



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB III

METODE PENELITIAN

3.1. Objek Penelitian

Objek penelitian ini adalah perancangan aplikasi yang membantu metode akses materi oleh mahasiswa-mahasiswi Universitas Multimedia Nusantara terhadap mata kuliah yang sedang dipelajari pada satu kelas dan jadwal tertentu. Dengan adanya aplikasi tersebut, mahasiswa tidak lagi datang ke kelas tanpa materi pembelajaran karena aplikasi dapat menyajikan materi-materi tadi secara *seamless*. Ada pula motif mengapa dipilihnya objek ini yakni metode akses materi yang efisien dan kontekstual dapat meningkatkan pemahaman mahasiswa terhadap mata kuliah yang sedang dipelajari tanpa mengganggu proses belajar dengan alasan-alasan kurang persiapan ataupun sulitnya mengakses materi.

Penggunaan alat *bluetooth beacon* menjadi satu-satunya cara untuk menjembatani konteks dunia nyata tersebut dengan aplikasi pada *mobile platform*. Jarak deteksi yang dapat diatur dan tidak dibutuhkannya pandangan secara langsung (*line of sight*) dengan gawai mobile menjadi alasan utama digunakannya alat *bluetooth beacon*. Berikut merupakan perbandingan *bluetooth beacon* dengan piranti dengan tujuan serupa menurut Statler (2016) pada tabel 3.1.

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A

Tabel 3. 1. Perbandingan Piranti Serupa

No.	Piranti	Kelebihan	Kekurangan
1.	Bluetooth Beacon	Jarak sinyal mencapai 70 sampai 200 meter dan dapat diatur penerimaannya; Tidak memerlukan <i>line of sight</i> langsung dengan gawai <i>mobile</i> ; Mayoritas gawai baru dengan dukungan Bluetooth 4 atau lebih dapat menerima sinyal <i>bluetooth beacon</i> ;	Membutuhkan biaya untuk pengadaan alat; Membutuhkan aplikasi tambahan untuk interpretasi sinyal dan informasi dari <i>beacon</i> ; Gawai yang sudah berumur mayoritas belum mendukung Bluetooth versi 4.
2.	Quick Response (QR) Code	Hemat biaya karena dapat dicetak sendiri; Bentuk, warna, ukuran dapat dimodifikasi sesuai identitas dari layanan.	Membutuhkan adanya <i>line of sight</i> langsung dengan kamera pada gawai <i>mobile</i> ; Jarak deteksi terbatas karena kamera harus dapat menangkap bentuk kode dengan jelas.
3.	Radio Frequency Identification (RFID)	Jarak sinyal mencapai 3 sampai 300 meter untuk jenis RFID aktif; Tidak memerlukan <i>line of sight</i> langsung dengan alat penerima.	Membutuhkan biaya untuk ekosistem pemancar; Gawai <i>mobile</i> tidak dapat menerima sinyal RFID secara langsung tanpa alat tambahan;
4.	Near Field Communication (NFC)	Tidak memerlukan <i>line of sight</i> langsung dengan NFC <i>reader</i> ; Mayoritas gawai <i>mobile</i> baru telah mendukung teknologi NFC <i>reader</i> .	Membutuhkan biaya untuk pemancar sinyal NFC; Jarak deteksi sangat terbatas karena gawai harus didekatkan minimal 8 inci dari pemancar.

Sumber: (Statler, 2016)

3.2. Metode Pengembangan Aplikasi

Dalam pelaksanaan pengembangan aplikasi ini, terdapat beberapa metode pengembangan yang dapat dipilih berdasarkan tujuannya masing-masing. Menurut Amlani (2012), terdapat pula beberapa kelebihan dan kekurangan dari masing-masing metode pengembangan aplikasi, dapat dilihat pada tabel 3.2.

Tabel 3. 2. Perbandingan Metode Pengembangan Aplikasi

No.	Metode	Kelebihan	Kekurangan
1.	Model <i>Waterfall</i>	Tinggi pemahaman akan kebutuhan perusahaan; Lengkapnya dokumentasi akan membantu kebutuhan revisi maupun proyek lain di masa depan; Setiap fase menjamin kesempurnaan hasil; Sumber daya terhitung rendah dibanding model lain.	Tidak dapat mengulang fase yang sudah dilakukan, sekali selesai artinya fase tersebut telah dikunci; Tidak memperbolehkan terjadinya perubahan kebutuhan klien saat pengembangan aplikasi telah dimulai; Meskipun rendah sumber daya, bisa terjadi adanya tim yang menganggur saat fase bukan tanggung jawabnya berjalan.
2.	Model <i>V-Shape</i>	Setiap fase menjamin adanya produk yang dihasilkan; Aktivitas <i>testing</i> dilakukan sebelum dimulainya <i>coding</i> , dimana akan menghemat waktu dan meningkatkan pemahaman kebutuhan saat awal proyek; Tingginya rasio kesuksesan karena perencanaan <i>testing</i> sudah dimulai sejak awal proyek.	Sangat tidak fleksibel; Aplikasi baru dimulai dikembangkan saat fase implementasi, tidak ada prototipe awal sama sekali; Tidak dapat mengakomodasi fase-fase yang berjalan bersamaan; Model tidak menyediakan solusi yang jelas ketika menghadapi masalah saat pengembangan aplikasi.
3.	Model <i>Prototype</i>	Klien mendapatkan banyak keuntungan dalam fase ini, dimana prototipe tersedia dengan cepat dan klien dapat menilai kemampuan <i>developer</i> maupun dapat merasakan fungsionalitas aplikasi secara langsung untuk memberi <i>feedback</i> yang jelas; Rendahnya resiko kegagalan berkat banyaknya prototipe; <i>Feedback</i> dari klien didapatkan secara periodik,	Proses memakan waktu yang cukup panjang; Sumber daya utama (pengembang) yang sedikit dan tingginya biaya yang dibutuhkan; Bisa terjadi kesalahpahaman dimana klien menganggap prototipe sebagai produk final; Kadang, tingginya keterlibatan klien menjadi hambatan bagi pengembang aplikasi untuk bekerja.

Tabel 3. 2. Perbandingan Metode Pengembangan Aplikasi (Lanjutan)

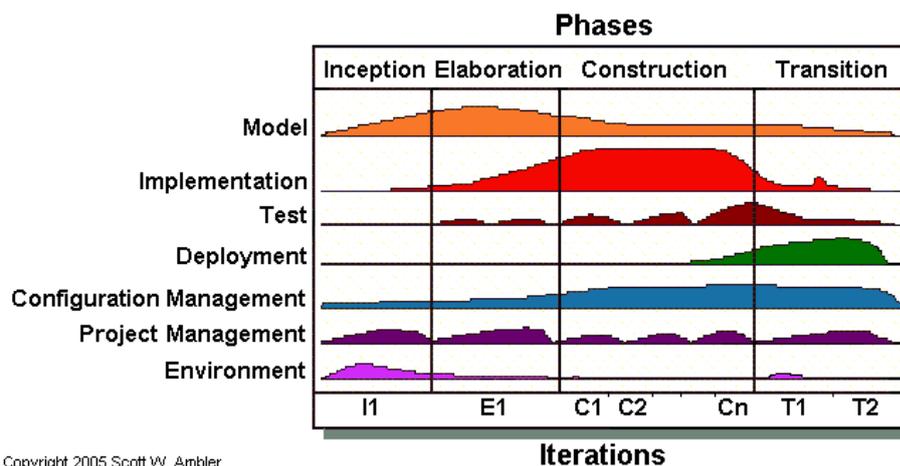
No.	Metode	Kelebihan	Kekurangan
		artinya pengembang dapat lebih mengerti akan kebutuhan klien seiring iterasi prototipe.	
4.	Model <i>Incremental</i>	Sangat fleksibel untuk mengakomodasi perubahan kebutuhan klien; Mudah untuk melakukan <i>testing</i> karena setiap iterasi telah dilakukan <i>testing</i> independen; Manajemen resiko yang lebih baik karena dapat dikonfirmasi antara kebutuhan klien dengan setiap iterasi baru; Setiap <i>increment</i> tidak memerlukan sumber daya yang tinggi.	Perubahan dalam setiap iterasi relatif kecil; Bisa terjadi masalah pada arsitektur sistem karena tidak semua <i>requirements</i> diketahui pada fase awal iterasi; Iterasi tidak dapat dilakukan dalam satu <i>increment</i> ; Dokumentasi lebih berat karena dibutuhkan <i>mapping requirement</i> ke masing-masing <i>increment</i> .
5.	Model <i>Agile Unified Process</i>	Hemat biaya dan waktu; Bersifat iteratif, menyediakan <i>feedback</i> berkelanjutan dari klien, sehingga lebih mudah untuk beradaptasi dengan perubahan yang konstan; Sangat cocok untuk teknologi <i>web</i> yang berubah cepat seiring waktu atau proyek media lainnya; Berkfokus pada <i>update</i> yang cukup sering sehingga memberikan detail pencapaian proyek yang jelas;	Apabila tugas utama tidak didefinisikan dengan baik, estimasi biaya dan waktu tidak akan akurat; Hanya cocok untuk proyek kecil, cepat bergerak, dengan tim kecil pula; Membutuhkan semua anggota tim yang kompeten di bidangnya. Apabila ada anggota awam, proyek tidak akan selesai tepat waktu.

Sumber : (Amlani, 2012)

Untuk mengakomodasi kebutuhan pengembangan aplikasi pada penelitian ini, metode pengembangan berbasis model *agile unified process* akan digunakan.

Terdapat beberapa faktor yang mendukung, yakni:

- I. Rancangan aplikasi memiliki persyaratan utama dengan batasan masalah yang jelas, sehingga dapat menjaga lingkup pengerjaan berbasis metode *agile* dengan fokus.
- II. Rentang waktu penelitian yang terbilang singkat (4 bulan) sesuai dengan metode *agile* karena dapat memberikan hasil aplikasi yang berjalan fungsi utamanya dengan cepat karena tidak harus 100% sempurna.
- III. Metode *agile* terbilang fleksibel untuk menangani masalah di tengah proses pengembangan aplikasi. Masalah-masalah seperti tidak didukungnya *plugin* tertentu, penemuan *error* saat tahap *testing*, tidak kompatibelnya aplikasi dengan gawai tertentu, dan sebagainya dapat diselesaikan dengan segera pada iterasi berikutnya tanpa mengulangi keseluruhan siklus dari awal kembali.



Gambar 3. 1. Model *Agile Unified Process*

Sumber : (Ambler, 2005)

Terdapat empat tahapan utama dalam model *Agile Unified Process*, dimana seluruh fase dapat dilakukan berulang-ulang prosesnya (disebut iterasi ke-n) hingga mencapai seluruh persyaratan yang paling mutakhir, seperti pada gambar 3.1. Berikut penjelasan dari masing-masing fasenya:

1. *Inception*

Pada tahap ini dilaksanakan pemetaan kondisi oleh pengembang aplikasi terhadap objek penelitian, dalam hal ini adalah mahasiswa Universitas Multimedia Nusantara. Untuk mengetahui betul aspek apa saja yang harus diperhatikan, dilakukan observasi langsung terhadap proses belajar mengajar di Universitas Multimedia Nusantara. Hasil dari observasi tersebut adalah gambaran *rich picture* mengenai pola pembelajaran yang sudah ada beserta deskripsinya. Selain observasi, tinjauan pustaka maupun penemuan-penemuan yang muncul pada saat pengembangan aplikasi akan menjadi informasi penting pada iterasi berikutnya.

2. *Elaboration*

Hasil dari observasi dipetakan ke dalam sebuah diagram alir untuk memberikan gambaran alur kerja dari cara mahasiswa mengakses materi secara konvensional saat masa perkuliahan. Keseluruhan informasi-informasi tadi akan dipakai untuk menarik kesimpulan apa saja yang harus menjadi sorotan pada tujuan aplikasi. Misal, ketika mahasiswa menyatakan bahwa benar mereka mengalami hambatan waktu dalam mengakses materi maka analisisnya adalah aplikasi harus bisa memberikan solusi berupa *trigger bluetooth beacon* untuk menampilkan materi yang relevan dan kontekstual

sesuai kebutuhan mahasiswa tadi saat masa perkuliahan. Selain itu, data-data lain yang akan muncul pada iterasi selanjutnya juga menjadi pedoman analisis pada iterasi selanjutnya pula.

Pada tahap desain, hasil dari analisis diterjemahkan ke berbagai dokumentasi dalam bentuk: *use case diagram*; *entity relationship diagram*; dan *activity diagram*. *Tool* yang digunakan dalam menerjemahkan aplikasi adalah StarUML. Sistem basis data juga tak luput dari tahap ini, dimana berdasar *entity relationship diagram* yang telah dirancang akan diterapkan ke dalam bentuk skema basis data. Skema basis data model *relationship* akan dipakai, sedang *tools* utama yang akan digunakan adalah phpMyAdmin dan MySQL Workbench.

Selain kebutuhan dokumentasi dan basis data, pengembang aplikasi juga diharuskan untuk mempersiapkan *mock-up user interface* yang dapat menunjang *user experience* terbaik dalam interaksi antara mahasiswa dengan aplikasinya nanti. *Mock-up user interface* aplikasi didesain menggunakan Adobe XD dengan Adobe Photoshop maupun Illustrator untuk mempersiapkan aset-asetnya.

3. *Construction*

a. *Code*

Pada tahap penulisan *source code*, pengembang aplikasi menerjemahkan seluruh modul yang dirancang pada tahap desain menjadi bentuk aplikasi konkret yang dapat dipakai. Terdapat tiga sektor yang harus diperhatikan dalam tahap ini: *front-end*, *back-end*, dan *supporting system*.

Front-end dalam hal ini adalah aplikasi *mobile* berbasis sistem operasi Android, yang akan dirancang menggunakan *tool* Android Studio. *Back-end* dalam hal ini adalah aplikasi web berbasis CodeIgniter yang menggunakan plugin GroceryCRUD sebagai pengolah database dan CI REST API sebagai pelayan *application programming interface*, yang akan dirancang menggunakan *tool* Sublime Text. Terakhir, *supporting system* akan dikelola menggunakan *dashboard* yang telah disediakan oleh Estimote, yaitu Estimote Cloud. Pada iterasi selanjutnya, modul-modul pemrograman yang telah ditulis dapat digunakan kembali maupun dimodifikasi untuk memenuhi perubahan.

b. *Test*

Aplikasi yang sudah jadi pada satu iterasi harus menghadapi fase *testing*, dimana aplikasi dijalankan dan diperiksa hasilnya sesuai ekspektasi fungsinya. Dalam melaksanakan fase *testing*, data-data seperti identitas user, material mahasiswa, konteks penghubung material dengan jadwal maupun ruangan, dan lain-lain juga wajib untuk dimuat ke dalam aplikasi. Pada akhirnya, hasil *testing* untuk membandingkan kenyataan dan ekspektasi akan ditampilkan pada sebuah tabel berdasar skenario penggunaan fungsi dari aplikasi yang dirancang.

4. *Transition*

Aplikasi yang telah selesai diuji disediakan bagi pengguna untuk kemudian disimpulkan bahwa aplikasi telah siap di lingkungan kampus atau harus menempuh tahap selanjutnya untuk mengulang iterasi yang lebih baik.

Adapun masalah maupun saran yang muncul setelah implementasi aplikasi tersebut akan dijadikan dasar-dasar utama pada iterasi selanjutnya.

3.3. Metode Pengujian Aplikasi

Dalam rangka menjawab rumusan masalah kedua, dilakukan uji coba untuk membandingkan kelas yang dianjurkan menggunakan aplikasi dan yang tidak. Adapun variabel yang diperhatikan adalah sebagai berikut:

- i. Variabel bebas adalah metode akses materi perkuliahan.
- ii. Variabel terikat adalah hasil nilai dari kelas yang dianjurkan menggunakan aplikasi mobile dan tidak menggunakan aplikasi mobile.
- iii. Variabel kontrol adalah kelas Konsep Sistem Operasi dengan jumlah 40 mahasiswa yang dihadapkan pada kuis tanpa menggunakan aplikasi *mobile*.

Tahap pertama adalah melakukan pengenalan terhadap dua kelas Konsep Sistem Operasi untuk melakukan instalasi dan mencoba aplikasi pada minggu ke-13. Pada waktu tersebut juga mahasiswa kelas terkait akan dianjurkan untuk menggunakan aplikasi *mobile* dalam proses belajar mengajar pada pertemuan selanjutnya dan diberikan sedikit informasi mengenai kuis yang akan dilaksanakan pada pertemuan selanjutnya. Setelah pengenalan selesai, materi yang akan diujikan dalam kuis diunggah ke aplikasi *back-end* sesuai dengan jadwal kelas tersebut. Pada minggu ke-14, *bluetooth beacon* diposisikan di dalam kelas terkait dan dilaksanakan proses belajar mengajar beserta kuis. Sebagai catatan, materi yang dipersiapkan pada aplikasi *mobile* juga diajarkan pada hari tersebut. Adapun soal

kuis yang akan diujikan juga diperlihatkan pada saat dosen mengajarkan materi tersebut.

Setelah seluruh kuis selesai dinilai, dilakukan uji statistik *two sample t-test* terhadap kondisi kelas yang menggunakan aplikasi mobile dan yang tidak. Karena kelas yang diujikan hanya mendapat anjuran bukan kewajiban, maka data tidak akan diuji secara mentah. Pengolahan data kondisi kelas pengguna aplikasi *mobile* dilakukan sepuluh kali untuk mendapat rata-rata hasil uji. Hipotesis pertama adalah nilai kuis kelas yang menggunakan aplikasi lebih tinggi secara signifikan daripada kelas yang tidak. Hipotesis null adalah tidak terdapat perbedaan yang signifikan antara kelas yang menggunakan aplikasi dan yang tidak. Nilai signifikansi yang dipakai adalah 0,05 dan tingkat kepercayaan yang digunakan adalah 95%. Baru kemudian disimpulkan pernyataan berdasarkan rata-rata hasil uji statistik

