resources* oleh kode dihitung.
 - 2.2. Kemudahan interaksi selalu diimplementasikan sesuai standar yang berlaku.



Output Pertemuan 2

1. Dokumen *Proof of concept*
2. Disain *mock-up* simulasi ATM
3. Kode program yang sesuai dengan *guidelines* dan *best practices*



Tugas Pertemuan 2

1. Lengkapi dokumen *Proof of Concept* (PoC)
2. Disain *mockup* simulasi ATM untuk menu Tarik Tunai, Transfer, dan Cek Saldo. Gunakan *prototyping tools* untuk membuat *mockup* (misal. Framer, Figma, dll)
3. Buat program