

Perancangan Aplikasi Asistensi Skripsi Mahasiswa berbasis *Progressive Web App* dengan Metode *Prototyping*

Welly Dian Astika¹, Adam Hendra Brata², Komang Candra Brata³

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya
Email: ¹wellydianastika@gmail.com, ²adam@ub.ac.id, ³k.candra.brata@ub.ac.id

Abstrak

Penyelesaian skripsi bagi mahasiswa merupakan salah satu syarat untuk mendapatkan gelar sarjana. Namun masih ada mahasiswa yang memiliki beberapa masalah dalam proses pengerjaan skripsi. Oleh karena itu, diperlukan wadah alternatif untuk menampung kebutuhan tambahan mahasiswa dalam satu lingkungan dan mampu membantu mempermudah mahasiswa dalam menyelesaikan aktivitas skripsinya yaitu aplikasi asistensi skripsi mahasiswa. Sistem ini dirancang menggunakan teknologi *Progressive Web App* (PWA) sehingga mahasiswa dan wali-nya dapat menggunakan aplikasi *web* ini secara *offline*, mendapatkan notifikasi informasi terbaru dan menambahkannya ke dalam *home screen*. Perancangan yang dilakukan adalah perancangan arsitektur, komponen, data, dan antarmuka. Perancangan aplikasi ini diuji menggunakan *Traceability Matrix*, (*Coupling between Object Classes*) CBO, (*Response for a Class*) RFC, (*Lack of cohesion in methods1*) LCOM1 metrics, dan (*Lack of cohesion in methods2*) LCOM2 metrics. Hasil perancangan memiliki kesesuaian pada semua artefak yang dibuat berdasarkan *use case* dengan pengujian *traceability matrix*. Hasil pengujian kualitas rancangan memiliki kopling rendah dengan rata-rata nilai berada pada tingkat $CBO \leq 5$ dan $RFC \leq 100$. Kohesi rendah pada ketiga *class* yang diuji dengan $LCOM1 > 1$ dan memenuhi nilai LCOM2 0 hingga 1, *understandability* tinggi karena kopling rendah, dan *adaptability* tinggi karena rata-rata nilai CBO diantara 1 hingga 3. Rata-rata nilai RFC diantara 1 hingga 69.

Kata kunci: *asistensi, skripsi, progressive web app, prototyping*

Abstract

Completion of thesis for students is one of the requirements to get a bachelor's degree. But there are still students who have some problems in the thesis process. Therefore, an alternative container is needed to accommodate the additional needs of students in one environment and be able to help facilitate students in completing their thesis activities, namely the application of student thesis assistance. This system is designed using Progressive Web App (PWA) technology so that students and their guardians can use this web application offline, get notification of the latest information and add it to the home screen. The design carried out is the design of architecture, components, data, and interfaces. The design of this application was tested using the Traceability Matrix, (Coupling between Object Classes) CBO, (Response for a Class) RFC, (Lack of cohesion in methods 1) LCOM1 metrics, and (Lack of cohesion in methods 2) LCOM2 metrics. The results of the design are compatible with all artifacts made based on the use case with traceability matrix testing. The design quality test results have a low coupling with an average value at the level of $CBO \leq 5$ and $RFC \leq 100$. Low cohesion in all three classes tested with $LCOM1 > 1$ and fulfills LCOM2 values 0 to 1, high understandability due to low coupling, and high adaptability due to average CBO values between 1 to 3. Average RFC values between 1 to 69.

Keywords: *assistance, minithesis, progressive web app, prototyping*

1. PENDAHULUAN

Penyelesaian skripsi bagi mahasiswa merupakan sebagian persyaratan untuk

memperoleh gelar sarjana di Fakultas Ilmu Komputer (FILKOM) Universitas Brawijaya. Dalam menyelesaikan skripsi, mahasiswa harus melewati berbagai tahapan aktivitas mulai dari

perancangan pra proposal skripsi hingga pelaksanaan ujian skripsi. Berdasarkan hasil observasi peneliti, aktivitas yang berkaitan secara langsung dalam pengerjaan skripsi yaitu proses bimbingan dengan dosen pembimbing sebagai syarat melakukan P1, P2 dan pendaftaran semhas dan ujian, proses pengurusan berkas-berkas selain skripsi yang dibutuhkan selama proses penyelesaian skripsi, serta penggunaan FILKOM APPS untuk mengetahui dan menyelesaikan tahapan aktivitas skripsi.

Berdasarkan hasil wawancara yang telah dilakukan dengan Prasetyo Iskandar selaku staff kemahasiswaan FILKOM UB, beliau menerangkan bahwa salah satu penyebab mahasiswa terlambat maupun gagal dalam menyelesaikan skripsinya dikarenakan kurang menguasai ilmu dan wawasan seputar topik skripsi yang diambil. Untuk memastikan kebenaran informasi dari narasumber, peneliti melakukan survei dengan teknik *purposive sampling* agar pemilihan anggota populasi sesuai dengan syarat profil responden yang dibutuhkan. Berdasarkan hasil survei yang telah peneliti lakukan, pada saat mahasiswa menempuh proses skripsi ditemukan beberapa masalah antara lain seputar ide judul skripsi, proses bimbingan dan komitmen mengerjakan. Permasalahan tersebut dapat menghambat penyelesaian skripsi yang menyebabkan proses skripsi pada semester tersebut harus digagalkan karena belum mencapai tahapan yang ditentukan.

Dengan adanya berbagai permasalahan diatas, diperlukan adanya teknologi komputer yang mampu membantu menyelesaikan permasalahan pada proses skripsi mahasiswa yang telah didefinisikan sebelumnya. Dengan adanya teknologi komputer yang berupa aplikasi, sistem dapat membantu memudahkan mahasiswa dalam mencari topik skripsi dalam kumpulan kata kunci (*keyword*). Proses bimbingan dapat dilakukan lebih teratur dengan adanya sistem slot bimbingan. Membantu mahasiswa beserta wali mahasiswa untuk bisa mendapatkan informasi mengenai semua aktivitas mahasiswanya selama skripsi dengan notifikasi oleh sistem pengingat (*reminder*). Maka dari itu, pengerjaan skripsi mahasiswa FILKOM UB membutuhkan wadah alternatif yang dapat menampung kebutuhan tambahan mahasiswa dalam satu lingkungan, yaitu *Progressive Web App* (PWA).

PWA adalah aplikasi *web* tradisional yang disempurnakan dengan teknologi *web* modern,

memungkinkan pengembang untuk memberikan pengalaman yang mirip dengan aplikasi perangkat bergerak (*mobile*) (Developers, 2017). Teknologi PWA ini cocok diterapkan pada aplikasi asistensi skripsi mahasiswa, mahasiswa dan wali mahasiswa dapat menggunakan aplikasi *web* ini secara *offline* dan mendapatkan notifikasi informasi terbaru hanya dengan mengakses halaman aplikasi *web* dan menambahkannya ke dalam *home screen*, sehingga jika diterapkan pada aplikasi asistensi skripsi mahasiswa pengguna hanya perlu menggunakan *smartphone* jenis apa saja dengan *browser* yang telah support teknologi PWA.

Maka dari itu, peneliti membuat rancangan aplikasi asistensi skripsi mahasiswa dengan menerapkan teknologi *web* modern *Progressive Web App* agar dapat membantu mempermudah mahasiswa dalam menyelesaikan aktivitas skripsi. Hasil rancangan diuji dengan *traceability matrix*, CBO (*Coupling between Object Classes*), RFC (*Response for a Class*), LCOM1 (*Lack of cohesion in methods 1*), dan LCOM2 (*Lack of cohesion in methods 2*). Pengujian ini dilakukan agar rancangan yang telah dibuat agar dapat diketahui kelengkapan artefaknya, tingkat *adaptability* dan *understandability*-nya (Hartasih, Rusdianto, & Brata, 2019). Rancangan aplikasi PWA pada asistensi skripsi mahasiswa ini dibuat oleh peneliti untuk mendapatkan hasil penyelesaian masalah secara lengkap beserta bentuk akhir rancangan aplikasi sebagai bahan untuk pembangunan aplikasi PWA asistensi skripsi mahasiswa oleh peneliti secara baik dan benar.

2. LANDASAN KEPUSTAKAAN

2.1 Kajian Pustaka

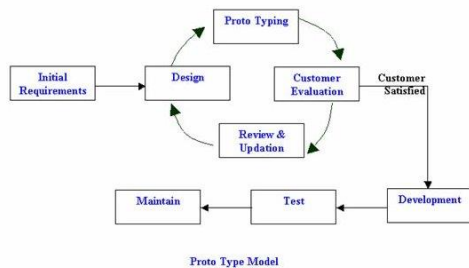
Kajian pustaka yang dibahas dalam penelitian ini adalah penelitian yang pernah dilakukan sebelumnya mengenai penerapan *web* menggunakan pendekatan *Progressive Web App* dan perancangan perangkat lunak beserta pengujiannya.

Platform *e-Learning* untuk pembelajaran pemrograman *web* menggunakan konsep *Progressive Web Apps* ini merupakan platform *e-Learning* dengan penerapan konsep *Progressive Web Apps* mampu menampilkan halaman secara *offline* tetapi tidak bisa menyimpan, mengubah, atau menghapus data pada basis data. Peneliti mengambil konsep PWA untuk menampilkan halaman secara *offline*

(Adi, Akbar, & Khotimah, 2017).

Extension Features Driven Use case Model for requirement traceability (e-FDUCM) mempunyai kemampuan untuk mendeskripsikan dan melacak kebutuhan dengan maju dan mundur. Peneliti mengambil konsep untuk malacak artefak ke kebutuhan, dan sebaliknya menggunakan *traceability matrix – use case* ke artefak dari realisasi *use case* (Kong & Yuan, 2009).

2.2 Software Prototype Model

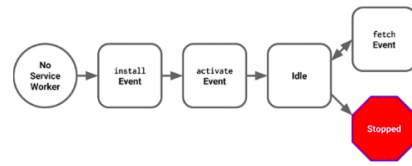


Gambar 1. *Software Prototype Model*

Prototype adalah versi awal dari sistem perangkat lunak yang digunakan untuk menunjukkan konsep, coba opsi desain, dan cari tahu lebih lanjut tentang masalah dan kemungkinannya solusi. Perkembangan iteratif yang cepat dari *prototype* sangat penting sehingga biaya dikendalikan dan pemangku kepentingan sistem dapat bereksperimen dengan *prototype* di awal proses perangkat lunak (Sommerville, 2011). Berdasarkan 6 tahapan dasar pada *software prototype model* dari Gambar 1 tersebut disesuaikan kembali dengan kebutuhan peneliti dalam menyelesaikan penelitian ini.

2.3 Progressive Web App

Progressive Web App (PWA) adalah konsep yang menggabungkan pengalaman terbaik dari *web* dan terbaik dari aplikasi. PWA sudah berfungsi bagi pengguna sejak kunjungan pertama menggunakan browser dan tidak diperlukan pemasangan. Saat pengguna semakin sering menggunakan aplikasi *web* dari waktu ke waktu membuat *loading* aplikasi semakin. Dapat memuat halaman *web* lebih cepat, bahkan di jaringan yang kurang stabil, mengirim push *notification* yang relevan, memiliki ikon sendiri di homescreen, dan mampu memberikan pengalaman full screen (LePage, 2018).

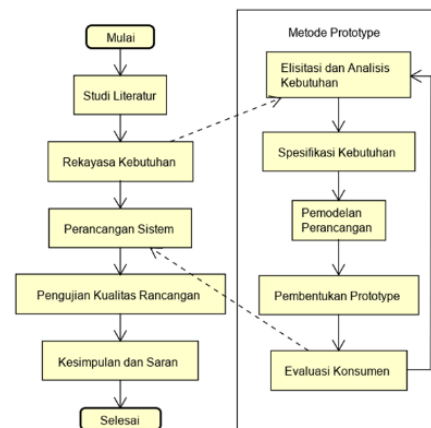


Gambar 2. *Key Concepts* dari PWA

Pada Gambar 2 tersebut merupakan alur konsep dari proses update setiap *event* yang terjadi selama *web app* dijalankan.

3. METODOLOGI PENELITIAN

Penelitian ini merupakan penelitian implementatif perancangan dengan menggunakan metode model purwarupa sebagai metode penyelesaian. Alur metodologi yang digunakan tidak sepenuhnya mengadopsi standar model dari buku *sommerville*, namun terdapat beberapa hal yang disesuaikan oleh peneliti agar memudahkan dalam proses pengerjaan penelitian. Pada tahap *initial requirements*, peneliti membaginya menjadi dua tahap yaitu tahap studi literatur dan sebagian tahap rekayasa kebutuhan mencakup elisitasi dan analisis kebutuhan dan spesifikasi kebutuhan. Pada tahap *design* dan tahap *prototyping*, diadopsi peneliti menjadi tahap pemodelan perancangan dan tahap pembentukan *prototype*. Pada tahap *customer evaluation* dan tahap *review and updation*, peneliti lebur jadi menjadi satu ke tahap evaluasi konsumen. Peneliti menghapus tahap *development* dan tahap *maintain* karena penelitian ini merupakan tipe perancangan dan merubah tahap *test* menjadi tahap pengujian kualitas rancangan. Satu tahap akhir yaitu kesimpulan dan saran sebagai penutup dari penelitian ini. Diagram alir runtutan pengerjaan penelitian direpresentasikan pada Gambar 3 berikut.



Gambar 3. Diagram Alir Metode Penelitian

Kegiatan studi literatur merupakan analisis dari rincian kajian pustaka dan dasar teori yang digunakan dalam penelitian ini yang didapat dari penelitian yang pernah dilakukan sebelumnya, berbagai sumber literatur *hardcopy*, conference proceedings dan beberapa literatur dari internet.

Pada tahap rekayasa kebutuhan atau analisis kebutuhan bertujuan untuk mendapatkan semua kebutuhan yang diperkukan dari sistem yang dibangun. Analisis kebutuhan dilakukan dengan mengidentifikasi semua kebutuhan (requirement) sistem dan siapa saja yang terlibat di dalamnya. Analisis juga dilakukan untuk mengetahui kondisi yang ada dilapangan sehingga dapat diketahui implementasi perangkat lunak yang akan digunakan (Adi, Akbar, & Khotimah, 2017).

Di dalam tahap rekayasa kebutuhan dan tahap perancangan, dilakukan metode *prototyping* yang proses iterasinya mencakup elisitasi dan analisis kebutuhan, spesifikasi kebutuhan, pemodelan perancangan, pembentukan *prototype*, dan evaluasi konsumen.

Pada tahap perancangan sistem dilakukan desain arsitektur aplikasi untuk menjelaskan desain arsitektur sistem, perancangan arsitektur menggunakan diagram UML untuk membantu membangun rancangan perangkat lunak sehingga memahami spesifikasi sistem dan rancangan sistem, perancangan komponen, perancangan data untuk pengelolaan data, dan perancangan antarmuka untuk menggambarkan interaksi antara aktor dan sistem (Pressman & Maxim, 2014)

Pada tahap pengujian kualitas rancangan dilakukan pengujian dasar dan pengujian kualitas. Pengujian dasar menggunakan *traceability matrix* untuk mengetahui apakah hasil perancangan merupakan realisasi dari *use case* dan kebutuhan yang telah dibuat sebelumnya. Pengujian kualitas untuk tingkat adaptasi perubahan (*adaptability*), tingkat kemudahan pemahaman (*understandability*), ketergantungan antar modul (*cohesion*), hubungan antar fungsi dalam satu modul (*coupling*) tinggi atau rendah dari aplikasi yang dirancang.

Di tahap penarikan kesimpulan dan saran berisi kesimpulan akhir yang menjawab rumusan masalah dan berisi saran untuk perancangan selanjutnya.

4. REKAYASA KEBUTUHAN

4.1 Gambaran Umum Sistem

Sistem asistensi skripsi mahasiswa yang dirancang pada penelitian ini merupakan sistem alat bantu alternatif bagi khususnya mahasiswa FILKOM UB dalam masa menempuh pengerjaan skripsi. Sistem ini dirancang untuk menjadi sebuah *web apps* yang menerapkan teknologi *progressive web app*. Dengan adanya teknologi PWA ini, aplikasi mampu disematkan dalam *homescreen smartphone* android, dapat diakses dalam keadaan jaringan internet lambat hingga *offline* dan *push notification* yang mampu meningkatkan *user engagement* pengguna secara *real time*.

4.2 Elisitasi dan Analisis Kebutuhan Iterasi Pertama

Elisitasi merupakan tahapan penggalian kebutuhan yang bertujuan membantu pengembang dalam memahami sistem yang dispesifikasi. Elisitasi ini dilakukan dengan metode wawancara terhadap Pak Prasetyo Iskandar dan survey kepada mahasiswa. Dari wawancara tersebut disimpulkan beberapa poin permasalahan yang dianggap benar adanya oleh Pak Prasetyo dengan penilaian minimal setuju berdasarkan standar skala likert, yaitu:

1. Mahasiswa kesulitan mencari ide untuk topik skripsi.
2. Mahasiswa kurang persiapan ilmu dan wawasan topik skripsi yang diambil.
3. Mahasiswa malas atau sering bermain.
4. Mahasiswa sibuk bekerja.
5. Mahasiswa malu atau takut dalam berkonsultasi apabila mendapat kendala dalam pengerjaan skripsi.
6. Mahasiswa belum melakukan pkl.
7. Mahasiswa lupa dengan timeline skripsi.

Dari poin permasalahan diatas, peneliti menawarkan solusi kepada narasumber dengan metode penilaian menggunakan standar skala likert. Hasil tawaran tersebut dipaparkan pada Tabel 1.

Tabel 1. Tawaran Solusi Kepada Narasumber

No	Solusi yang ditawarkan	Penilaian Solusi
1	Adanya Informasi terbaru mengenai hal-hal yang bisa dijadikan topik skripsi	Sangat Setuju
2	Adanya Instruksi Cara Pengerjaan Skripsi	Setuju
3	Adanya Instruksi Cara Pengurusan Pelayanan di akademik seputar	Netral

4	skripsi Adanya Peningat untuk menjadwalkan suatu aktivitas	Sangat Setuju
5	Informasi mengenai bidang kemampuan dosen beserta jadwal bimbingannya	Sangat Setuju
6	Aplikasi bisa diakses secara offline atau koneksi tidak stabil	Sangat Setuju
7	Adanya milestones sebagai informasi detil proses skripsi mahasiswa	Sangat Setuju
8	Setiap Kemajuan Pengerjaan Skripsi Mahasiswa bisa terinformasikan langsung ke handphone mahasiswa dan walinya dalam bentuk notifikasi	Sangat Setuju

Penggalian kebutuhan juga dilakukan dengan metode survey pada mahasiswa FILKOM UB. Berdasarkan survey tersebut, peneliti menghitung hasil dengan skala likert dan hasilnya dapat dilihat pada Tabel 2.

Tabel 2. Survey Kebutuhan Mahasiswa Iterasi 1

No.	Nama Kebutuhan	Nilai Akhir
1	Saya ingin ditampilkan informasi terbaru mengenai hal-hal yang bisa dijadikan topik skripsi.	92%
2	Saya ingin ditampilkan informasi dosen berupa biodata dan kontak yang dapat dihubungi beserta kemampuan nya dalam topik apa saja.	92%
3	Saya ingin disediakan fitur pengingat untuk menjadwalkan suatu aktivitas.	92%
4	Saya ingin disediakan fitur intruksi peminjaman alat untuk kebutuhan skripsi.	36%
5	Saya ingin aplikasi dapat diakses secara offline maupun kondisi jaringan yang lemah.	84%
6	Saya ingin disediakan fitur intruksi untuk pengurusan permasalahan akademik.	96%
7	Saya ingin disediakan fitur intruksi pengurusan berkas akademik dan kemahasiswaan yang berhubungan dengan skripsi.	96%
8	Saya ingin disediakan fitur intruksi cara pengerjaan skripsi.	96%
9	Saya ingin disediakan fitur untuk memberikan informasi secara real time kepada perangkat bergerak milik mahasiswa dan wali mahasiswa.	100%

4.3 Identifikasi Aktor Iterasi Pertama

Pemangku kepentingan atau peran dari sebuah sistem dapat dilihat dari masalah yang terkait. Berdasarkan pengamatan peneliti dan hasil elisitasi yang telah dilakukan, maka

pendefinisian aktor beserta deskripsi umum dapat dilihat pada Tabel 3.

Tabel 3. Identifikasi Aktor Iterasi 1

Aktor	Deskripsi
<i>Pengguna</i>	Aktor yang belum masuk ke dalam sistem dan memiliki hak akses <i>sign in</i> dan <i>sign up</i> .
<i>Mahasiswa</i>	Aktor yang terdaftar di dalam sistem dan memiliki hak akses untuk fungsi milestone skripsi dan <i>reminder</i> secara personal.
<i>Wali Mahasiswa</i>	Aktor yang terdaftar di dalam sistem dan memiliki hak akses untuk fungsi milestone skripsi dan <i>reminder</i> sesuai dengan mahasiswa tanggung jawabnya.
<i>Dosen</i>	Aktor yang terdaftar di dalam sistem dan memiliki hak akses untuk fungsi milestone skripsi.
<i>Adminisrator</i>	Aktor yang terdaftar di dalam sistem dan memiliki hak akses untuk melakukan manipulasi data wali mahasiswa dan <i>keyword</i> .

4.4 Kebutuhan Fungsional Iterasi Pertama

Aplikasi ini harus memenuhi beberapa kebutuhan yang disebut dengan kebutuhan fungsional. Untuk mendapatkan kebutuhan fungsional yang sesuai dengan yang dibutuhkan oleh *stakeholder*, dilakukan wawancara dan survey yang kemudian dilakukan elisitasi. Analisis kebutuhan fungsional sistem sesuai dari hasil elisitasi yang telah dilakukan menghasilkan 46 kebutuhan fungsional sistem pada iterasi pertama.

4.5 Kebutuhan Non-Fungsional Iterasi Pertama

Untuk menunjang pengalaman pengguna terdapat kebutuhan lain yang disebut dengan kebutuhan non-fungsional. Terdapat satu kebutuhan non-fungsional yaitu *usability* yang menilai apakah rancangan antarmuka yang dihasilkan mudah dimengerti oleh pengguna.

4.6 Use Case Diagram Iterasi Pertama

Diagram *use case* merupakan pemodelan hasil representasi kegiatan atau aktivitas sistem yang bisa dilakukan dari sisi aktor. Diagram *use case* yang terbentuk pada iterasi pertama berjumlah 46 *use case* yaitu *use case* Tambah ke Home Screen, Izinkan Notifikasi, Sign In, Sign Up, Lupa Password, Lihat Profil, Ganti Password, Lihat Daftar Skripsi Mahasiswa, Lihat Detail Skripsi, Lihat Daftar Tawaran Judul

Skripsi, Tambah Tawaran Judul Skripsi, Ambil Tawaran Judul Skripsi, Lihat Daftar Rancangan Proposal, Lihat Detail Rancangan Proposal, Edit Rancangan Proposal, Tambah Rancangan Proposal, Hapus Rancangan, Lihat Milestones Skripsi, Pengajuan Proposal Skripsi, Status Pengerjaan P0, Pengajuan P0, Status Pengerjaan P1, Pengajuan P1, Status Pengerjaan P2, Pengajuan P2, Status Pendaftaran Seminar Hasil dan Ujian, Status Persiapan Seminar Hasil, Status Persiapan Ujian, Pembatalan Skripsi, Tambah Rangkaian Milestone, Lihat Daftar Rangkaian Milestone, Edit Rangkaian Milestone, Hapus Rangkaian Milestone, Tambah Reminder, Lihat Reminder, Edit Reminder, Hapus Reminder, Verifikasi pendaftaran Wali Mahasiswa, Lihat Data Wali Mahasiswa, Edit Data Wali Mahasiswa, Hapus Wali Mahasiswa, Lihat Daftar Keywords, Tambah Keywords, Edit Keywords, Hapus Keywords, dan Sign Out. Seluruh *use case* tersebut telah sesuai berdasarkan hasil analisis kebutuhan iterasi pertama.

4.7 Elisitasi dan Analisis Kebutuhan Iterasi Kedua

Penggalian kebutuhan iterasi kedua dilakukan dengan metode survey pada mahasiswa FILKOM UB dengan poin bahasan mengenai penambahan sistem bimbingan antara mahasiswa dengan dosen. Berdasarkan survey tersebut, peneliti menghitung hasil dengan skala likert dan hasilnya dapat dilihat pada Tabel 4.

Tabel 4. Survey Kebutuhan Mahasiswa Iterasi 2

No.	Nama Kebutuhan	Nilai Akhir
1	Saya ingin ditampilkan informasi mengenai jadwal bimbingan dosen.	92%
2	Saya ingin disediakan fitur untuk memudahkan saya untuk melakukan bimbingan kepada dosen.	92%

4.8 Identifikasi Aktor Iterasi Kedua

Berdasarkan hasil elisitasi dari survey yang telah dilakukan sebagai hasil iterasi kedua, terdapat penambahan deskripsi umum pada aktor mahasiswa dan dosen yang dapat dilihat pada Tabel 5.

Tabel 5. Identifikasi Aktor Iterasi 2

Aktor	Deskripsi
Mahasiswa	Aktor yang terdaftar di dalam sistem dan memiliki hak akses tambahan untuk bimbingan.
Dosen	Aktor yang terdaftar di dalam sistem dan

memiliki hak akses tambahan untuk bimbingan.

4.9 Kebutuhan Fungsional Iterasi Kedua

Berdasarkan hasil elisitasi dari survey yang telah dilakukan sebagai hasil iterasi kedua, analisis kebutuhan fungsional sistem mendapat tambahan 4 kebutuhan fungsional baru dengan total menjadi 50 kebutuhan fungsional sistem pada iterasi kedua.

4.10 Use Case Diagram Iterasi Kedua

Pada hasil analisis kebutuhan iterasi kedua. Terdapat 4 *use case* baru yang terbentuk dari kebutuhan fungsional iterasi kedua yaitu *use case* Lihat Jadwal Bimbingan, Pengajuan Bimbingan, Batalkan Bimbingan, dan Edit Jadwal Bimbingan. Dengan tambahan *use case* baru dari iterasi kedua, total jumlah *use case* menjadi 50.

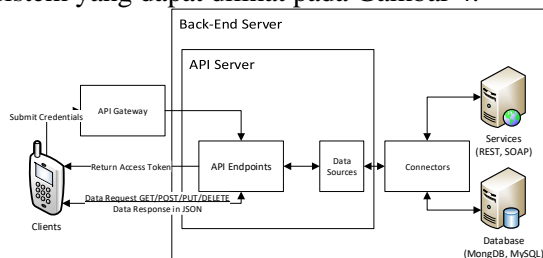
5. PERANCANGAN

Perancangan yang dibuat berdasarkan dari hasil analisis kebutuhan. Pembahasan mengenai perancangan sendiri terdiri dari perancangan arsitektur, perancangan basis data, dan perancangan antarmuka.

5.1 Perancangan Arsitektur Iterasi Pertama

Perancangan arsitektur sistem asistensi skripsi mahasiswa berbasis *progressive web app* ini terbagi menjadi dua bagian yaitu sisi aplikasi dan sisi server. Sisi aplikasi digunakan oleh pengguna untuk berinteraksi dengan sistem sedangkan sisi server peneliti menggunakan API server untuk menangani segala pertukaran data. Aplikasi dirancang untuk diterapkan pada ReactJS dan LoopbackJS yang merupakan *front-end framework* yang sudah menyediakan pengaturan dasar untuk *progressive web app* yang dibangun dan *back-end framework* sebagai API server. Untuk *back-end framework* peneliti memanfaatkan FILKOM API sebagai sumber eksternal data dan membuat API server sendiri untuk menangani penyimpanan dan pertukaran data yang dibutuhkan oleh sistem tapi tidak dimiliki oleh FILKOM API. Aplikasi dirancang untuk menggunakan arsitektur metode komunikasi menggunakan *REST Web Service*. *Back-end server* mencakup keseluruhan API server dan *database* beserta *service* yang digunakan oleh sistem. Pengguna dan server saling berinteraksi dengan perantara *user*

interface dari aplikasi. Rancangan *Back-end server* tidak dibahas secara rinci dalam penelitian ini karena fokus penelitian ini hanya pada rancangan sisi aplikasi *client*-nya saja. Berikut merupakan gambar rancangan arsitektur sistem yang dapat dilihat pada Gambar 4.

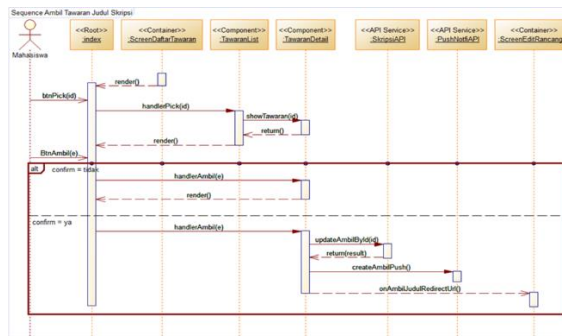


Gambar 4. Arsitektur Asistensi Skripsi Mahasiswa

Berikut merupakan alur komunikasi *client* dengan *server*:

1. *Client* melakukan autentikasi ke *API gateway* dengan mengirimkan data kredensial menggunakan https.
2. *API gateway* menerima data autentikasi dan melakukan pengecekan data kredensial.
3. Setelah lolos autentikasi kemudian proses dilanjutkan *API endpoints* untuk memberi *access token* kepada aplikasi *client*.
4. *Client* melakukan *request GET/POST/PUT/DELETE* yang di tujukan kepada *API endpoint*.
5. *API endpoint* menerima *request* dari *client* kemudian mengakses *repository* dari *data sources* tertentu sesuai dengan data permintaan *client*.
6. *Data sources* melakukan aksi *querying* dengan bantuan *connector* sebagai perantara ke *database*.
7. Data hasil *querying* dikirimkan kepada *API server* untuk diolah menjadi JSON.
8. Melalui *API endpoint* data JSON dikirimkan kepada aplikasi *client*.
9. Respon JSON diterima *client* dan penyesuaian ke format kebutuhan aplikasi.
10. *User interface* menampilkan data yang telah diproses kepada pengguna.

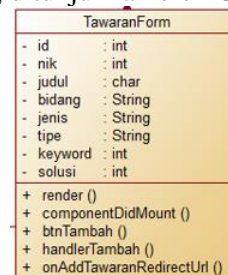
Setiap *sequence* diagram menggambarkan setiap *use case* yang ada yang telah didefinisikan pada tahap rekayasa kebutuhan sebelumnya. Terdapat 46 *sequence* diagram salah satu contohnya yaitu diagram *sequence* ambil tawaran judul skripsi pada Gambar 5.



Gambar 5. *Sequence* Ambil Tawaran Skripsi

Berdasarkan penggunaan arsitektur *client server* dan *component-based engineering*, *class* diagram yang dirancang terdiri dari *class root*, *class service worker*, *class screen*, *class component*, dan *class API service*. *Class screen* merupakan *container* untuk *class component*, kemudian *class component* menggunakan *class API service* untuk mengambil data dan kemudian di *render* di *class screen*.

Berikut adalah salah satu contoh *class diagram* yang berfungsi sebagai komponen form tawaran yang ditunjukkan oleh Gambar 6.



Gambar 6. Diagram *Class Component* TawaranForm

Class screen yang terdapat dalam rancangan yang telah dibuat terdiri dari *ScreenSignIn*, *ScreenSignUp*, *ScreenLupa Pswd*, *ScreenDetailProfil*, *ScreenGantiPswd*, *ScreenDaftarSkripsi*, *ScreenDetailSkripsi*, *ScreenDaftarTawaran*, *ScreenTambahTawaran*, *ScreenDetailTawaran*, *ScreenDaftarRancangan*, *ScreenDetailRancangan*, *ScreenTambah Rancangan*, *ScreenEditRancangan*, *Screen HomeMilestone*, *ScreenTodoList*, *ScreenDaftar RangkaianMilestone*, *ScreenTambahRangkaian Milestone*, *ScreenEditRangkaianMilestone*, *ScreenDaftarReminder*, *ScreenTambah Reminder*, *ScreenEditReminder*, *ScreenDaftar Wali*, *ScreenEditWali*, *ScreenDaftarKeyword*, *ScreenTambahKeyword*, *ScreenEditKeyword*, *ScreenHeader*, dan *ScreenNavbar*.

Class component yang terdapat dalam rancangan yang telah dibuat terdiri dari *SignInForm*, *SignUpForm*, *LupaPswdForm*, *ProfilDetail*, *GantiPswdForm*, *SkripsiList*,

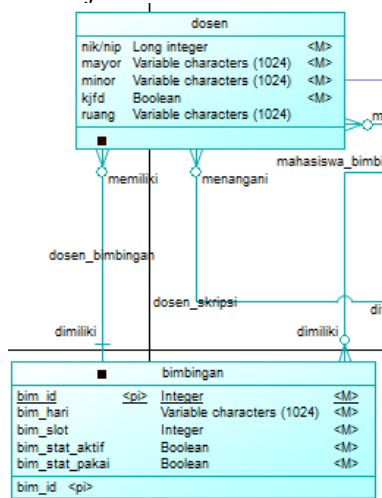
SkripsiDetail, TawaranList, TawaranForm, TawaranDetail, RancanganList, RancanganDetail, RancanganForm, BatalSkripsiForm, MilestoneSlider, TodoList, TodoDetail, MilestoneList, MilestoneForm, RemiderList, ReminderForm, WaliMhsList, WaliMhsForm, KeywordList, dan KeywordForm.

Class API yang terdapat dalam rancangan yang telah dibuat terdiri dari FilkomAPI, SkripsiAPI, MilestoneAPI, ReminderAPI, WaliMhsAPI, KeywordAPI, dan PushNotifAPI.

Untuk *back-end* sistem terdiri dari beberapa jenis komponen seperti *controller*, *repository*, *data source*, *connector*, dan *model* tetapi tidak dijelaskan secara detail dalam penelitian ini.

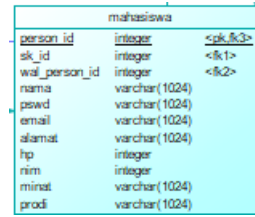
5.2 Perancangan Basis Data Iterasi Pertama

Conceptual Data Model (CDM) digunakan dalam perancangan data untuk menjelaskan relasi antar tabel dalam database. Dalam rancangan CDM, terdapat 10 tabel dengan contoh tabel dosen dan bimbingan beserta relasinya ditunjukkan oleh Gambar 7.



Gambar 7. CDM Tabel Dosen Dan Bimbingan

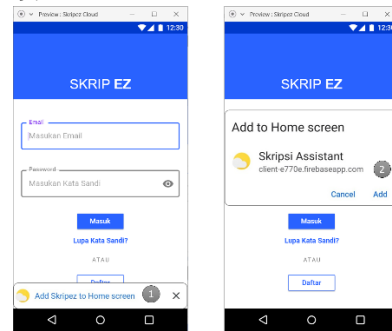
Physical Data Model (PDM) digunakan dalam perancangan data untuk menjelaskan *primary key* dan *foreign key* dalam tabel di database. Dalam rancangan CDM, terdapat 11 tabel dengan contoh tabel dosen dan bimbingan beserta relasinya ditunjukkan oleh Gambar 8.



Gambar 8. PDM Tabel Mahasiswa

5.3 Perancangan Antarmuka Iterasi Pertama

Pada perancangan antarmuka, dibuat desain *mock up* halaman yang dikelompokan berdasarkan *use case* yang diwakiliknya. Salah satu dari antarmuka tersebut ialah Antarmuka Tambah ke HomeScreen yang ditunjukkan pada gambar 9.



Gambar 9. Antarmuka Tambah ke Homescreen

Gambar di atas merupakan merupakan gambar dari halaman untuk menambahkan aplikasi ke *homescreen* smartphone. Langkah yang dilakukan adalah dengan tap pada popup menambahkan ke *homescreen* dan menyetujui untuk menambahkan pada *homescreen*.

5.4 Perancangan Arsitektur Iterasi Kedua

Pada perancangan arsitektur kedua terdapat 4 *use case* tambahan dari hasil elisitasi dan analisis kebutuhan. Untuk itu peneliti merancang 4 diagram *sequence* tambahan dan diagram *class* baru. Diagram *sequence* baru yaitu diagram *sequence* Lihat Jadwal Bimbingan, Pengajuan Bimbingan, Batalkan Bimbingan, dan Edit Jadwal Bimbingan. Untuk diagram *class*, terdapat 2 *screen* tambahan yaitu ScreenDaftarBimbingan dan ScreenEdit Bimbingan, kemudian 2 *component* tambahan yaitu BimbinganDetail dan BimbinganForm beserta API tambahan yaitu BImbinganAPI.

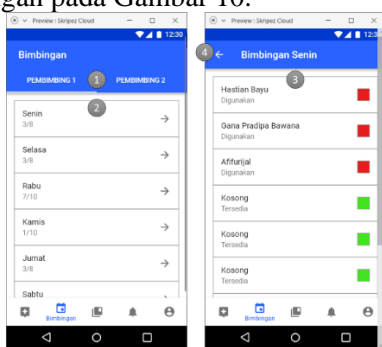
5.5 Perancangan Basis Data Iterasi Kedua

Pada iterasi kedua, CDM diagram menambahkan tabel bimbingan dengan relasi ke mahasiswa dan dosen agar fungsi bimbingan

bisa dijalankan dalam sistem. Untuk PDM diagram menjelaskan hubungan *primary key* dan *foreign key* antar tabel bimbingan dengan relasi ke mahasiswa dan dosen agar fungsi bimbingan bisa dijalankan dalam sistem.

5.6 Perancangan Antarmuka Iterasi Kedua

Pada iterasi kedua, terdapat 4 antarmuka baru yang telah disesuaikan dengan masing-masing tambahan *use case* iterasi kedua. Berikut merupakan antarmuka halaman Lihat Jadwal Bimbingan pada Gambar 10.



Gambar 10. Antarmuka Lihat Jadwal Bimbingan

Gambar di atas merupakan gambar dari halaman untuk mahasiswa dan dosen melihat jadwal bimbingan berdasarkan slot hari dalam seminggu.

6. PENGUJIAN

5.1 Pengujian Dasar

Pengujian dasar dengan *Traceability Matrix* pada setiap kolom dan baris pada *traceability matrix* memiliki nilai 1, yang menandakan bahwa *use case* terhubung secara konsisten pada artefaknya sehingga dapat dilacak antara *use case* dan artefaknya. Perancangan yang dibuat termasuk konsisten dan mangkus karena semua *use case* menghasilkan artefak menurut baris dan *use case* tidak ada yang kehilangan sumber artefak atau berasal dari requirement menurut kolom.

5.2 Pengujian Kualitas

Pengujian dilakukan menggunakan perhitungan (*Coupling between Object Classes*) CBO, (*Response for a Class*) RFC, (*Lack of cohesion in methods1*) LCOM1 metrics, dan (*Lack of cohesion in methods2*) LCOM2 metrics.

Tabel 6. Tingkat *adaptability* berdasarkan *metric* CBO, RFC, LCOM1, dan LCOM2

No.	Metric	Adaptable	Fairly Adaptable	Poorly Adaptable
1	CBO	1-3	4-5	>5
2	RFC	1-69	70-100	>100
3	LCOM1	1	-	>1
4	LCOM2	0-1	2	>2

Tabel 6 menunjukkan tingkat *adaptability* jika menggunakan metrik CBO, RFC, LCOM1, LCOM2 (Akwukwuma & Udo, 2015). Nilai *adaptability* sesuai dengan satuan CBO, RFC, LCOM1, LCOM2 dapat dibagi menjadi *adaptable* (mudah untuk diadaptasi), *fairly adaptable* (cukup dapat diadaptasi), dan *poorly adaptable* (susah untuk diadaptasi). Selain *adaptability*, nilai dari CBO dan RFC dapat mendefinisikan nilai *coupling*, dan LCOM1 dan LCOM2 memperlihatkan tingkat kohesi, dimana *coupling* yang sedikit akan terjadi hanya jika CBO kurang dari atau sama dengan 5 dan RFC kurang dari atau sama dengan 100. Sehingga pada kondisi selain itu menghasilkan *coupling* yang banyak. Kohesi tinggi terlihat ketika $LCOM1 > 1$ atau $LCOM1 = 1$ dan $LCOM2 \leq 2$. Selain dari kondisi tersebut, kohesi yang ditunjukkan memiliki nilai yang rendah (Akwukwuma & Udo, 2015).

Rata-rata pada nilai CBO yaitu 2,324, berarti penilaian CBO berada diantara 1 hingga 3, sedangkan rata-rata nilai RFC 5,324 berarti nilai RFC diantara 1 hingga 69. Dalam hal ini berarti ikatan antar fungsi pada satu komponen (*coupling*) menunjukkan kemudahan *adaptability* saat ada perubahan dalam sistem (hal ini memperlihatkan bahwa nilai dari *adaptability* tinggi). Rendahnya nilai RFC membuat mudah untuk dilakukan testing dan debug *class* akibat dari kompleksitas yang rendah. Nilai CBO yang rendah berarti memiliki *coupling* yang rendah, sehingga menunjukkan bahwa perancangan *class* mudah dipahami (hal ini memperlihatkan bahwa nilai dari *understandability* tinggi) karena modul mudah untuk dipahami dan dirubah.

Kopling yang dihasilkan termasuk rendah (rata-rata nilai $CBO \leq 5$ dan $RFC \leq 100$) karena mayoritas setiap component hanya terhubung dengan *API Service* saja sehingga membantu mengurangi kopling.

Dalam uji LCOM1 dan LCOM 2 peneliti mengambil sampel tiga kelas yaitu *ToDoList*, *BimbinganDetail*, *ReminderList*. Dari ketiga kelas yang diujikan ini untuk hasil LCOM2 berada pada nilai antara 0 hingga 1. Ini menunjukkan bahwa perancangan ini memiliki tingkat *adaptability* tinggi (yakni kemudahan

No.	Metric	Adaptable	Fairly	Poorly
-----	--------	-----------	--------	--------

dalam beradaptasi) namun pada uji LCOM 1, ketiga *class* ini mendapatkan nilai 1 yang menunjukkan memiliki tingkat *adaptability* yang buruk.

7. KESIMPULAN DAN SARAN

Kesimpulan berdasarkan hasil rekayasa kebutuhan yang dilakukan, aplikasi asistensi skripsi mahasiswa memiliki 5 macam aktor dan mempunyai 46 kebutuhan fungsional dan 1 kebutuhan non fungsional pada iterasi pertama dan 4 kebutuhan fungsional pada iterasi kedua. Dihasilkan beberapa pemodelan kebutuhan seperti *use case diagram*, *class diagram*, dan *sequence diagram*.

Kesimpulan berdasarkan hasil elisitasi kebutuhan sesuai dengan wawancara dan survei yang telah dilakukan, ditentukanlah aktor dan kebutuhan fungsional dan non-fungsional sistem. Permasalahan yang teridentifikasi yaitu mahasiswa kesulitan mencari ide skripsi, kesulitan dan takut dalam berkonsultasi, dan sering lupa dengan timeline skripsi. Kemudian beberapa hasil kebutuhan fungsional yang mampu menyelesaikan permasalahan tersebut antara lain fungsi tawaran skripsi, fungsi penjadwalan bimbingan, dan fungsi *reminder*.

Kesimpulan berdasarkan hasil rekayasa kebutuhan, diperoleh perancangan arsitektur, perancangan data, perancangan komponen, dan perancangan antarmuka. Dalam perancangan arsitektur terdapat rancangan *sequence diagram* dan *class diagram* yang dijelaskan dengan rinci. Pada perancangan data dari CDM diagram hingga PDM. Pada perancangan komponen terdapat *pseudocode* dari *class controller* dan *API service* yang digunakan pada aplikasi. Pada perancangan antarmuka berisi purwarupa gambaran aplikasi yang dibangun.

Pada hasil pengujian dasar, dapat disimpulkan bahwa semua artefak yang dibuat telah berdasarkan *use case* karena setiap kolom dan baris pada *traceability matrix* memiliki nilai x yang menunjukkan telah sesuai.

Pada hasil pengujian kualitas dapat disimpulkan bahwa aplikasi memiliki kopling rendah karena hasil dari uji CBO= 2,324 dan uji RF=5,324 yang berada pada tingkat CBO=a *daptable* dan RFC= *adaptable*. Pengujian kohesi dilakukan pada tiga *class* dengan hasil LCOM1>1 terdapat tiga *class* dan LCOM2=0 hingga 1 terdapat tiga *class*. Menurut hasil dari LCOM1 rancangan yang dibuat memiliki kohesi rendah dengan tingkat *adaptability* rendah,

namun menurut hasil dari LCOM2 rancangan memiliki kohesi tinggi dengan tingkat *adaptability* tinggi. Dari empat macam pengujian diatas, tiga teknik pengujian mendapatkan hasil dengan tingkat *adaptability* tinggi dan satu teknik pengujian mendapatkan hasil dengan tingkat *adaptability* rendah.

Saran yang diberikan untuk perancangan aplikasi asistensi skripsi mahasiswa selanjutnya yaitu sistem bisa dimodelkan untuk bisa siap digunakan mulai dari semester 5 mahasiswa FILKOM UB. Sehingga mulai mata kuliah Metodologi Penelitian, mahasiswa FILKOM UB sudah memiliki pandangan yang jelas mengenai bidang, jenis, tipe, dan produk apa yang akan diteliti oleh mahasiswa tersebut sebagai bahan skripsi.

8. DAFTAR PUSTAKA

- Adi, L., Akbar, R. J., & Khotimah, W. N. (2017). Platform e-Learning untuk Pembelajaran Pemrograman Web Menggunakan Konsep Progressive Web Apps. *Jurnal Teknik ITS*, VI(2), 645-649.
- Akwukwuma, V. V., & Udo, E. N. (2015). Predicting Adaptability Level of Object-Oriented Software Using Metrics and Threshold Values. *The Pacific Journal of Science and Technology*, 16(2), 124 - 134.
- Developers, G. (2017). *Introduction to Progressive Web App Architectures*. Tersedia di: <https://developers.google.com/web/ilt/pwa/introduction-to-progressive-web-app-architectures> [Diakses 22 Februari 2019]
- Hartasih, W., Rusdianto, D. S., & Brata, A. H. (2019). Perancangan Aplikasi pada Pengelolaan Sewa Barang. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 3(5), 4415-4423.
- Kong, L., & Yuan, T. (2009). Extension Features-Driven Use Case Model for requirement traceability. *4th International Conference on Computer Science & Education*, 866-870.
- LePage, P. (2018). *Your First Progressive Web App*. Tersedia di: <https://developers.google.com/web/fundamentals/codelabs/your-first-pwapp/> [Diakses 17 Februari 2019]
- Pressman, R. S., & Maxim, B. R. (2014). *Software Engineering: A Practitioner's*

Approach (8th Edition ed.). New York: McGraw-Hill Education.

Russell, A. (2015). *Progressive Web Apps: Escaping Tabs Without Losing Our Soul*. Tersedia di: <https://medium.com/@slightlylate/progressive-apps-escaping-tabs-without-losing-our-soul-3b93a8561955> [Diakses 16 Februari 2019]

Sommerville, I. (2011). *Software Engineering* (9th ed.). Boston: Pearson Education.