

BAB II

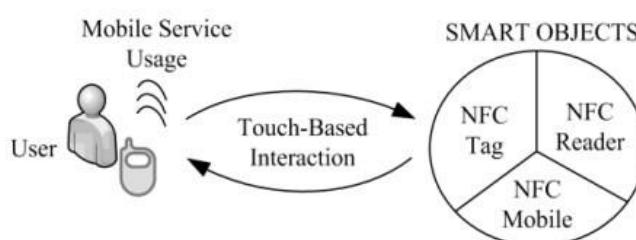
LANDASAN TEORI

2.1. Pengertian Aplikasi

Menurut Jogiyanto (2003), aplikasi adalah perangkat lunak yang digunakan untuk melayani berbagai macam kebutuhan. Agar perangkat keras dapat berfungsi maka perlu diberikan instruksi – instruksi tertentu yang disebut sebagai perangkat lunak.

2.2. Near Field Communication

Menurut Ozdenizci, Alsadi, Ok, & Coskun (2013), Near Field Communication (NFC) adalah teknologi yang memanfaatkan komunikasi nirkabel jarak pendek dan beroperasi pada gelombang 13,56 MHz serta memiliki kecepatan transmisi data hingga 424 kbit per detik dalam jarak kurang dari 10 cm. Penggunaan NFC pada perangkat Android dapat dilakukan secara peer-to-peer yaitu dengan menghubungkan dua perangkat yang sama – sama aktif atau secara satu arah antara tag RFID dengan NFC.



Gambar 2. 1 Komunikasi NFC dengan obyek lain

Pada tahun 2006, perusahaan Nokia memperkenalkan Nokia 6131 sebagai ponsel genggam pertama yang telah ditanam NFC di dalamnya. Inisiasi yang dilakukan oleh Nokia diikuti oleh perusahaan smartphone lain pada tahun berikutnya. Pada tahun 2013, perusahaan ponsel besar dari Korea, Samsung,

bersama Visa memulai pengembangan NFC untuk digunakan sebagai alat pembayaran bergerak. Google kemudian memperkenalkan sistem pembayaran berbasis NFC yang diberi nama Android Pay menjelang akhir tahun 2015. Sistem pembayaran ini membutuhkan smartphone yang telah ditanami NFC untuk kemudian didekatkan dengan mesin pembayaran elektronik. Android Pay saat ini baru tersedia di Amerika Serikat dan akan diterapkan di Australia pada tahun 2016. Di Indonesia, Telkomsel adalah pionir yang telah menerapkan NFC untuk mobile payment pada kartu TCash mereka.

Sampai saat ini, pengembangan NFC terus dilakukan terutama di bidang akademik yang berhubungan dengan NFC. Sebagai contoh, di bidang layanan kesehatan NFC mampu menyediakan pemantauan dan pengendalian kesehatan jarak jauh. Dan juga terdapat layanan yang memberikan perbaikan kualitas hidup seseorang dengan menggunakan sistem resep berbasis NFC dan penyimpanan data medis yang terenkripsi pada NFC tag. Selain layanan kesehatan, NFC memberikan kemudahan lain bagi pengguna yang ingin membeli tiket bioskop, konser, transportasi umum hingga pemesanan kamar hotel.

2.3. Radio Frequency Identification (RFID)

Menurut Simoes, Rodrigues, Veiga, & Ferreira (2011), Radio Frequency Identification (RFID) adalah teknologi yang memanfaatkan medan elektromagnetik yang secara otomatis mengidentifikasi dan melacak *tags* yang menempel pada suatu obyek. RFID modern pertama kali didemonstrasikan pada tahun 1973 dengan menggunakan gelombang 915 Mhz dan 12-bit tags.

Pada dasarnya, teknologi RFID terdiri dari tag yang mengidentifikasi manusia, hewan atau benda yang dapat mengirim maupun menerima sinyal

elektromagnetik. Cara kerja RFID terdapat dua macam, aktif dan pasif. Pada mode aktif, RFID bekerja dengan memancarkan gelombang elektromagnetik. Sedangkan pada mode pasif, RFID akan menunggu sampai ada gelombang elektromagnetik yang diterima. Beberapa kelebihan RFID adalah:

1. Adanya kemungkinan untuk membedakan setiap tag dengan informasi yang spesifik.
2. Dapat memberi ijin akses atau baca kepada setiap tag.
3. Dapat melakukan pertukaran informasi antara RFID pemancar dengan RFID penerima tanpa melakukan sentuhan sedikitpun.

Meskipun demikian, terdapat beberapa kekurangan dari teknologi RFID, antara lain :

1. Adanya dua atau lebih sinyal yang saling memancarkan, dapat mengganggu penerima gelombang yang mengakibatkan ketidaksesuaian respon yang dihasilkan.
2. Data yang disimpan pada RFID bersifat statis dan terbatas, bergantung pada kapasitas memori kartu atau tag.

2.4. Android

Menurut Kasman (2013), Android merupakan sistem operasi yang dikembangkan oleh Google yang berbasis Linux dan didesain untuk perangkat layar sentuh seperti smartphone atau pun komputer tablet. Android memiliki perkembangan yang begitu cepat karena Google memberikan kebebasan bagi para pengembang untuk bisa leluasa menciptakan aplikasi di berbagai perangkat.

Pemberian nama versi Android menggunakan sistem nama hidangan penutup. Dimulai dengan menggunakan nama Android Cupcake sebagai versi

pertamanya, kini Android telah merilis versi ke enam dengan nama Android Marshmallow. Pada tahun 2015, Android menjadi sistem operasi yang terpopuler digunakan oleh smartphone dan tablet.

2.5. Unified Modeling Language (UML)

Menurut Dennis, Wixom, & Tegarden (2009), *Unified Modeling Language* (UML) diperkenalkan oleh Grady Booch, Ivar Jacobson, dan James Rumbaugh pada tahun 1995. UML adalah standar yang menyediakan teknik untuk memodelkan pengembangan sistem informasi dengan pendekatan berorientasi objek. Diagram yang disediakan oleh UML antara lain :

1. Diagram *use case*
2. Diagram aktivitas
3. Diagram sekuensial
4. Diagram kolaborasi
5. Diagram kelas
6. Diagram komponen, dan
7. Diagram *deployment*

2.5.1 Diagram Use Case

Diagram *use case* adalah potongan fungsionalitas tingkat tinggi yang disediakan oleh sistem (Sholiq, 2010). Diagram ini menunjukkan beberapa *use case* dalam sistem, beberapa aktor yang mengakses sistem dan relasi yang dilakukan antar aktor dengan *use case*. Dalam sebuah sistem, mungkin untuk terdapat lebih dari satu *use case*. *Use case* yang dibuat difungsikan dengan tujuan tertentu. Mengelompokkan *use case* menjadi beberapa paket – paket tertentu menjadikan susunan *use case* lebih mudah terbaca. *Use case* juga dapat

menggambarkan fungsi bisnis tertentu dalam sebuah organisasi seperti Departemen Keuangan, Departemen Produksi, dan lain lain.

2.5.2 Diagram Aktivitas

Diagram aktivitas adalah sebuah cara untuk memodelkan alur kerja dari *use case* bisnis dalam bentuk grafik (Sholiq, 2010). Dalam diagram aktivitas, proses yang dilakukan oleh aktor dalam *use case* bisnis digambarkan lebih detail sehingga memudahkan untuk dibaca.

2.5.3 Diagram Sekuensial

Diagram sekuensial adalah diagram interaksi yang disusun berdasarkan urutan waktu (Sholiq, 2010). Cara membaca diagram ini adalah dari atas ke bawah. Diagram sekuensial disusun dari *use case* yang telah dibuat alur utama maupun alur alternatifnya. Sehingga diagram sekuensial akan merepresentasikan satu atau beberapa alur dari *use case*.

2.5.4 Diagram Kolaborasi

Diagram kolaborasi menampilkan alur skenario tertentu dalam sebuah *use case* (Sholiq, 2010). Jika diagram sekuensial disusun berdasarkan urutan waktu, maka diagram kolaborasi disusun berdasarkan hubungan antara obyek – obyek yang membangunnya.

2.5.5 Diagram Kelas

Diagram kelas digunakan untuk menampilkan kelas – kelas atau paket – paket dalam sebuah sistem dan menampilkan hubungan yang terjadi antara kelas – kelas tersebut (Sholiq, 2010). Suatu diagram kelas merupakan turunan dari diagram kelas yang lain. Dalam diagram kelas, terdapat atribut dan fungsi – fungsi yang dimiliki kelas tersebut maupun dari kelas lain.

2.5.6 Diagram Komponen

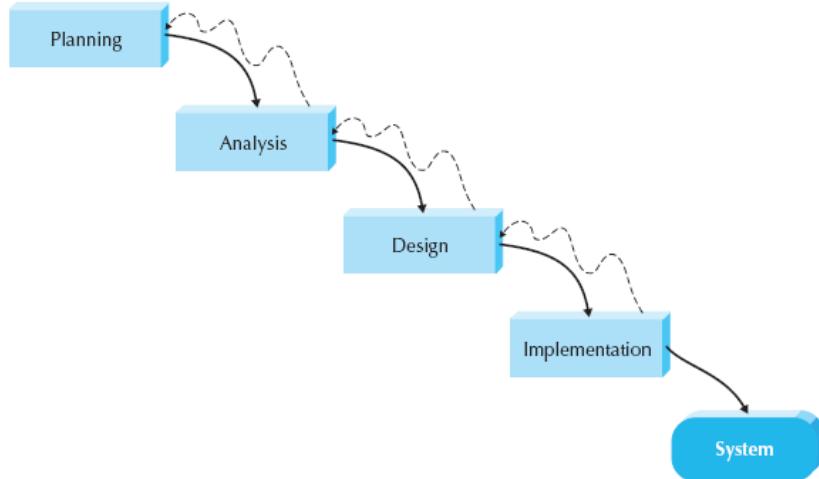
Menurut Sholiq (2010), diagram komponen adalah diagram yang memodelkan modul fisik dari kode yang dibangun. Fungsi dari diagram ini adalah agar *programmer* mengetahui komponen mana yang harus dilakukan kompilasi kode terlebih dahulu, mengingat setiap komponen yang dilakukan kompilasi bersifat dependen dengan komponen yang lain.

2.5.7 Diagram Deployment

Diagram *deployment* adalah segala hal yang berkaitan dengan penyebaran fisik aplikasi (Sholiq, 2010). Diagram *deployment* juga menggambarkan *layout* jaringan serta lokasi komponen – komponen jaringan. Perangkat keras yang digunakan oleh aplikasi juga digambarkan pada diagram ini.

2.6. System Development Life Cycle (SDLC)

Menurut Dennis et al. (2009), Software Development Life Cycle (SDLC) adalah proses memahami bagaimana sistem informasi dapat mendukung kebutuhan bisnis dengan cara menganalisa dan membangun sistem. SDLC membagi pengembangan sistem informasi ke dalam beberapa tahapan, mulai dari analisa hingga implementasi.



Gambar 2. 2 Tahapan dalam SDLC

2.6.1 Tahapan Perencanaan

Menurut Dennis et al. (2009), *Software Development Life Cycle (SDLC)* adalah proses memahami bagaimana sistem informasi dapat mendukung kebutuhan bisnis dengan cara menganalisa dan membangun sistem. SDLC membagi pengembangan sistem informasi ke dalam beberapa tahapan, mulai dari analisa hingga implementasi.

Menurut Dennis et al. (2009), Tahap perencanaan merupakan proses untuk memahami mengapa sistem informasi perlu dibangun dan bagaimana tim proyek mengerjakannya. Terdapat dua fase dalam tahap perencanaan, yaitu:

1. Fase inisisasi proyek, dan
2. Fase manajemen proyek

2.6.2 Tahapan Analisis

Menurut Dennis et al. (2009), Tahapan analisis berguna untuk menjawab siapa yang menggunakan sistem, apa yang dilakukan oleh sistem, kapan dan dimana sistem tersebut digunakan. Pada tahapan ini, tim proyek mengidentifikasi

keadaan sistem yang ada dan mencari kelemahan sistem untuk kemudian dikembangkan ke arah yang lebih baik dengan sistem yang baru.

Dalam melakukan analisis sistem, pengembang diharapkan memiliki kemampuan untuk menerjemahkan kebutuhan pengguna dan menemukan peluang yang dapat dilakukan oleh sistem. Pengembang melakukan langkah – langkah untuk menetukan kebutuhan pengguna, teknik dan strategi yang diperlukan dalam melakukan analisis sistem, dan memodelkan konsep serta kebutuhan pengguna ke dalam rancangan yang lebih spesifik dan nyata.

2.6.3 Tahapan Desain

Menurut Dennis et al. (2009), Tahapan ini menentukan bagaimana nantinya sistem akan bekerja, menentukan kebutuhan hardware, software, dan infrastruktur yang dibutuhkan. Termasuk dalam tahapan ini adalah merancang user interface, bentuk laporan dan juga struktur basis data yang digunakan. Terdapat empat langkah pada tahapan ini, yaitu :

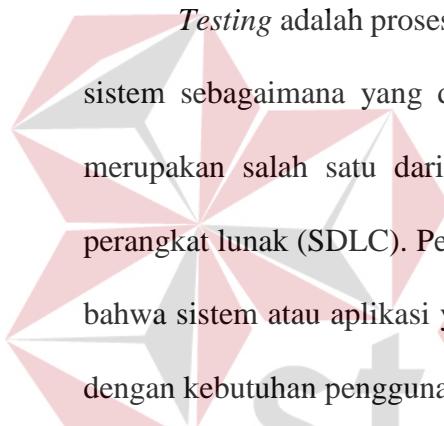
1. Membuat strategi yang digunakan dalam mendesain sistem.
2. Mengembangkan desain arsitektur untuk sistem.
3. Membangun basis data dan *file specification*.
4. Membuat desain program.

2.6.4 Tahapan Implementasi

Tahapan akhir dari SDLC adalah tahapan implementasi dimana sistem mulai dibuat. Tahap implementasi biasanya membutuhkan waktu yang lebih lama dari pada tahapan yang lain, serta membutuhkan biaya yang lebih besar. Pada tahap ini, tim proyek melakukan konstruksi sistem, melakukan uji coba, membuat dokumentasi hingga melakukan pemasangan sistem sesuai permintaan pengguna.

Pengembang dalam melakukan implementasi sistem yang baru juga perlu untuk melakukan perencanaan migrasi dari sistem yang lama ke sistem yang baru. Pengembang diharapkan akan mampu menentukan strategi atau metode apa saja yang diperlukan agar sistem yang baru dapat berjalan dengan baik dan dapat diterima oleh penggunanya. Selain itu, pengembang juga harus selalu memberikan dukungan atau *support* kepada pengguna atas hal – hal yang terjadi setelah sistem berhasil diimplementasikan.

2.7. Pengujian



Testing adalah proses pemantapan kepercayaan akan kinerja program atau sistem sebagaimana yang diharapkan (Romeo, 2003). *Testing* atau pengujian merupakan salah satu dari beberapa tahapan yang ada pada pengembangan perangkat lunak (SDLC). Pengujian memberikan keyakinan kepada pengembang bahwa sistem atau aplikasi yang dibuat dapat digunakan dengan baik dan sesuai dengan kebutuhan pengguna.

Dalam pengujian, terdapat beberapa macam cara atau metode yang dilakukan oleh pengembang, antara lain adalah *Black Box Testing* dan *White Box Testing*. *Black Box Testing* digunakan oleh pengembang untuk mengetahui kesalahan yang akan mungkin muncul dari aplikasi tanpa harus mengetahui kode – kode dari aplikasi. Pengujian tipe ini dapat dilakukan oleh orang yang tidak ikut membangun sistem atau aplikasi.

Sedangkan *White Box Testing* merupakan metode pengujian dengan desain test case yang menggunakan struktur kendali dari desain prosedural (Romeo, 2003). Dalam pengujian jenis ini, tes dilakukan oleh pengembang sistem atau aplikasi itu sendiri yang lebih mengetahui kode atau desain dari aplikasi.

Pengujian *White Box* memastikan bahwa alur logika program telah sesuai dan struktur internal data telah valid.

