

Tugas Personal ke-1

Week 2

Nama: Frans Sebastian

NIM: 2502121162

Soal.

Pada masa ini, hampir semua aspek kehidupan manusia telah menggunakan komputer, misalnya penggunaan perangkat lunak pada bidang:

- Politik: ada aplikasi untuk perhitungan hasil pilpres, kampanye kandidat capres.
- Ekonomi: terdapat aplikasi untuk jual beli barang, perhitungan saham.
- Hiburan: aplikasi pemutar video, music.
- Perbankan: adanya aplikasi *e-banking*, *mobile banking*.

Dari contoh diatas pilihlah salah satu perangkat lunak, kemudian buatlah berdasarkan teori *software engineering* mengenai asumsi dan analisa dari perangkat lunak yang dipilih mengenai:

- a. (LO 1.; 25%) *Software cost, attribute software, key challenging* untuk perangkat lunak yang dipilih!
- b. (LO 1.; 25%) Buatlah *software structure process*, tentukan *flow process* dan berikan alasannya, dan tentukan *process model* dan berikan alasannya!
- c. (LO 2: 25%) Buatlah *requirement engineering*! (Gunakan product backlog untuk aplikasi menggunakan konsep *user stories* atau *Unified modeling language* untuk aplikasi berbasis *proses requirement*)!
- d. (LO 2: 25%) Buatlah perancangan *user interface* dari aplikasi yang dipilih!
- e. (LO3: 25%) Apa saja *metrics* yang dipakai di dalam mengukur kualitas perangkat lunak yang dipilih! Buatlah perancangan evaluasi pada aplikasi yang dipilih menggunakan *metrics* tersebut!

Jawaban

Dari study kasus di atas saya memilih perangkat lunak aplikasi e-banking dalam bidang perbankan:

a. Software Cost, Attribute, dan Key Challenges:

1. Software Cost (Biaya Perangkat Lunak):

- **Biaya Pengembangan:** Ini mencakup biaya untuk pengembangan perangkat lunak itu sendiri, termasuk upah tim pengembang, biaya perangkat keras dan perangkat lunak, serta biaya lisensi perangkat lunak.
- **Biaya Keamanan:** Melindungi aplikasi e-banking dari ancaman siber memerlukan investasi dalam keamanan perangkat lunak, seperti firewall, antivirus, dan sistem deteksi intrusi.
- **Biaya Pemeliharaan:** Biaya untuk pemeliharaan rutin, pembaruan, dan perbaikan setelah peluncuran aplikasi.
- **Biaya Dukungan Pelanggan:** Menyediakan dukungan pelanggan yang responsif dan berkualitas memerlukan sumber daya tambahan.

Software Attributes (Atribut Perangkat Lunak):

- **Keamanan:** Keamanan data nasabah adalah atribut paling kritis. Aplikasi harus melindungi data sensitif pengguna dengan enkripsi dan proteksi yang kuat.
- **Kepuasan Pelanggan:** Antarmuka pengguna yang mudah digunakan, waktu respons yang cepat, dan layanan pelanggan yang baik adalah atribut yang meningkatkan kepuasan pelanggan.
- **Kinerja:** Aplikasi harus memiliki kinerja yang baik, termasuk waktu respons yang cepat dan kapasitas beban yang memadai untuk menangani lonjakan pengguna.
- **Kepatuhan Regulasi:** Aplikasi harus mematuhi regulasi perbankan dan privasi data yang berlaku.
- **Skalabilitas:** Kemampuan aplikasi untuk tumbuh dan mengakomodasi peningkatan pengguna atau fitur baru.
- **Integrasi Sistem:** Aplikasi harus dapat mengintegrasikan dengan sistem inti bank yang ada.

Key Challenges (Tantangan Utama):

Keamanan: Melindungi data sensitif nasabah dan aplikasi dari ancaman siber seperti peretasan, phishing, dan serangan malware adalah tantangan utama. Perangkat lunak e-banking harus selalu terbaru dengan teknologi keamanan terkini.

- **Kepatuhan Regulasi:** Mengikuti peraturan perbankan yang berubah-ubah dan privasi data dapat menjadi rumit dan mahal.
- **Skalabilitas:** Aplikasi harus siap untuk menghadapi lonjakan pengguna saat waktu tertentu, seperti akhir bulan ketika banyak orang melakukan transaksi.
- **Integrasi Sistem Eksisting:** Mengintegrasikan aplikasi dengan sistem inti bank yang sudah ada bisa rumit dan memakan waktu.
- **Teknologi dan Perubahan Tren:** Perkembangan teknologi dan perubahan tren dalam perbankan digital memerlukan perangkat lunak yang dapat beradaptasi secara cepat.

b. Software Structure Process:

1. Model Pengembangan Perangkat Lunak:

- Pengembangan perangkat lunak aplikasi e-banking dapat mengikuti model iteratif dan inkremental seperti Scrum. Hal ini memungkinkan fleksibilitas dalam menangani perubahan kebutuhan pelanggan dan regulasi perbankan yang mungkin berubah seiring waktu.

2. Tim Pengembang:

- Tim pengembang harus terdiri dari berbagai peran, termasuk pengembang perangkat lunak, pengujian kualitas, analis bisnis, dan desainer UI/UX. Mereka harus berkolaborasi secara efisien selama seluruh siklus pengembangan.

3. Alur Proses:

- Berikut adalah alur umum dari proses pengembangan perangkat lunak untuk aplikasi e-banking:

a. Perencanaan (Planning):

- Membuat rencana proyek yang mencakup estimasi biaya, jadwal, dan sumber daya yang diperlukan.

• Menentukan kebutuhan dan prioritas pelanggan.

b. Analisis Kebutuhan (Requirements Analysis):

- Mengidentifikasi dan merinci kebutuhan pengguna dan persyaratan teknis.
- Menggunakan metode seperti user stories dan UML untuk mendefinisikan persyaratan.

c. Desain (Design):

- Merancang desain UI yang user-friendly dan sesuai dengan standar keamanan.
- Mengembangkan arsitektur perangkat lunak yang memenuhi persyaratan kinerja dan skalabilitas.
- Menentukan teknologi dan alat yang akan digunakan.

d. Implementasi (Implementation):

- Tim pengembang mulai mengkode aplikasi berdasarkan desain dan persyaratan yang telah dibuat.
- Menerapkan keamanan perangkat lunak seperti enkripsi data dan mekanisme otentikasi.

e. Pengujian (Testing):

- Menggunakan teknik pengujian seperti uji fungsional, uji keamanan, dan uji kinerja untuk memverifikasi bahwa aplikasi berjalan dengan baik dan aman.
- Melakukan uji integrasi untuk memastikan integrasi yang benar dengan sistem inti bank.

f. Pengiriman (Deployment):

- Menjalankan langkah-langkah peluncuran yang terkontrol untuk mengaktifkan aplikasi e-banking bagi nasabah.
- Mengawasi proses peluncuran untuk mengatasi masalah yang mungkin muncul.

g. Pemeliharaan (Maintenance):

- Memantau dan memelihara aplikasi secara teratur untuk memastikan ketersediaan dan kualitas.
- Merespons perubahan regulasi dan kebutuhan pelanggan yang muncul.

4. Kualitas dan Keamanan:

- Selama seluruh siklus pengembangan, keamanan dan kualitas perangkat lunak harus menjadi perhatian utama. Tim pengembang harus memastikan bahwa aplikasi e-banking terlindungi dari ancaman siber dan mematuhi regulasi keuangan yang berlaku.

5. Kontrol Versi (Version Control):

- Menggunakan sistem kontrol versi untuk melacak perubahan kode dan dokumentasi selama pengembangan. Hal ini memudahkan kolaborasi tim dan pengembangan yang aman.

6. Manajemen Proyek:

- Menggunakan alat manajemen proyek seperti Jira atau Trello untuk mengawasi kemajuan proyek, mengelola tugas, dan mengkomunikasikan perubahan dalam tim.

7. Uji Coba dan Evaluasi Berkala:

- Menjadwalkan uji coba dan evaluasi berkala untuk memastikan bahwa aplikasi e-banking tetap sesuai dengan kebutuhan pengguna dan mengikuti perkembangan teknologi yang baru.

8. Dokumentasi Proses:

- Selama setiap tahap, dokumen proyek harus diperbarui secara berkala untuk mencerminkan perubahan dan perkembangan. Ini termasuk dokumentasi kebutuhan, desain, dan hasil uji coba.

c. Requirement Engineering:

○ **Identifikasi Stakeholder:**

- Identifikasi pemangku kepentingan termasuk nasabah (perorangan dan bisnis), manajemen bank, pengawas regulasi, tim teknis, dan lembaga keuangan terkait.

○ **Pengumpulan Kebutuhan:**

- **User Stories (Untuk Kebutuhan Pengguna):**
 - Sebagai nasabah, saya ingin dapat melakukan transfer uang ke rekening Bank lain dengan cepat dan aman.
 - Sebagai nasabah bisnis, saya ingin mengelola catatan transaksi dan rekening bisnis saya dengan mudah.
 - Sebagai nasabah, saya ingin melihat grafik pergerakan saham dan berita terkini terkait pasar keuangan.
- **Persyaratan Teknis:**
 - Aplikasi harus mengimplementasikan teknologi enkripsi tingkat tinggi (misalnya, SSL/TLS) untuk melindungi data pengguna.
 - Aplikasi harus mematuhi regulasi keuangan dan privasi data yang berlaku di negara setempat dan internasional.
 - Aplikasi harus memiliki mekanisme pemulihan data dan keandalan tinggi untuk mencegah downtime yang berlebihan.
 - Aplikasi harus mendukung integrasi dengan sistem inti bank, termasuk sistem basis data transaksi dan sistem keamanan.
- **Analisis dan Prioritasi:**
 - Melakukan analisis kebutuhan untuk memahami tingkat prioritas dan ketergantungan antara berbagai kebutuhan.
 - Kebutuhan yang berkaitan dengan layanan inti seperti transfer dana dan manajemen rekening akan memiliki prioritas lebih tinggi.
- **Modeling (Menggunakan UML):**
 - **Use Case Diagram:** Menunjukkan interaksi antara pemangku kepentingan dan sistem. Contoh use case:
 - "Transfer Dana" dengan aktor "Nasabah" dan "Sistem E-Banking."
 - "Manajemen Rekening Bisnis" dengan aktor "Nasabah Bisnis" dan "Sistem E-Banking."

- **Class Diagram:** Menunjukkan entitas dan hubungan antara entitas dalam sistem, seperti "Nasabah," "Rekening," dan "Transaksi."
- **Validasi Kebutuhan:**
 - Melibatkan pemangku kepentingan utama dalam validasi kebutuhan untuk memastikan bahwa mereka memahami dan setuju dengan kebutuhan yang telah didefinisikan. Ini dapat melibatkan pertemuan, demonstrasi prototipe, atau uji coba pengguna.
- **Dokumentasi Kebutuhan:**
 - Membuat dokumen kebutuhan yang komprehensif dan terstruktur, termasuk User Stories, use case, class diagram, dan persyaratan teknis.
 - Dokumen ini akan menjadi dasar bagi tim pengembang untuk memulai pengembangan aplikasi e-banking yang sesuai dengan standar keamanan dan kualitas seperti yang dimiliki oleh bank besar seperti BCA.

d. Perancangan User Interface:

- **Tampilan Utama (Dashboard):**
 - Desain tampilan utama harus memberikan pandangan cepat tentang informasi penting seperti saldo rekening, riwayat transaksi terbaru, dan opsi utama. Gunakan layout yang bersih dan efisien.
- **Navigasi Intuitif:**
 - Menyediakan navigasi yang mudah dimengerti dan intuitif. Gunakan ikon dan label yang jelas untuk menu dan submenu seperti "Transfer Dana," "Pembayaran Tagihan," "Investasi," dan "Manajemen Rekening."
- **Transfer Dana:**
 - Desain tampilan transfer dana yang user-friendly. Sediakan opsi untuk mentransfer ke rekening bank lain, rekening sendiri, atau rekening bisnis.
 - Gunakan alur yang jelas dan langkah-langkah yang mudah diikuti dengan pilihan untuk menyimpan penerima yang sering digunakan.
- **Manajemen Rekening:**
 - Tampilkan grafik visual yang jelas tentang pergerakan saldo rekening dari waktu ke waktu.

- Sediakan opsi untuk melihat detail transaksi dan mengunduh laporan transaksi dalam format yang berbeda.
- **Keamanan:**
 - Tampilkan status keamanan dalam bentuk ikon atau indikator yang jelas. Jika digunakan autentikasi dua faktor, pastikan prosesnya mudah dimengerti oleh pengguna.
 - Berikan informasi tentang langkah-langkah keamanan yang dapat diambil oleh pengguna.
- **Notifikasi:**
 - Tampilkan notifikasi penting seperti pemberitahuan transaksi atau pesan dari bank secara jelas dan terstruktur di area yang mudah ditemukan.
- **Pencarian dan Filter:**
 - Sediakan fitur pencarian yang efisien untuk membantu pengguna menemukan transaksi atau informasi tertentu dengan cepat.
 - Tambahkan opsi filter untuk menyaring transaksi berdasarkan tanggal, jenis transaksi, atau rekening terkait.
- **Konsistensi Desain:**
 - Pastikan keseragaman desain UI di seluruh aplikasi. Gunakan palet warna yang konsisten, tipografi yang serupa, dan elemen desain yang sama.
- **Responsif dan Multi-Platform:**
 - Desain UI harus responsif, artinya harus berfungsi dengan baik di berbagai perangkat seperti smartphone, tablet, dan desktop.
 - Pastikan tampilan mobile-friendly yang optimal.
- **Tes Pengguna (Usability Testing):**
 - Lakukan uji coba pengguna untuk memastikan bahwa desain UI memenuhi kebutuhan nasabah dan mudah digunakan.
 - Ambil umpan balik dari nasabah dalam proses perancangan untuk melakukan perbaikan yang diperlukan.
- **Dokumentasi Desain:**

- Dokumentasikan pedoman desain UI yang menguraikan elemen-elemen desain, palet warna, jenis huruf, dan prinsip navigasi yang harus diikuti oleh pengembang UI selama implementasi.

- **Kontinuitas Perbaikan:**

- Berencana untuk melakukan perbaikan berkelanjutan berdasarkan umpan balik pengguna dan perkembangan teknologi yang baru. Terus perbarui UI sesuai dengan perubahan kebutuhan dan tren.

e. Metrics untuk Mengukur Kualitas Perangkat Lunak:

1. Keamanan (Security Metrics):

- Tingkat Serangan Ditolak: Mengukur jumlah upaya serangan yang berhasil ditolak oleh sistem keamanan perangkat lunak. Semakin tinggi, semakin baik.
- Kerentanan yang Ditemukan: Melacak jumlah kerentanan keamanan yang telah diidentifikasi dan diperbaiki.
- Waktu Respon Terhadap Serangan: Mengukur berapa lama sistem merespons serangan atau insiden keamanan.

2. Kinerja (Performance Metrics):

- Waktu Respon: Mengukur berapa lama aplikasi merespons permintaan pengguna. Semakin cepat, semakin baik.
- Waktu Muat Halaman: Mengukur berapa lama halaman utama aplikasi memuat setelah pengguna masuk. Ini penting untuk pengalaman pengguna yang nyaman.
- Kapasitas Beban: Menentukan seberapa banyak pengguna yang dapat diakomodasi aplikasi tanpa mengalami penurunan kinerja yang signifikan.

3. Kepuasan Pelanggan (Customer Satisfaction Metrics):

- Survei Kepuasan Pelanggan: Melakukan survei reguler kepada nasabah untuk mengukur kepuasan mereka terhadap aplikasi e-banking.
- Tingkat Keluhan Pelanggan: Mengukur jumlah dan jenis keluhan yang diterima dan waktu yang dibutuhkan untuk menyelesaikannya.

4. Kualitas Kode (Code Quality Metrics):

- Jumlah Bug (Bug Count): Mengukur jumlah bug atau masalah dalam kode aplikasi. Semakin sedikit, semakin baik.
 - Tingkat Kode Duplikat: Mengukur sejauh mana kode yang duplikat ada dalam aplikasi. Kode yang lebih bersih memiliki sedikit duplikasi.
 - Kode Sumber Terdokumentasi dengan Baik: Mengukur sejauh mana kode sumber memiliki dokumentasi yang jelas dan berguna.
5. Ketersediaan (Availability Metrics):
- Waktu Downtime: Mengukur berapa lama aplikasi tidak tersedia dalam satu tahun. Semakin sedikit, semakin baik.
 - Waktu Pemulihan: Mengukur berapa lama dibutuhkan untuk memulihkan aplikasi jika terjadi kegagalan atau downtime.
6. Kinerja Layanan (Service Performance Metrics):
- Waktu Transaksi: Mengukur berapa lama dibutuhkan untuk menyelesaikan transaksi perbankan seperti transfer dana atau pembayaran tagihan.
 - Ketersediaan Layanan: Mengukur sejauh mana layanan seperti layanan pelanggan atau bantuan teknis tersedia dan responsif.
7. Kepatuhan (Compliance Metrics):
- Kepatuhan Regulasi: Mengukur tingkat kepatuhan aplikasi terhadap peraturan perbankan dan privasi data yang berlaku.
 - Audit Kepatuhan: Melacak hasil audit internal atau eksternal terkait kepatuhan aplikasi.
8. Pemeliharaan (Maintenance Metrics):
- Waktu Rilis Perangkat Lunak (Software Release Time): Mengukur berapa lama dibutuhkan untuk merilis pembaruan perangkat lunak. Semakin cepat, semakin baik.
 - Rasio Kode Baru terhadap Kode Pemeliharaan: Mengukur perbandingan antara kode baru yang ditambahkan dengan kode yang diperbaiki atau dipelihara. Semakin tinggi rasio kode baru, semakin baik.
9. Evaluasi Pelanggan dan Uji Beta:

- Evaluasi Pelanggan: Melibatkan nasabah dalam uji coba beta untuk mengumpulkan umpan balik tentang kualitas, kinerja, dan keamanan aplikasi.
- Tingkat Partisipasi Uji Beta: Mengukur seberapa banyak nasabah yang berpartisipasi dalam uji coba beta.

10. Pelaporan Masalah dan Perbaikan (Issue Reporting and Resolution):

- Tingkat Pelaporan Masalah: Mengukur jumlah masalah yang dilaporkan oleh pengguna atau pemantauan otomatis.
- Waktu Resolusi Masalah: Mengukur berapa lama dibutuhkan untuk menyelesaikan masalah setelah dilaporkan.