

Implementasi *Mobile-D* Dalam Pengembangan Aplikasi *Mobile* Berbasis Android

Firmansyah¹, Agus Yulianto², Dikdik Permana Wigandi³

¹Program Studi
Manajemen Informatika
AMIK BSI Jakarta
firmansyah.fmh@bsi.ac.id

^{2,3}Program Studi
Sistem Informasi
STMIK Nusa Mandiri Jakarta
²agus.aag@nusamandiri.ac.id
³dikdik.dkn@nusamandiri.ac.id

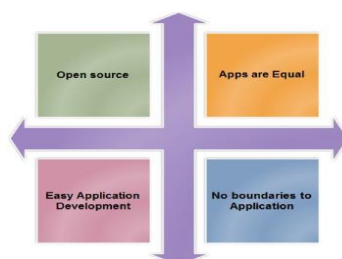
Abstrak - Perkembangan teknologi semakin menuntun pengembang aplikasi khususnya berbasis Android untuk dapat membuat aplikasi yang lebih cepat, memudahkan bagi developer dan dengan tahapan yang lebih rinci komprehensif setiap fase nya, *Mobile-D* merupakan salah satu metode pengembangan aplikasi khusus berbasis mobile dimana prakteknya didasarkan pada pengembangan metode tangkas. Android merupakan sistem operasi sumber terbuka yang memiliki empat fitur yaitu sumber terbuka, pengembangan aplikasi yang mudah, aplikasi sama yang digunakan dan tidak ada batasan dalam membangun aplikasi sehingga dengan mudah siapapun dapat mengembangkan aplikasi berbasis Android. Metode pengembangan *Mobil-D* memiliki lima siklus yaitu *Explore, Initialize, Productionize, Stabilize* dan *System Test and Fix*. Dalam pembahasan ini akan membahas implementasi metode *Mobile-D* digambarkan dengan membangun aplikasi *cleaning complaint report* dimana aplikasi digunakan untuk komplain terkait dengan kebersihan pesawat dalam penerbangan komersial, pembangunan aplikasi menggunakan operating system *Android* sehingga dapat menghasilkan aplikasi yang diharapkan dengan menganalisa beberapa kebutuhan yang ada pada penerbangan komersial dapat menghasilkan aplikasi yang dibutuhkan.

Kata Kunci: *Mobile-D*, Pengembangan Aplikasi, Android.

PENDAHULUAN

Semakin banyaknya perangkat mobile saat ini berdampak pada kebutuhan akan aplikasi yang bisa berjalan di sistem operasi mobile seperti Android, Apple, Windows dan sistem operasi mobile lainnya. Kebutuhan akan aplikasi yang bisa berjalan di perangkat mobile, juga berdampak akan kebutuhan pengembang aplikasi mobile (Flora & Chande, 2013).

Android merupakan pilihan banyak pengembang dikarenakan sifatnya sumber-terbuka, sehingga siapapun dapat mengembangkan aplikasi Android. Untuk membangun aplikasi Android dibutuhkan bahasa pemrograman Java yang juga bahasa sumber-terbuka. Android terbagi menjadi 4 fitur inti, seperti gambar di bawah ini (Goggin, 2012).



Gambar 1: Empat fitur Android

1. Metode *Mobile-D*

Sejak metode *waterfall* untuk pengembangan perangkat lunak diperkenalkan pertama kali ke publik oleh Winston Royce pada tahun 1970 (Wescon, 1970), hingga saat ini banyak metode yang dibuat dan diperkenalkan oleh beberapa peneliti di bidang pengembangan perangkat lunak. Setiap metode memiliki perbedaan, mulai dari karakter, proses, fase, dokumentasi dan peran.

Setiap metode dalam pengembangan perangkat lunak juga memiliki kelebihan dan kekurangan. Dalam studinya Despa, dipaparkan bahwa setiap metode memiliki perbedaan karakter, kekuatan dan kelemahan (Livi Despa, 2014). Pemilihan metode yang baik bukan bergantung pada bagus atau tidaknya sebuah metode namun bergantung pada jumlah tim yang akan terlibat, dokumentasi yang dibutuhkan, waktu penyelesaian proyek dan jenis perangkat lunak yang akan dikembangkan.

Metode *Mobile-D* adalah salah satu metode yang cocok untuk pengembangan aplikasi *mobile* karena bersifat tangkas (*agile*) dan fleksibel. *Mobile-D* merupakan pengembangan dari beberapa *framework* yaitu Extreme Programming, Crystal dan Rationale

Unified Process. Banyak alasan mengapa untuk pengembangan *mobile* membutuhkan metode tangkas, metode tangkas memiliki karakteristik yang cocok untuk pengembangan *mobile* seperti dapat berjalan pada lingkungan sistem yang sering berubah, jumlah tim yang kecil, mampu mengidentifikasi pengguna, lingkungan sistem berorientasi objek, aman, berada pada level aplikasi, sistem yang dibangun kecil dan waktu pengembangan yang relatif pendek (Abrahamsson et al., 2004). Berikut merupakan siklus dari Mobile-D :



Gambar 2: Tahapan Mobile-D

1. *Explore*

Tahap ini merupakan tahap untuk menentukan platform yang akan dipakai, pengguna yang akan menggunakan aplikasi, infrastruktur, bahasa pemrograman, kebutuhan aplikasi.

2. *Initialize*

Melakukan penyiapan sebelum tahap produksi, hal yang harus dilakukan adalah menentukan kebutuhan, product backlog, dan sprint planning.

3. *Productionize*

Tahap pengembangan aplikasi yang terdiri dari 3 tahap, hari perencanaan, hari kerja dan hari rilis aplikasi.

4. *Stabilize*

Memastikan bahwa aplikasi stabil di perangkat dan sistem operasi.

5. *System Test and Fix*

Melakukan tahap testing sebelum rilis, mulai dari pengujian internal dan pengujian external.

METODOLOGI PENELITIAN

Metode penelitian yang digunakan dalam melakukan design sampai dengan implementasi menggunakan Mobile-D method dan pengumpulan data menggunakan observasi dan wawancara.

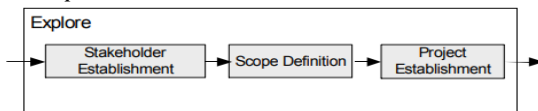
A. Pembahasan Metode Mobile-D.

Android memiliki empat fitur yang dapat digunakan sebagai acuan pengembangan sistem yaitu

1. Sumber terbuka
2. Pengembangan aplikasi yang mudah
3. Aplikasi sama
4. Tidak ada batasan untuk aplikasi

Metode yang akan digunakan dalam pengembangan aplikasi mobile berbasis Android adalah Mobile-D dengan 5 fase yaitu *explore*, *initialize*, *productionize*, *stabilize*, *system and fix*.

1. *Explore*



Gambar 3: Fase Explore

a. *Stakeholder establishment*

Langkah pertama, identifikasi pemilik produk. Langkah ini bertujuan untuk mengidentifikasi pengguna yang akan bertanggung jawab dan berkomunikasi kepada tim proyek mengenai detail produk. Dalam proyek ini melibatkan 1 orang sebagai pengguna yang mewakili dan akan berkomunikasi kepada tim proyek.

Langkah kedua, komitmen pengguna. Pemilihan sejumlah pengguna yang akan menggunakan langsung produk ini yaitu 1 orang.

Langkah ketiga, mendefinisikan tugas. Semua pengguna baik pemilik dan pengguna produk yang terlibat dalam proyek didefinisikan tugas dan tanggung jawabnya.

b. *Scope definition*

Tahap ini dibagi menjadi dua bagian, yaitu membuat dokumen kebutuhan dalam bentuk product backlog dan membuat rencana proyek dalam bentuk sprint planning.

Pengembang bertemu dengan pengguna dan mengumpulkan kebutuhan. Kebutuhan-kebutuhan dimasukkan ke dalam product backlog.

Product backlog yang sudah dibuat kemudian diberikan waktu pengerjaan dan nama pengembang yang membuat.

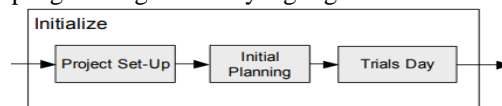
c. *Project establishment*

Tahap ini adalah tahap desain arsitektur perangkat lunak dan templat perangkat lunak. ERD, Flowchart dan MockUp dibuat untuk mendesain perangkat lunak.

- 1) Software menggunakan ADT, meliputi android-sdk dan eclipse-SDK-3.62win32.
- 2) Erd memiliki tiga table tbl_stn, tbl_trx dan tbl_usr.

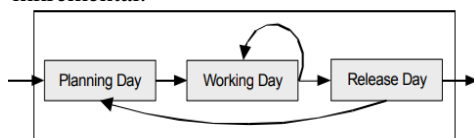
2. *Initialize*

Tahap ini bertujuan untuk menyiapkan proyek yang akan dilaksanakan, seperti penyiapan lingkungan sistem, pengguna, pengembang dan alat yang digunakan.



Gambar 4: Fase Explore

- a. **Project setup**
Melakukan penyiapan lingkungan perangkat lunak seperti sistem operasi, alat, teknologi, perangkat keras dan perangkat lunak untuk proses pengembangan. Training juga dilakukan kepada tim untuk menjelaskan secara spesifik metode yang digunakan, alat dan hal teknis lainnya. Dokumen untuk merekam komunikasi antara pengguna dengan pengembang dibuat agar memudahkan pengembang dalam mengecek kebutuhan.
 - b. **Planning day in 0 iteration**
Melakukan rencana skenario pengujian, pada mobile-d, proses uji dilakukan dengan TDD (Test Driven Development) dengan mendefinisikan terlebih dahulu skenario pengujian
 - c. **Working day in 0 iteration**
Melakukan evaluasi terhadap rencana proyek yang sudah dibuat sebelumnya.
 - d. **Release day in 0 iteration**
Dalam mobile-d, proses ini bersifat opsional dan tidak ada proses rilis pada tahap ini.
3. **Productionize**
Tahap ini adalah untuk mengimplementasikan kebutuhan ke dalam produk dengan pengembangan secara inkremental.



Gambar 5: Fase Productionize

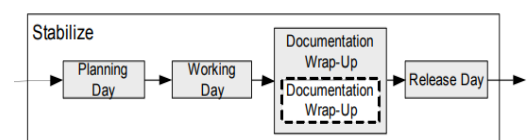
- a. **Planning day**
Melakukan konfirmasi kepada pengguna bahwa kebutuhan di dalam product backlog sudah sesuai. Membuat skenario pengujian berdasarkan product backlog, membuat action point, task card dan melakukan evaluasi terhadap skenario pengujian.
- b. **Working day**
Pagi hari sebelum pengembangan dimulai dilakukan proses wrap-up, yaitu proses diskusi dan pemilihan pekerjaan. Setelah wrap-up, pengembang membuat kode program untuk TDD. Setelah TDD dibuat, pengembang melakukan proses pengembangan dengan metode pair-programming. Pair programming melibatkan sedikitnya dua orang, yaitu pengembang yang membuat kode dan supervisi pengembang yang mengarahkan dan mengontrol kode program.

- Perangkat lunak kontrol versi (SVN) diinstal di server untuk melakukan rilis dan mengintegrasikan modul-modul. Refactoring diperlukan jika ada kode sumber yang perlu dimodifikasi dengan tujuan agar aplikasi lebih rapi, cepat namun tidak mengubah esensi dasar. Setelah proses-proses tadi sudah dilakukan, pengembang menginfokan dan mengirim progres kepada pengguna.
- c. **Release day**
Setelah pengembang membangun modul-modul di dalam aplikasi, modul yang terpisah diintegrasikan sehingga menjadi satu kesatuan. Integrasi ini juga menggunakan SVN untuk melakukan rilis modul. Modul-modul yang sudah dibuat, diuji oleh pengujian untuk memastikan masih atau tidak ada cacat. Jika dalam proses uji ada modul yg cacat, dicatat dalam dokumen defect list. Setelah selesai uji, tim pengembang melakukan cek rilis dengan mengisi dokumen cek rilis.

Selama aplikasi yang dibangun belum sesuai dan belum terintegrasi, proses akan melakukan iterasi ke planning day. Setelah semua modul sudah terintegrasi, langkah berikutnya adalah stabilize.

4. Stabilize

Tujuan tahap ini adalah untuk memastikan kualitas implementasi proyek setelah dilakukan pengembangan pada tahap Productionize.



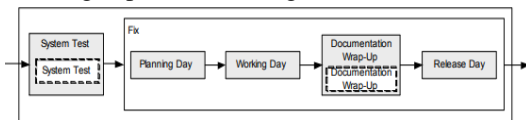
Gambar 6: Fase Stabilize

- a. **Planning day**
Tahap ini adalah mendefinisikan task card untuk menentukan daftar kerja yang masih tersisa (jika ada) dan untuk meningkatkan kualitas produk secara eksternal dan internal.
- b. **Working day**
Finalisasi implementasi dari produk yang dibangun sesuai dengan task card dan memastikan kualitas dari produk dengan berkomunikasi dengan pengguna. Jika masih ada yang belum sesuai dengan kertas kerja, tim pengembang menyelesaikan sepenuhnya pada tahap ini serta didampingi oleh pengguna.
- c. **Documentation wrap-up**
Finalisasi arsitektur aplikasi, desain dan tampilan layar. Dalam hal ini tim pengembang berkomunikasi dengan pengguna.

- d. *Release day*
 Tahap ini untuk memverifikasi dan validasi keseluruhan implementasi, kualitas aplikasi beserta dokumentasinya. Memastikan bahwa aplikasi sudah final dan aplikasi siap diuji melalui tahap System Test and Fix.

5. *System Test and Fix*

Tahap ini tim pengembang melakukan pengujian terhadap sistem dan aplikasi. Tim pengembang dan pengguna bersama-sama memastikan tidak ada cacat dan sesuai dengan product backlog.



Gambar 7 : Fase Test and Fix

- a. *System test*
 Tahap ini untuk menemukan cacat dalam aplikasi pasca pengembangan dengan membuat system test plan, test report dan test log.
- b. *Planning day*
 Melakukan perencanaan uji dengan membuat system test plan. System test plan adalah dokumen yang berisi rencana uji terhadap aplikasi.
- c. *Working day*
 Melakukan uji aplikasi berdasarkan dokumen system test plan. Hasil uji dimasukkan ke dokumen system test log, kemudian dipindahkan dalam bentuk rangkuman ke dalam dokumen system test report. Modul yang masih ditemui cacat, diperbaiki oleh tim pengembang sampai aplikasi tidak ada cacat.
- d. *Release day*
 Rilis hasil uji setelah proses uji sudah selesai, sesuai dengan dokumen tampilan layar, sesuai product backlog dan tidak ada cacat. Aplikasi yang sudah final, dapat dirilis dan dipublikasi ke pengguna.

B. Teknik Pengumpulan Data

1. Observasi yaitu metode pengumpulan data melalui pengamatan secara langsung terhadap kebutuhan cleaning complaint report saat melakukan perjalanan menggunakan pesawat terbang.
2. Metode Studi Pustaka adalah penulis melakukan pengumpulan data dengan cara membaca buku-buku, journal, untuk mencari data-data dari sumber yang berhubungan dengan penelitian. (Yulianto & Wahdini, 2017)

HASIL DAN PEMBAHASAN

Implementasi aplikasi yang akan dibangun adalah aplikasi komplain kebersihan maskapai yang diberi nama Cleaning Complaint Report selanjutnya disingkat aplikasi CCR yang berjalan di perangkat ponsel. Aplikasi ini memungkinkan pengguna untuk memasukkan komplain terkait dengan kebersihan di dalam pesawat terbang. Pengguna hanya cukup unduh aplikasi kemudian mengisi data tanggal, no penerbangan, komplain dan gambar hasil foto. Data yang dimasukkan oleh pengguna akan tampil di aplikasi khusus untuk administrator aplikasi CCR.

1. Explore

Tim pengembang membuat dokumen kebutuhan pengguna dalam bentuk Product Backlog dengan cara melakukan observasi dan wawancara.

Tabel 1. Product Backlog

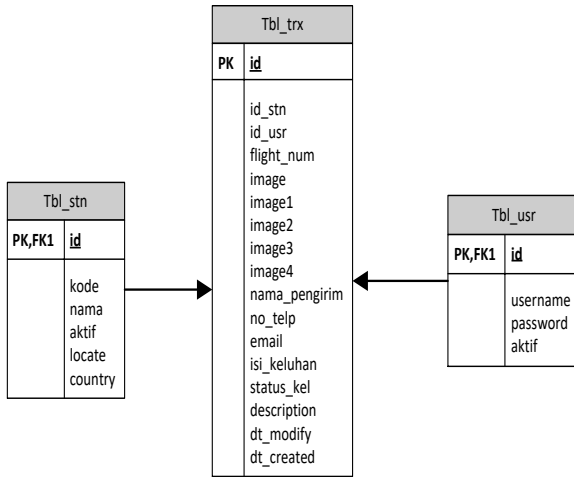
Kebutuhan	Prioritas
Aplikasi yang dapat diakses secara mobile oleh pengguna	High
Diperlukan aplikasi yang dapat merekam dengan foto kemudian diupload melalui aplikasi	High
Aplikasi admin yang dapat melihat laporan dari pengguna	Medium

Dari kebutuhan yang sudah dibuat dalam product backlog, maka dibuat sprint backlog untuk menentukan modul yang akan dibuat, waktu, nama pengembang dan hasil capaian.

Tabel 2. Sprint Backlog

PBI's	Estimasi	Durasi	PIC	Capaian
Membuat Database Relationship	1	1	Feri	Done
Membuat Mockup	1	1	Feri	Done
Membuat Flowchart sistem	1	1	Feri	Done
Sales Order	5	6	Andi	Done
Testing SO Shipment	1	1	Feri	Done
Testing Shipment	1	1	Feri	Done
Invoice	5	2	Soni	Done
Testing Invoice	1	1	Feri	Done
Payment	5	4	Soni	Done
Testing Payment	1	1	Feri	Done
Invoice	5	2	Soni	Done
Testing payment	1	1	Feri	Done

Setelah product backlog dibuat, tim pengembang membuat desain arsitektur aplikasi seperti ERD dan tampilan layar



Gambar 8. Entity Retalionship Diagram

Gambar 9. Form Login

Gambar 10. Layar Home

Gambar 11. Form Cleaning Report

NO	REPORT NO	DATE	STATION	STATUS
XX	XXXXX	XXXXXX	XXXXXX	OPEN DETAIL
XX	XXXXX	XXXXXX	XXXXXX	OPEN DETAIL
XX	XXXXX	XXXXXX	XXXXXX	OPEN DETAIL

Gambar 12. Report Cleaning

2. Initalize

Melakukan penyiapan server dan database agar aplikasi bisa diakses dari internet. Sebagai awal, tim menyiapkan *shared hosting* sebesar 2GB. Alat pendukung dan perangkat lunak yang dibutuhkan sudah terpasang di komputer pengembang, jadi tidak perlu dilakukan instalasi lagi.

3. Productionize

Tim membuat action point list dan task card. Action point list berfungsi sebagai dokumentasi untuk merekam semua permasalahan pada fase ini, sedangkan task card berfungsi sebagai dokumentasi untuk merekam pekerjaan yang sudah dilakukan oleh tim pengembang.

Tabel 3. Dokumen Action Point List

Topik : Test Driven Development				
Finding	Action Point	Actor	Follo w up plan	Realization
Tim masih minim pengetahuan mengenai TDD	Tim yg pengala man akan member ikan pelatiha n	Project Manager	Akan berdis kusi menge nai waktu trainin g	Sudah dilakukan training
Tim	PM	Project	PM	Akan

pengembang memiliki kompetensi yang berbeda	akan melakukan standarisasi kode dan framework yang digunakan	manajer	akan menyelesaikan waktu khusus untuk meeting dengan tim	dilakukan meeting
---	---	---------	--	-------------------

Tabel 4. Dokumen Task Card

ID	Task	Estimasi	SDM	Prioritas
1	membuat template	2 hari	Alfias	High
2	membuat database	1 hari	Alfias	Hight
2	membuat form input	7 hari	Alfias	High
3	membuat laporan	2 hari	Alfias	Medium

Task yang sudah dirilis dicatat dalam dokumentasi cek rilis seperti di bawah ini :

Tabel 5
 Dokumen Cek Rilis

ID Task	Yes	No	N/A
1	Yes		
2	Yes		
3	Yes		

4. Stabilize

Tahap ini mengevaluasi apakah masih ada *task* dari *task card* yang belum selesai. Jika masih ada *task* yang belum selesai, maka tim pengembang wajib menyelesaikannya. Tim pengembang juga harus berkomunikasi dengan pengguna mengenai *task* yang sudah diselesaikan oleh pengembang, apakah sudah sesuai atau belum sesuai.

5. System Test and Fix

Tester membuat dokumen dokumen pengujian dengan *test plan* dan *test log*. Dokumen *test plan* berfungsi untuk mendefinisikan rencana pengujian, sedangkan *test log* adalah serangkaian pengujian

yang sudah dilakukan oleh *tester*. *Tester* melakukan pengujian berdasarkan *test plan* yang dibuat, apabila ditemui kesalahan kode atau fungsi, *tester* mencatat dalam *test log* dan menginformasikan kepada pengembang untuk segera diperbaiki sampai tidak ada lagi kesalahan aplikasi sampai dirilis.

KESIMPULAN

Metode Mobile-D cocok digunakan untuk pengembangan aplikasi mobile dengan metode pair-programming. Mobile-D mengatur kelengkapan dokumentasi dan metode pengembangan tidak seperti Scrum yang minim dokumentasi. Metode ini bisa diadaptasi ke dalam banyak platform aplikasi mobile seperti mobile web, aplikasi natif, phone gap dan titanium.

REFERENSI

- Abrahamsson, P., Hanhineva, A., Hulkko, H., Ihme, T., Jääliñoja, J., Korkala, M., ... Salo, O. (2004). Mobile-D: An Agile Approach for Mobile Application Development. *International Journal of Service Industry Management*, 174–175. <https://doi.org/10.1145/1028664.1028736>
- Flora, H. K., & Chande, S. V. (2013). a Review and Anaysis on Mobile Application Development Processes Using Agile, 3(4), 9–18.
- Goggin, G. (2012). Google phone rising: The Android and the politics of open source. *Continuum*, 26(5), 741–752. <https://doi.org/10.1080/10304312.2012.706462>
- Liviu Despa, M. (2014). Comparative Study on Software Development Methodologies. *Database Systems Journal*, 5(3), 37–56. <https://doi.org/10.1109/MAHC.1983.10102>
- Wescon, I. (1970). MANAGING THE DEVELOPMENT OF LARGE SOFTWARE SYSTEMS Dr. Winston W. Rovce INTRODUCTION, (August), 1–9.
- Yulianto, A., & Wahdini, A. (2017). *Laporan Akhir Penelitian Mandiri*. Jakarta.