Dear all,

IF-UNIKOM-UP-RPL2-NOKELOMPOK-2018

DOKUMEN PEMBANGUNAN PERANGKAT LUNAK

KUMATO

Versi: 1.0.0-26/Final

untuk:

Matakuliah Rekayasa Perangkat Lunak 2

Dipersiapkan oleh:

Kelompok 4, 10116260 Ary Sugiarto

10116263 Farhan Arif N

10116264 Feryanto Abi F

10116236 Anisa Dewi Umi K

10116242 Ahmad Tri Utama

KELAS IF-6

Jurusan Teknik Informatika – Universitas Komputer Indonesia

Jl. Dipati Ukur Nomor 112-114, 40132

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| unikom | **Program Studi**  **Teknik Informatika UNIKOM** | Nomor Dokumen | | Halaman |
| *IF-UNIKOM-UP-RPL2-2018* | | *1/40* |
| Revisi | *0* | *Tgl: 1 November 2018* |

# DAFTAR ISI

[DAFTAR ISI 5](#_Toc534377237)

[TENTANG DOKUMEN 6](#_Toc534377238)

[1 PENDAHULUAN 7](#_Toc534377239)

[1.1 *Latar Belakang Masalah* 7](#_Toc534377240)

[1.2 *Identifikasi Masalah* 7](#_Toc534377241)

[1.3 *Maksud dan Tujuan* 8](#_Toc534377242)

[1.4 *Batasan Masalah* 8](#_Toc534377243)

[1.5 *Metodologi Penelitian* 8](#_Toc534377244)

[1.5.1 Metode Pengumpulan Data 9](#_Toc534377245)

[1.5.2 Metode Pembangunan Perangkat Lunak 13](#_Toc534377246)

[2 TINJAUAN PUSTAKA 15](#_Toc534377247)

[2.1 *Validasi Ide* 15](#_Toc534377248)

[2.2 *Javelin Board* 15](#_Toc534377249)

[2.3 *Google Design Sprint* 16](#_Toc534377250)

[2.4 *OOAD dengan UML* 16](#_Toc534377251)

[2.4.1 UML 17](#_Toc534377252)

[2.4.2 Usecase 18](#_Toc534377253)

[2.4.3 Usecase scenario 19](#_Toc534377254)

[2.4.4 Activity diagram 19](#_Toc534377255)

[2.4.5 Class diagram 20](#_Toc534377256)

[2.4.6 Sequence diagram 21](#_Toc534377257)

[2.5 *Mobile Development (Native Android/Native Ios/Progressive Web Apps)* 21](#_Toc534377258)

[2.6 *Pengujian Perangkat Lunak Dengan Heart Metric* 22](#_Toc534377259)

[2.7 *Business Model Canvas* 22](#_Toc534377260)

[3 KEBUTUHAN PERANGKAT LUNAK 23](#_Toc534377261)

[3.1 *Analisis Sistem Berjalan* 23](#_Toc534377262)

[3.2 *Analisis Arsitektur Sistem* 23](#_Toc534377263)

[3.3 Spesifikasi Kebutuhan Perangkat Lunak 24](#_Toc534377264)

[3.3.1 Kebutuhan Fungsional 24](#_Toc534377265)

[3.3.2 Kebutuhan Non Fungsional 24](#_Toc534377266)

[3.4 Analisis Kebutuhan Nonfungsional 24](#_Toc534377267)

[3.4.1 Analisis Kebutuhan Perangkat Lunak 24](#_Toc534377268)

[3.4.2 Analisis Kebutuhan Perangkat Keras 25](#_Toc534377269)

[3.4.3 Analisis Kebutuhan Perangkat Pikir 26](#_Toc534377270)

[3.4.4 Analisis Kebutuhan Jaringan 26](#_Toc534377271)

[3.5 Analisis Kebutuhan Data 26](#_Toc534377272)

[3.6 Analisis Kebutuhan Fungsional 27](#_Toc534377273)

[4 ANALISIS SISTEM 27](#_Toc534377274)

[4.1 *Model* Use Case 27](#_Toc534377275)

[4.1.1 Diagram Use Case 28](#_Toc534377276)

[4.1.2 Definisi Actor 28](#_Toc534377277)

[4.1.3 Definisi Use Case 29](#_Toc534377278)

[4.1.4 Skenario Use Case 30](#_Toc534377279)

[4.2 *Model Activity Diagram* 36](#_Toc534377280)

[DAFTAR PUSTAKA 39](#_Toc534377281)

[LAMPIRAN 40](#_Toc534377282)

# PENDAHULUAN

## *Latar Belakang Masalah*

Produktivitas berarti kemampuan menghasilkan sesuatu. Sedangkan kerja berarti kegiatan melakukan sesuatu yang dilakukan untuk mencari nafkah mata pencahrian (Poerwadarminta, 1984 : 70). Segala macam metode untuk meningkatkan produktivitas seperti yang dilansir Kompas.com, ada beberapa hal yang harus anda hindari agar dapat lebih produktif diantaranya memprioritaskan kerja ketimbang tidur, meletakan ponsel di sebelah bantal, tidak sarapan, mengerjakan banyak hal dalam satu waktu dan terlalu banyak kegiatan.

Berdasarkan hasil wawancara dari 14 orang responden yang kita wawancarai selama 2 pekan di sekitar lingkungan kampus UNIKOM (Universitas Komputer Indonesia) di temukan masalah utama yang sering di hadapi oleh mahasiswa. Masalah pertama dari responden yang sudah di wawancarai mengatakan sering lupa mengerjakan tugas atau pekerjaannya. Hal ini di perkuat dari hasil wawancara sebanyak 80% responden memiliki kebiasaan sering lupa mengerjakan tugas dan pekerjaannya, yang disebabkan karena kurang fokus dan memiliki kekhawatiran tugas atau pekerjaannya tidak selesai. Dari 14 responden yang kita wawancara terdapat 60% responden tidak megetahui adanya teknik pomodoro (Teknik peningkatan kefokusan).

Berdasarkan pemaparan fakta dan masalah yang telah dijelaskan sebelumnya, ditemukan suatu teknologi yang diharapkan menjadi solusi yaitu dengan membangun sebuah perangkat lunak bernama KUMATO berbasis Mobile (Android) untuk membantu mengingatkan kegiatan atau pekerjaan yang akan dilakukan serta meningkatkan kefokusan dalam mengerjakan sesuatu.

## *Identifikasi Masalah*

Dari latar belakang masalah yang telah dijelaskan, dapat diidentifikasi masalah yang terjadi adalah sebagai berikut:

1. Sering lupa tugas dan pekerjaan
2. Kurang fokus dalam mengerjakan tugas dan pekerjaan

## *Maksud dan Tujuan*

Maksud dari penelitian ini adalah dengan membangun perangkat lunak KUMATO berbasis Mobile (Android) dengan tujuan sebagai berikut:

1. Memudahkan mahasiswa untuk mengingat tugas dan pekerjaan.
2. Memudahkan mahasiswa untuk meningkatkan produktivitas dengan cara lebih fokus.

## *Batasan Masalah*

1. Mahasiswa yang sering lupa.
2. Mahasiswa dengan umur 19 – 25 tahun.
3. Mahasiswa dengan jenis kelamin laki-laki atau perempuan.
4. Kondisi fisik dan psikisnya normal.
5. Area Bandung.

## *Metodologi Penelitian*

Metodologi penelitian yang digunakan dalam penelitian ini adalah metode-metode deskriptif dimana metode ini digunakan dalam penelitian awal untuk menghimpun data tentang kondisi yang ada, faktor pendukung dan penghambat pengembangan, serta penggunaan produk dimana produk tersebut akan diterapkan. Metode ini diambil dari sebuah paper penelitian pendidikan karya Surya Dharma MPA., Ph. D dimana metode ini termasuk kedalam Penelitian dan Pengembangan atau *Research and Development* (R&D) yang merupakan rangkaian proses atau langkah-langkah dalam rangka mengembangkan satu produk baru atau menyempurnakan produk yang telah ada agar dapat dipertanggungjawabkan. Penelitian dan pengembangan merupakan metode penghubung atau pemutus kesenjangan antara penelitian dasar dengan penelitian terapan.

Metode penelitian ini memiliki dua metode dalam pelaksanaanya yaitu metode pengumpulan data dan metode pembangunan perangkat lunak. Berikut adalah pemaparannya:

### Metode Pengumpulan Data

Berikut adalah metode pengumpulan data dalam penelitian ini:

1. Studi Literatur

Pengumpulan data dilakukan adalah dengan cara mempelajari, meneliti, dan menelaah berbagai literatur dari perpustakaan yang bersumber dari buku buku, jurnal ilmiah, situs internet, dan bacaan lainnya yang berkaitan dengan penelitian yang dilakukan seperti {judul teori-teori di tinjauan pustaka}.

Literatur mengenai “Lupa”. Lupa adalah suatu kondisi dimana suatu informasi yang telah disimpan dalam memori otak jangka panjang hilang. Menurut Kamus Besar Bahasa Indonesia (KBBI) lupa adalah lepas dari ingatan; tidak dalam pikiran, tidak teringat, tidak sadar. Lupa, adalah suatu hal yang semua orang pernah mengalaminya baik muda maupun tua. “Manusia adalah tempatny salah dan lupa” pernyataan tersebut sering kita dengar, tetapi bukan berarti kita dapat menggunakan kalimat – kalimat tersebut secara sembarangan sebagai alasan utnuk pasrah dan menyalahkan keadaan saat kita lupa serta menjadikan kita malas mengingat kembali.

Gulo (1982) dan Reber (1988) mendefinisikan lupa sebagai ketidakmampuan mengenal atau mengingat sesuatu yang pernah dipelajari. Jadi lupa bukanlah peristiwa hilangnya item informasi dan pengetahuan dari akal kita

Muhibbinsyah (1996) dalam bukunya yang berjudul psikologi pendidikan mengartikan lupa sebagai hilangnya kemampuan untuk menyebut kembali atau memproduksi kembali apa – apa yang sebelumnya kita pelajari secara sederhana.

Menurut Hallosehat, lupa merupakan hal yang wajar dikarenakan informasi yang diterima belum sampai ke ingatan atau otak terdalam, atau anda memang sedang tidak fokus dalam mengingat hal yang baru anda pelajari.

Banyak penyebab yang dapat memengaruhi kemampuan otak untuk mengingat :

* Kurang Tidur
* Depresi dan Stress
* Usia
* Terlalu banyak konsumsi lemak jenuh
* Jarang Olahraga

Literatur Teknik Pomodoro. Menurut Wikipedia, Teknik Pomodoro adalah metode manajemen waktu yang dikembangkan oleh Francesco Cirillo diakhir tahun 1980. Teknik ini menggunakan timer untuk menyelesaikan pekerjaan dalam beberapa interval dengan waktu selama 25 menit, dipisahkan dengan istirahat pendek. Interval ini diberi nama Pomodoro, yang merupakan bahasa Italia dari tomat. Setelah Cirillo menggunakan timer memasak berbentuk tomat terbelah setengah di Universitas.

Teknik ini semakin populer karena lusinan aplikasi dan website yang menyediakan timer dan petunjuk menggunakannya. Konsep ini terhubung dengan konsep timeboxing dan pengembangan Iteratif dan incremental yang digunakan dalam desain perangkat lunak, metode ini diadopsi dalam Pair Programming.

1. Wawancara

|  |  |
| --- | --- |
| Rekaputlasi Interview Kumato tahap pertama | |
| Minggu 1 Oktober 2018 |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Tempat | Valid | Total Wawancara | Nama | Problem Walkthrough | | |
| Pembatasan | Pengingat | Disiplin |
| Kosan Belakang Unpad | 1 | 1 | Fuad | Ya | Ya | Ya |
| Kosan Gang Amat 1 | 3 | 3 | Dodi | Ya | Ya | Ya |
| Dalih | Ya | Ya | Ya |
| Mufti | Ya | Ya | Ya |
| UNIKOM | 5 | 6 | Ariza | Ya | Ya | Ya |
| Lutfi | Ya | Ya | Ya |
| Rica | Ya | Ya | Ya |
| Dio | Ya | Ya | Ya |
| Adit | Ya | Ya | Ya |
| Annatsa | Ya | Tidak | Ya |
| Kosan Belakang Unikom | 0 | 1 | Andre | Ya | Tidak | Tidak |
| Kosan Putri | 1 | 1 | Yuka | Ya | Ya | Ya |
| Muslimah center | 0 | 1 | Juniarti | Ya | Tidak | Tidak |
| Tikah | Ya | Tidak | Tidak |

Rekaputlasi Interview Kumato tahap kedua

Minggu 24 Oktober 2018

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Tempat | Valid | Total Wawancara | Nama | Pendidikan | Problem Walkthrough | | |
| S1 | Pembatasan | Pengingat | Disiplin |
| Kosan Belakang Unikom | 2 | 2 | Andre | Ya | Ya | Tidak | Ya |
| Adit | Ya | Ya | Ya | Ya |
| Unikom | 6 | 6 | Suryo | Ya | Ya | Ya | Ya |
| Djauhari | Ya | Ya | Ya | Ya |
| Yuka | Ya | Ya | Ya | Ya |
| Dio | Ya | Ya | Ya | Ya |
| Annatsa | Ya | Ya | Ya | Ya |
| Aliraf | Ya | Ya | Ya | Ya |
| Kosan Sekeloa | 3 | 3 | Lutfi | Ya | Ya | Ya | Ya |
| Dalih | Ya | Ya | Ya | Ya |
| Mufti | Ya | Ya | Ya | Ya |

Daftar pertanyaan wawancara yang penting seputar inti masalah:

1. Tanyakan nama dan informasi pribadi
2. Menanyakan pekerjaan
3. Apakah sering lupa mengerjakan tugas?
4. Sebab lupa mengerjakan tugas
5. Pernah tidak mengalami ketidakfokusan dalam mengerjakan sesuatu?

Kira-kira saat mengerjakan apa?

1. Apakah anda mengetahui, teknik agar bisa tetap fokus dalam mengerjaan sesuatu?
2. Apakah anda mengetahui, teknik pomodoro?

Jika mengetahui, maka meminta responden untuk menjelaskan. Jika tidak mengetahui, maka pewawancara akan menjelaskan.

1. Menanyakan keinginan responden terhadap penerapan aplikasi.

### Metode Pembangunan Perangkat Lunak

Pada pembangunan perangkat lunak ini kami menggunakan Metode Google Sprint. Metode Google Design Sprint adalah Design sprint adalah suatu metode untuk build product concept dan prototype dalam waktu 5 hari dimana memiliki 5 tahapan yang sangat komunikatif dan interaktif untuk mengeluarkan semua ide, inspirasi, masalah yang ada, solusi yang kemudian diwujudkan dalam prototype yang harus dan di cross check ke calon pengguna[9].

Berikut ini tahapan yang sudah kami lakukan, di antaranya:

**Tahapan Pertama adalah Validasi Ide :**

Validasi Ide

* Wawancara
* Mencari masalah
* Mengumpulkan informasi mengenai keinginan pengguna dalam produk yang akan di bangun.

**Tahapan Kedua adalah Design Sprint :**

Kami mempraktikkan langsung bagaimana menggunakan metode Google Design Sprint dalam membangun produk dengan waktu yang sedikit.

Tahapannya adalah sebagai berikut :

1. Understand

Pada hari pertama, intinya adalah pengenalan tiap anggota, persiapan yang harus dilakukan dan mendiskusikan bagaimana cara menyelesaikan masalah dan hasil wawancara tersebut menjadi bahan diskusi bersama dengan semua anggota tim. Dari hasil diskusi tersebut kita dapat menentukan rencana jangka panjang dan masalah yang mungkin akan datang, dari hal tersebut kita membuat MAP atau alur aplikasi yang akan dibuat dan disempurnakan dengan mendiskusikan solusi berupa How Might We Notes (HMW).

1. Diverge

Pada hari kedua, Membuat sketsa aplikasi. Tiap anggota membuat sketsa sesuai ide masing – masing yang dilakukan setelah adanya penggambaran demo aplikasi

1. Decide

Masih dihari kedua. Setelah membuat sketsa, tiap anggota menjelaskan design sketsa yang telah dibuat . Tim berkumpul untuk melakukan voting, dimana ide terbaik diputuskan oleh suara terbanyak akan tetapi produk ownerlah yang mempunyai keputusan mutlak. Setelah mendapatkan hasil yang relevan, maka dibuatlah storyboard untuk tahap prototype.

1. Prototype

Pada hari ketiga, designer akan membuat prototipe dari design yang sudah dibuat dan disepakati pada hari sebelumya. Lalu tim akan mencoba maupun menganalisis prototype yang sudah dibuat sebelum dicoba oleh pengguna. Prototype ini hanya sekedar menampilkan fitur utama dari aplikasi yang akan dibuat, yang akan digunakan oleh pengguna kedepannya.

1. Validate

Prototype yang sudah jadi pada hari sebelumnya, pada hari keempat ini kita akan terjun kelapangan untuk meminta pendapat dari calon pengguna. Apa saja kekurangan dan kelebihan dari hasil prototype dalam sudut pandang pengguna.

# TINJAUAN PUSTAKA

## *Validasi Ide*

Validasi ide adalah proses awal yang harus dilakukan sebelum meluncurkan nama bisnis, slogan, produk, layanan, atau aplikasi baik yang berbasis website ataupun Mobile. Ini seperti proses yang digunakan perusahaan besar untuk melakukan penelitian dan pengembangan untuk menguji ide produk sebelum dirilis ke masyarakat umum. Penelitian dapat mencakup survei, wawancara, riset online, dan sebagainya untuk mendapatkan hasil terbaik [2].

## *Javelin Board*

Javelin Board atau Papan Validasi merupakan salah satu cara untuk validasi ide startup menguraikan menggunakan Experiment Board. Dimana validasi ide merupakan salah satu tahap krusial yang menentukan. Dari analisis menggunakan Javelin Board kita dapat mengetahui titik kritis, identifikasi masalah, asumsi, melakukan metode pengujian dan juga penentuan kriteria sukses. Board ini terbagi menjadi dua bagian, yaitu bagian kiri atau disebut dengan area Brainstorming. Dan bagian kanan yang disebut dengan Area Execute [1].



Gambar 1 *Javelin Board*

Langkah pertama yang harus di ketahui adalah, menentukan calon konsumen untuk produk yang dibuat. Kumpulkan beberapa calon potensial, kemudian pilih salah satu yang akan menjadi fokusnya. Untuk tahap selanjutnya di javelin Board sudah terdapat instruksi secara singkat supaya mempermudah kita dalam pengisian [1].

## *Google Design Sprint*



Gambar 2 Google Design sprint

Design sprint adalah suatu metode untuk build product concept dan prototype dalam waktu 5 hari dimana memiliki 5 tahapan yang sangat komunikatif dan interaktif untuk mengeluarkan semua ide, inspirasi, masalah yang ada, solusi yang kemudian diwujudkan dalam prototype yang harus dan di cross check ke calon pengguna[9]. Design Sprint ini dibuat oleh Jake Knapp dari Google Venture pada tahun 2010[12]. Sehingga disebut dengan Google Design Sprint.

## *OOAD dengan UML*

OOAD adalah metode analisis yang memerikasa requirements dari sudut pandang kelas kelas dan objek yang ditemui dalam ruang lingkup permasalahan yang mengarahkan arsitektur software yang didasarkan pada manipulasi objek-objek system atau subsistem[11]. Sekitar akhir tahun 80-an dan awal tahun 90-an, UML (Unified Modeling Language) menjadi pengganti dari metode analisis berorientasi object dan design berorientasi object(OOAD&D/object oriented analysis and design)[13].

### UML



Gambar 3 Unified Modeling Language

Menurut Booch (2005:7) UML adalah Bahasa standar untuk membuat rancangan software. UML biasanya digunakan untuk menggambarkan dan membangun, dokumen artifak dari software –intensive system [3].

Menurut Nugroho (2010:6), UML (Unified Modeling Language) adalah ‘bahasa’ pemodelan untuk sistem atau perangkat lunak yang berparadigma ‘berorientasi objek”. Pemodelan (modeling) sesungguhnya digunakan untuk penyederhanaan permasalahan-permasalahan yang kompleks sedemikian rupa sehingga lebih mudah dipelajari dan dipahami [3].

Menurut Nugroho (2009:4), UML (Unified Modeling Language) adalah Metodologi kolaborasi antara metoda-metoda Booch, OMT (Object Modeling Technique), serta OOSE (Object Oriented Software Enggineering) dan beberapa metoda lainnya, merupakan metodologi yang paling sering digunakan saat ini untuk analisa dan perancangan sistem dengan metodologi berorientasi objek mengadaptasi maraknya penggunaan bahasa “pemrograman berorientasi objek” (OOP) [3].

Menurut Herlawati (2011:10), bahwa beberapa literature menyebutkan bahwa UML menyediakan sembilan jenis diagram, yang lain menyebutkan delapan karena ada beberapa diagram yang digabung, misanya diagram komunikasi, diagram urutan dan diagram pewaktuan digabung menjadi diagram interaksi [3].

### Usecase

|  |  |  |  |
| --- | --- | --- | --- |
| **NO** | **GAMBAR** | **NAMA** | **KETERANGAN** |
| 1 | <https://4.bp.blogspot.com/-DmmpR4cj7GU/UrWedi5CfcI/AAAAAAAAEQo/L14DTlPUWkM/s1600/SP32-20131221-205540.jpg> | *Actor* | Menspesifikasikan himpuan peran yang pengguna mainkan ketika berinteraksi dengan *use case*. |
| 2 | <https://1.bp.blogspot.com/-YPtVTcMBAXI/UrWerJKJuiI/AAAAAAAAEQw/bGREX-RwKiE/s1600/SP32-20131221-205548.jpg> | *Dependency* | Hubungan dimana perubahan yang terjadi pada suatu elemen  mandiri *(independent)* akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (*independent*). |
| 3 | *<https://4.bp.blogspot.com/-ONkVKj3uLZI/UrWezeqXgRI/AAAAAAAAEQ4/JkJQXjJSR08/s1600/SP32-20131221-205555.jpg>* | *Generalization* | Hubungan dimana objek anak (*descendent*) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor*). |
| 4 | *<https://4.bp.blogspot.com/-ZzSmtK9CU-g/UrWezrFPxjI/AAAAAAAAEQ8/N_DAeYFrdjE/s1600/SP32-20131221-205601.jpg>* | *Include* | Menspesifikasikan bahwa *use case*sumber secara *eksplisit*. |
| 5 | *<https://1.bp.blogspot.com/-jiup8MT1hXE/UrWe0FKtvcI/AAAAAAAAERI/rVn-_4lG9kc/s1600/SP32-20131221-205608.jpg>* | *Extend* | Menspesifikasikan bahwa *use case*target memperluas perilaku dari *use case* sumber pada suatu titik yang diberikan. |
| 6 | *<https://2.bp.blogspot.com/-T-kfx0KI3Gg/UrWe0I3BkRI/AAAAAAAAERQ/DRTJba0PtIk/s1600/SP32-20131221-205614.jpg>* | *Association* | Apa yang menghubungkan antara objek satu dengan objek lainnya. |
| 7 | *<https://1.bp.blogspot.com/-CVHs3Tv-31Y/UrWe0M3xtDI/AAAAAAAAERM/4m7aP5gxIcE/s1600/SP32-20131221-205620.jpg>* | *System* | Menspesifikasikan paket yang menampilkan sistem secara terbatas. |
| 8 | *<https://1.bp.blogspot.com/-eidvEXFUJtU/UrWe02fSVdI/AAAAAAAAERU/AMyp_ZDSndQ/s1600/SP32-20131221-205627.jpg>* | *Use Case* | Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor |
| 9 | *<https://1.bp.blogspot.com/-ORllnBInnag/UrWe1Gz64YI/AAAAAAAAERo/9wVR-BYAHyI/s1600/SP32-20131221-205633.jpg>* | *Collaboration* | Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan prilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi). |
| 10 | *<https://3.bp.blogspot.com/-jXsDdS5d2PA/UrWe1Nu9B0I/AAAAAAAAERs/IpIDKWf9HvM/s1600/SP32-20131221-205639.jpg>* | *Note* | Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi |

Diagram Use Case atau *Use Case Diagram* adalah pemodelan untuk menggambarkan *behavior* / kelakuan sistem yang akan dibuat [2]. Selain itu, diagram use case juga digunakan untuk memahami fungsi apa saja yang ada di dalam sebuah sistem dan siapa saja yang dapat menggunakan fungsi-fungsi tersebut.

### Usecase scenario

Skenario use case / *use case skenario* adalah alur jalannya proses use case dari sisi aktor dan system [2]. Alur skenario inilah yang nantinya menjadi landasan pembuatan sequence diagram / diagram sekuen.

### Activity diagram



Gambar 4 Notasi Activity Diagram

*Activity diagrams* adalah diagram yang menggambarkan berbagai aliran aktivitas dalam sebuah sistem yang sedang dirancang dan bagaimana masing-masing aliran berawal, keputusan yang mungkin terjadi, dan bagaimana aktivitas tersebut berakhir [3]. Diagram aktivitas ini memiliki beberapa komponen dengan bentuk tertentu yang dihubungan dengan tanda panah. Dimana tanda panah tersebut mengarahkan urutan aktivitas yang terjadi dari awal sampai akhir

### Class diagram



Gambar 5 Notasi Class Diagram

*Class diagram* adalah diagram yang menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem [4]. Kelas ini memiliki 3 bagian utama yaitu *attribute, operation*, dan *name*. Kelas-kelas yang ada pada struktur sistem harus dapat melakukan fungsi-fungsi sesuai dengan kebutuhan system.

### Sequence diagram



Gambar 6 Notasi Sequence Diagram

*Sequence diagram* ini adalah diagram yang menggambarkan kolaborasi dinamis antara sejumlah object [5]. *Sequence diagram* juga digunakan untuk menunjukkan rangkaian pesan yang dikirim antara object juga interaksi antara object.

## *Mobile Development (Native Android/Native Ios/Progressive Web Apps)*

Aplikasi Native adalah aplikasi yang di bangun dengan bahasa pemrograman yg spesifik tertentu platform tertentu, dan di pembangunan perangkat lunak ini berbasis android maka bahasa yg akan digunakan yaitu java.

Java adalah bahasa pemrograman tingkat tinggi yang dikembangkan oleh sun microsystems dan dirilis pertama kali pada tahun 1995 [6]. Pemrograman java ini dapat berjalan di berbagai platform seperti Windows, Mac dan bermacam-macam versi UNIX. Pemrograman java juga semakin banyak peminatnya karena dengan menguasai pemrograman java, berarti akan mudah untuk membuat sebuah aplikasi android.

## *Pengujian Perangkat Lunak Dengan Heart Metric*

Silahkan gunakan tools <https://www.appsee.com/> atau sejenisnya.

## *Business Model Canvas*

Model Bisnis Kanvas adalah merupakan salah satu strategi bisnis yang dapat mendesain, menggambarkan hingga menyimpulkan aspek – aspek bisnis menjadi satu strategi yang utuh. Model Bisnis Kanvas mencakup 9 area atau 9 blok yang akan membantu seseorang dalam merumuskan ide bisnis mereka menjadi suatu model bisnis yang real. [10]

# KEBUTUHAN PERANGKAT LUNAK

Dalam pembuatan perangkat lunak pastinya diperlukan kebutuhan yang akan digunakan dala pembangunan perangkat lunak itu sendiri dan disini akan dibahas kebutuhan-kebutuhan yang akan digunakan dalam pembangunan perangkat lunak.

## *Analisis Sistem Berjalan*

* Ketika seseorang memerlukan sebuah pengingat, dia akan membuat pengingat dengan menuliskan kegiatan yang akan dilakukan pada sebuah kertas dan akan di tempel pada temat yang biasa mereka lihat;
* Membuat pengingat waktu di handphone atau smartphone mereka;
* Mereka akan bertanya kepada teman kantor atau kuliahnya. Apakah ada pekerjaan atau tugas yang harus dikerjakan;
* Jika masih banyak pekerjaan yang harus di lakukan. Maka jangan lupa untuk istirahat;
* Janagan mengerjakan banyak hal dalam satu waktu.

## *Analisis Arsitektur Sistem*



Gambar 7 Arsitektur Sistem

## Spesifikasi Kebutuhan Perangkat Lunak

Spesifikasi kebutuhan perangkat lunak dibagi menjadi 2 kategori kebutuhan, yaitu kebutuhan fungsional dan kebutuhan non fungsional. Berikut ini kebutuhan fungsional dan kebutuhan non fungsional pada Aplikasi Kumato :

### Kebutuhan Fungsional

SRS-F-001 Sistem menyediakan fasilitas bagi pengguna untuk menambahkan list kegiatan

SRS-F-002 Sistem menyediakan fasilitas bagi pengguna untuk mengatur waktu pengingat kegiatan

SRS-F-003 Sistem menyediakan fasilitas bagi pengguna untuk menggunakan teknik pomodoro dengan otomatis.

### Kebutuhan Non Fungsional

SRS-NF-001 Sistem memberikan respon terhadap aksi pengguna yaitu menampilkan data list kegiatan

SRS-NF-001 Sistem memberikan respon terhadap aksi pengguna yaitu menampilkan notifikasi kegiatan dengan waktu yang telah di tentukan.

## Analisis Kebutuhan Nonfungsional

Kebutuhan nonfungsional dibagi menjadi beberapa bagian, yaitu kebutuhan perangkat lunak, kebutuhan perangkat keras, kebutuhan perangkat pikir, dan kebutuhan jaringan. Berikut ini analisis kebutuhan nonfunsional pada Aplikasi Kumato :

### Analisis Kebutuhan Perangkat Lunak

Aplikasi Kumato yang akan di bangun ini adalah sebuah aplikasi berbasis mobile android yang tidak dapat secara langsung mengakses system database. Aplikasi Kumato dapat menampilkan list data kegiatan setelah pengguna melakukan pengisian data yang akan dilakukan atau dikerjakan. Setelah itu Aplikasi Kumato akan melakukan request data pada system database untuk ditampilkan pada halaman utama. Pengguna akan mendapat pemberitahuan tentang kegiatan yang harus dikerjakan sesuai dengan waktu dan tanggal yang telah ditentukan. Pengguna dapat melalukan perubahan data jika melakukan kesalahan di saat data sudah tersimpan dan ditampilkan pada halaman utama, pengguna jugs dapat menghapus list data kegiatan yang ketika sudah dikerjakan.

Pada aplikasi Kumato ini terdapat fitur lainnya yang dapat digunakan yang berguna untuk meningkatkan kefokusan dalam mengerjakan sesuatu. Pengguna akan diberikan waktu selama 25 menit dalam hitungan mundur utuk mengerjakan sesuatu, jika waktu pertama ini telah selesai maka pengguna akan mendapat pemberitahuan dan diberikan waktu istirahat selama 5 menit. Ketika pengguna telah melakukannya selama 4 sesi, makan akan deberikan waktu istirahat sesidit lebih lama.

### Analisis Kebutuhan Perangkat Keras

Platform : Mobile Android

Procesor : ARM Cortex-A7 1.0 GHz

Ram : 512 Mb

Memory : 4GB

Layar : 3.5’’

### Analisis Kebutuhan Perangkat Pikir

|  |  |  |  |
| --- | --- | --- | --- |
| **Costumer** | **Umur** | **Behavior** | **Psycologis** |
| Mahasiswa | 19 – 25 Tahun | Butuh adanya pengingat, Biasa menggunakan Smartphone, membutuhkan fokus dan tetap santai. | Kondisinya Normal. |

### Analisis Kebutuhan Jaringan

Menggunakan SQlite yang berfungsi untuk menampung data yang di inputkan, data yang berhasil di inputkan akan di tampilkan di halaman utama aplikasi dalam bentuk list data.

## Analisis Kebutuhan Data

Analisis data yang menjadi acuan dalam membangun perangkat lunak ini adalah dokumen-dokumen yang berhubungan dengan donasi serta objek-objek lain yang terkait. Berikut adalah analisis data dari pembangunan perangkat lunak ini:

Tabel III.1 Tabel Entitas

| **ENTITAS** | **ATRIBUT** |
| --- | --- |
| Kegiatan | = {deskripsi\_kegiatan, tanggal\_waktu} |
|  |  |
|  |  |

Dari entitas tersebut, asumsi dalam penelitian ini adalah:

1. Satu kegiatan hanya dapat menggunakan 1 waktu pengingat.

Konsep analisis data yang digunakan adalah dengan menggunakan konsep ORM (*Object Relational Mapping*) dimana ORM ini akan memetakan sebuah objek dengan *database*. Dengan kata lain, ORM ini akan membantu menjadi jembatan antara objek yang didefinisikan dalam kode program dengan *database*. Objek-objek yang akan terlibat dalam ORM ini adalah objek mahasiswa.

## Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional dalam penelitian ini menggunakan pendekatan OOAD (*Object Oriented Analysis and Design*). Pemaparan lebih detail akan dipaparkan pada BAB 4.

# ANALISIS SISTEM

Analisis sistem bertujuan untuk mengidentifikasi permasalahan-permasalahan yang terdapat pada sistem serta menentukan kebutuhan-kebutuhan dari sistem yang akan dibangun. Analisis tersebut meliputi analisis masalah, analisis sistem yang sedang berjalan, analisis arsitektur sistem, spesifikasi kebutuhan non fungsional, analisis kebutuhan non fungsional, analisis data, dan analisis kebutuhan fungsional. Dalam analisis kebutuhan fungsional ini, pendekatan analisis yang digunakan adalah dengan OOAD. Diagram yang akan digambarkan pada bagian ini adalah *Use case diagram, activity diagram, class diagram,* dan *Sequence Diagram.*

## *Model* Use Case

Pada bagian model *usecase* ini akan dipaparkan mengenai gambar diagram *usecase* keseluruhan, definisi aktor, definisi *usecase* dan skenario *usecase*.

### Diagram Use Case



Gambar 8 Usecase Diagram

### Definisi Actor

Tabel 1‑1 Daftar Aktor

| **NO** | **AKTOR** | **DESKRIPSI** |
| --- | --- | --- |
| *1* | *User (Mahasiswa)* | *Aktor dengan role ini mempunyai wewenang untuk melakukan penambahan list kegiatan, ubah list kegiatan dan hapus list kegiatan serta mengatur waktu dan tanggal reminder nya. Dan juga aktor ini dapat menggunakan sebuah fitur yang ada pada aplikasi yaitu fitur Teknik Pomodoro.* |
| *2* | *Database* | *Aktor dengan role ini mempunyai wewanang untuk memberikan atau menyimpan data yang direquest atau ditambahkan pada Aplikasi Kumato.* |

### Definisi Use Case

Tabel 1‑2 Daftar *Usecase*

| **NO** | **USE CASE** | **DESKRIPSI** |
| --- | --- | --- |
| *1* | Tambah Kegiatan | *User menambah sebuah kegiatan baru dan sistem menampilkan daftar list kegiatan* |
| *2* | Ubah Jadwal Kegiatan | *User mengubah jadwal kegiatan dan sistem menampilkan form ubah list kegiatan* |
| *3* | Hapus Jadwal Kegiatan | *User menghapus jadwal kegiatan dan sistem menampilkan form ubah list kegiatan* |
| *4* | Notifikasi Kegiatan. | *Sistem menampilkan notifikasi pengingat waktu kegiatan* |
| *5* | Penyediaan Informasi Teknik Pomodoro | *Akan menampilkan detail tentang teknik pomodoro.* |
| *6* | Penggunaan Teknik Pomodoro | *Sistem menjalan timer teknik pomodoro* |
| *7* | Request data | *Sistem menampilkan daftar list kegiatan* |
| *8* | Notifikasi Penggunaan Teknik Pomodoro | *Sistem menampilkan notifikasi bahwa kegiatan telah selesai dikerjakan.* |

### Skenario Use Case

Tabel 1.3 *Usecase Scenario* Tambah Kegiatan

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Tambah Kegiatan | |
| **Related Requirement** | - | |
| Goal In Context | Menambahkan data kegiatan ke dalam aplikasi | |
| Description | Fungsionalitas ini digunakan oleh user untuk menambahkan data kegiatan ke dalam database aplikasi | |
| Related Use Case | Request data kegiatan | |
| Preconditions | - | |
| Successful End Condition | Aktor berhasil menambahkan data kegiatan | |
| Failed End Condition | Aktor tidak menambahkan data kegiatan | |
| Primary Actors | User (Mahasiswa) | |
| Secondary Actors | - | |
| Trigger | User memilih/menekan tombol berwarna merah dan berbentuk “+” | |
| Main Flow | Step | Action |
|  | 1. | User memilih tombol berwarna merah dan berbentuk “+” |
|  | 2. | Sistem menampilkan halaman penginputan data kegiatan |
|  | 3. | User mengisi data kegiatan berupa nama kegiatan beserta waktu remindernya |
|  | 4.  include:: CRUD data kegiatan | Sistem menyimpan data kegiatan baru ke dalam database |
|  | 5. | Sistem menampilkan halaman home yang berisi list data kegiatan terbaru |
| Extension | Step | Branching Action |
|  | 4.1 | Sistem gagal menyimpan data kegiatan |
|  | 5.1 | Sistem gagal menampilkan halaman home yang berisi list data kegiatan terbaru |

Tabel 1.4 *Usecase Scenario* Ubah Kegiatan

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Ubah Kegiatan | |
| **Related Requirement** | - | |
| Goal In Context | Mengubah data kegiatan yang sudah tersimpan | |
| Description | Fungsionalitas ini digunakan oleh user untuk mengubah data kegiatan dalam aplikasi | |
| Related Use Case | Request data kegiatan | |
| Preconditions | - | |
| Successful End Condition | Aktor berhasil mengubah data kegiatan | |
| Failed End Condition | Aktor tidak mengubah data kegiatan | |
| Primary Actors | User (Mahasiswa) | |
| Secondary Actors | - | |
| Trigger | User memilih/menekan kegiatan yang akan diubah | |
| Main Flow | Step | Action |
|  | 1. | User memilih/menekan kegiatan yang akan diubah |
|  | 2.  Include:: Request data kegiatan | Sistem melakukan permintaan detail data kegiatan ke database |
|  | 3. | Sistem menampilkan halaman detail data kegiatan |
|  | 4. | User mengubah detail data kegiatan diantaranya nama kegiatan atau waktu remindernya |
|  | 5.  Include:: CRUD data kegiatam | Sistem akan menyimpan data kegiatan yang sudah diubah ke database |
|  | 6. | Sistem menampilkan halaman home yang berisi list data kegiatan terbaru |
| Extension | Step | Branching Action |
|  | 2.1 | Sistem gagal mengambil detail data kegiatan |
|  | 5.1 | Sistem gagal menyimpan detail data kegiatan |
|  | 6.1 | Sistem gagal menampilkan halaman home yang berisi list data kegiatan terbaru |

Tabel 1.5 *Usecase Scenario* Hapus Jadwal Kegiatan

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Hapus Kegiatan | |
| **Related Requirement** | - | |
| Goal In Context | Menghapus data kegiatan yang sudah tersimpan | |
| Description | Fungsionalitas ini digunakan oleh user untuk menghapus data kegiatan dalam aplikasi | |
| Related Use Case | Request data kegiatan | |
|  | - | |
| Successful End Condition | Aktor berhasil menghapus data kegiatan | |
| Failed End Condition | Aktor tidak menghapus data kegiatan | |
| Primary Actors | User (Mahasiswa) | |
| Secondary Actors | - | |
| Trigger | User memilih/menekan tombol berbentuk tong sampah | |
| Main Flow | Step | Action |
|  | 1. | User memilih/menekan kegiatan yang akan dihapus |
|  | 2.  Include:: Request data kegiatan | Sistem melakukan permintaan detail data kegiatan ke database |
|  | 3. | Sistem menampilkan halaman detail data kegiatan |
|  | 4. | User memilih/menekan tombol berbentuk tong sampah yang berada di pojok kanan atas |
|  | 5.  Include:: CRUD data kegiatam | Sistem akan menghapus data kegiatan yang dipilih pada database |
|  | 6. | Sistem menampilkan halaman home yang berisi list data kegiatan terbaru |
| Extension | Step | Branching Action |
|  | 2.1 | Sistem gagal mengambil detail data kegiatan |
|  | **4.1** | **Sistem menampilkan validasi penghapusan** |
|  | 5.1 | Sistem gagal menyimpan detail data kegiatan |
|  | 6.1 | Sistem gagal menampilkan halaman home yang berisi list data kegiatan terbaru |

Tabel 1.6 *Usecase Scenario* Penggunaan Notifikasi Kegiatan

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Penggunaan Notifikasi Kegiatan | |
| **Related Requirement** | - | |
| Goal In Context | Mengingatkan kegiatan yang telah dibuat | |
| Description | Fungsionalitas ini digunakan oleh user untuk membantu mengingatkan kegiatan melalui aplikasi | |
| Related Use Case | - | |
| Preconditions | Sudah memasuki waktu reminder kegiatan | |
| Successful End Condition | Menampilkan notifikasi kegiatan | |
| Failed End Condition | Notifikasi kegiatan tidak muncul | |
| Primary Actors | User (Mahasiswa) | |
| Secondary Actors | - | |
| Trigger | - | |
| Main Flow | Step | Action |
|  | 1. | Sistem menampilkan notifikasi kegiatan |
|  | 2. | User menekan notifikasi kegiatan |
|  | 3. | Sistem menampilkan halaman Home pada aplikasi |
| Extension | Step | Branching Action |
|  | 1.1 | Sistem gagal menampilkan notifikasi aplikasi |

Tabel 1.7 *Usecase Scenario* Penggunaan Teknik Pomodoro

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Penggunaan Teknik Pomodoro | |
| **Related Requirement** | - | |
| Goal In Context | Menggunakan teknik pomodoro dengan baik | |
| Description | Fungsionalitas ini digunakan oleh user untuk menggunakan teknik pomodoro dengan baik | |
| Related Use Case | - | |
| Preconditions | - | |
| Successful End Condition | Aktor berhasil menggunakan teknik pomodoro dengan baik | |
| Failed End Condition | Aktor tidak menggunakan teknik pomodoro dengan baik | |
| Primary Actors | User (Mahasiswa) | |
| Secondary Actors | - | |
| Trigger | Aktor memilih/menekan tombol Mulai | |
| Main Flow | Step | Action |
|  | 1. | User memilih/menekan tombol Mulai |
|  | 2. | Timer melakukan waktu mundur selama 25 menit |
|  | 3. | Sistem menampilkan notifikasi untuk beristirahat ketika waktu habis |
|  | 4. | User menekan notifikasi beristirahat |
|  | 5. | Sistem mengubah waktu pada timer menjadi 5 menit |
|  | 6. | User memilih/menekan tombol Mulai |
|  | 7. | Timer melakukan waktu mundur selama 5 menit |
|  | 8. | Sistem menampilkan notifikasi waktu beristirahat sudah habis |
|  | 9. | User menekan notifikasi waktu beristirahat sudah habis |
|  | 10. | Sistem mengembalikan lagi waktu pada timer menjadi 25 menit |
|  | 11. | User mengulangi langkah 1 hingga 10 sampai target tercapai |
| Extension | Step | Branching Action |
|  | 2.1 | Timer tidak berjalan |
|  | 3.1 | Sistem gagal menampilkan notifikasi |
|  | 5.1 | Sistem gagal mengubah waktu pada timer |
|  | 7.1 | Timer tidak berjalan |
|  | 8.1 | Sistem gagal menampilkan notifikasi |
|  | 10.1 | Sistem gagal mengembalikan waktu pada timer |

Tabel 1.9 *Usecase Scenario* Penyediaan Informasi Teknik Pomodoro

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Penyediaan Informasi Teknik Pomodoro | |
| **Related Requirement** | - | |
| Goal In Context | Mendapatkan informasi Teknik Pomodoro | |
| Description | Fungsionalitas ini digunakan oleh user untuk mendapatkan informasi tentang definisi dan cara penggunaan Teknik Pomodoro | |
| Related Use Case | - | |
|  | - | |
| Successful End Condition | Pengguna akan mengerti dan mengetahui Teknik Pomodoro | |
| Failed End Condition | Tidak membaca informasi tentang Teknik Pomodoro | |
| Primary Actors | Mahasiswa | |
| Secondary Actors | - | |
| Trigger | Ketika sudah selesai membaca informasi tentang teknik pomodoro makan akan langsung menuju menu utama aplikasi. | |
| Main Flow | Step | Action |
|  | 1. | Buka Aplikasi |
|  | 2. | Tampil informasi mengenai Teknik Pomodoro |
|  | 3. | Silahkan dibaca, dan swip ke kanan untuk menuju halaman selanjutnya |
| Extension | Step | Branching Action |
|  |  | - |
|  |  |  |

## *Model Activity Diagram*

* **Model Activity Diagram Tambah Data List Kegiatan**

****

Gambar 9 Tambah Data Kegiatan

* **Model Activity Diagram Hapus List Kegiatan**

****

Gambar 10 Hapus Data Kegiatan

* **Model Activity Diagram Ubah List Kegiatan**

****

Gambar 11 Hapus Data Kegiatan

# DAFTAR PUSTAKA

|  |  |
| --- | --- |
| [1] | E. G. Ramadhan, “Javelin Experiment Board,” Jumat November 2018. [Online]. Available: https://medium.com/codelabs-unikom/javelin-experiment-board-487c65c991f3. [Diakses 02 November 2018]. |
| [2] | A. Rahmani, “Cara Cerdas Melakukan Validasi Ide ke Dalam Sebuah Rencana Bisnis,” 02 November 2018. [Online]. Available: https://www.jurnal.id/id/blog/2018/cara-cerdas-memvalidasi-ide-besar-bisnis-ke-dalam-sebuah-rencana. [Diakses 02 November 2018]. |
| [3] | P. Ritonga, “Pengertian Unified Modeling Language ( UML ) dan Modelnya Menurut Pakar dan Ahli,” 02 November 2018. [Online]. Available: https://bangpahmi.com/pengertian-unified-modeling-language-uml-dan-modelnya-menurut-pakar-dan-ahli/. [Diakses 02 November 2018]. |
| [4] | “Ngoding00,” [Online]. Available: https://www.ngodingoo.com/sekilas-tentang-bahasa-pemrograman-java/. |
| [5] | “Codepolitan,” [Online]. Available: https://www.codepolitan.com/belajar-uml-sequence-diagram-57fdb1a5ba777-17044. |
| [6] | “Dictio,” [Online]. Available: https://www.dictio.id/t/apa-yang-dimaksud-dengan-diagram-aktivitas-atau-activity-diagram/15129. |
| [7] | “UMIALFAH,” [Online]. Available: http://fatimahumi.blogspot.com/2014/03/uml-class-diagram.html. |
| [8] | “Materi Dosen,” [Online]. Available: http://www.materidosen.com/2017/04/use-case-diagram-lengkap-studi-kasus.html. |
| [9] | Iosipratama, “Product Design Sprint 101 — Bahasa Indonesia,” 1 Februari 2018. [Online]. Available: https://medium.com/@iosipratama/product-design-sprint-101-bahasa-indonesia-4b2acde80540. |
| [10] | B. Logistik, “Model Bisnis Kanvas Adalah Cara Memetakan Bisnis Lebih Simpel dan Efisien,” MGT Logistik, 29 Agustus 2018. [Online]. Available: https://mgt-logistik.com/model-bisnis-kanvas-adalah/. [Diakses 20 Desember 2018]. |
| [11] | D. Amelia, “Makalah OOAD Dan UML,” Scribd, 1 Januari 2017. [Online]. Available: https://id.scribd.com/document/341568507/Makalah-OOAD-Dan-UML. [Diakses 20 Desember 2018]. |
| [12] | Yasha, “Memahami Google Design Sprint Demi Kesuksesan Bisnis,” Dewaweb, 9 November 2017. [Online]. Available: https://www.dewaweb.com/blog/google-design-sprint/. [Diakses 20 Desember 2018]. |
| [13] | A. R. Pratama, “Belajar Unified Modeling Language (UML) - Pengenalan,” [Online]. Available: https://www.codepolitan.com/unified-modeling-language-uml. |

# LAMPIRAN

Bagian lampiran diisi dengan

* Pembagian tugas

Scrum Master : Ary Sugiarto

Product Owner : Farhan Arif N

Team Development : Feryanto Abi F

Anisa Dewi Umi K

Ahmad Tri Utama

* Rencana iterasi pembangunan perangkat lunak
* Validasi Ide
* Diskusi Hasil Validasi Ide
* Analisis Desain tampilan
* Diskusi Hasil Analisi Desain
* Proses Development (Koding)
* notulen koordinasi team
* Feryanto Abi F
* Resume hasil jawaban wawancara per responden
* Foto kegiatan wawancara

-