SISTEM MONITORING KINERJA PENCAPAIAN OBJECTIVE TIM SCRUM MENGGUNAKAN OBJECTIVE DAN KEY RESULT



Disusun Oleh:

N a m a : Pujiarto Tri Atmaja

NIM : 14523236

PROGRAM STUDI INFORMATIKA – PROGRAM SARJANA

FAKULTAS TEKNOLOGI INDUSTRI

UNIVERSITAS ISLAM INDONESIA

2021

HALAMAN PENGESAHAN DOSEN PEMBIMBING

SISTEM MONITORING KINERJA PENCAPAIAN OBJECTIVE TIM SCRUM MENGGUNAKAN OBJECTIVE DAN KEY RESULT

TUGAS AKHIR

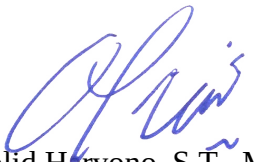
Disusun Oleh:

N a m a : Pujiarto Tri Atmaja

NIM : 14523236

Yogyakarta, 14 Juni 2021

Pembimbing,



(Kholid Haryono, S.T., M.Kom.)

HALAMAN PENGESAHAN DOSEN PENGUJI

SISTEM MONITORING KINERJA PENCAPAIAN OBJECTIVE TIM SCRUM MENGGUNAKAN OBJECTIVE DAN KEY RESULT

TUGAS AKHIR

Telah dipertahankan di depan sidang penguji sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer dari Program Studi Informatika – Program Sarjana di Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, 15 Juli 2021

Tim Penguji

Kholid Haryono, S.T., M.Kom.

Anggota 1

Ari Sujarwo, S.Kom., M.I.T.

Anggota 2

Septia Rani, S.T., M.Cs.

Mengetahui,

Ketua Program Studi Informatika – Program Sarjana

Fakultas Teknologi Industri

Universitas Islam Indonesia

(Dr. Raden Teduh Dirgahayu, S.T., M.Sc.)

HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama : Pujiarto Tri Atmaja

NIM : 14523236

Tugas akhir dengan judul:

SISTEM MONIORING KINERJA PENCAPAIAN OBJECTIVE TIM SCRUM MENGGUNAKAN OBJECTIVE DAN KEY RESULT

Menyatakan bahwa seluruh komponen dan isi dalam tugas akhir ini adalah hasil karya saya sendiri. Apabila di kemudian hari terbukti ada beberapa bagian dari karya ini adalah bukan hasil karya sendiri, tugas akhir yang diajukan sebagai hasil karya sendiri ini siap ditarik kembali dan siap menanggung risiko dan konsekuensi apapun.

Demikian surat pernyataan ini dibuat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 14 Juni 2021



(Pujiarto Tri Atmaja)

HALAMAN PERSEMBAHAN

1. Kedua orang tua dengan segala dukungan yang diberikan sehingga saya dapat menyelesaikan tugas akhir ini
2. Bapak Kholid Haryono, S.T., M.Kom. selaku dosen pembimbing yang dengan sabar membimbing tugas akhir ini hingga selesai.
3. Teman-teman terdekat atas segala dukungan dan bantuannya

HALAMAN MOTO

Sesungguhnya Allah tidak akan mengubah nasib suatu kaum
kecuali kaum itu
sendiri yang mengubah nasibnya"
(Ar Rad : 11)

KATA PENGANTAR

Assalamu'alaikum Warahmatullahi Wabarakatuh

Puji syukur penulis panjatkan ke hadirat Allah SWT yang telah memberikan rahmat dan hidayah dan karunia-Nya, sehingga laporan tugas akhir ini bisa diselesaikan oleh penulis. Tak lupa sholawat serta salam kami haturkan kepada junjungan Nabi Muhammad SAW, yang telah membawa kita dari zaman jahiliyah menuju zaman terang benderang.

Tugas Akhir ini dibuat sebagai salah satu syarat yang harus dipenuhi untuk memperoleh gelar sarjana di Jurusan Informatika Universitas Islam Indonesia. Tugas akhir ini memiliki topik mengenai Sistem Monitoring Kinerja Pencapaian *Objective* Tim *Scrum* Menggunakan *OKR*. Tujuan dari tugas akhir ini adalah membuat sebuah sistem berbasis *web* yang digunakan untuk memonitor kinerja dari tim *Scrum*. Tugas akhir ini dapat terselesaikan atas bantuan, dukungan, dan bimbingan yang diberikan dari berbagai pihak, oleh karna itu penulis ingin menyampaikan rasa terima kasih kepada :

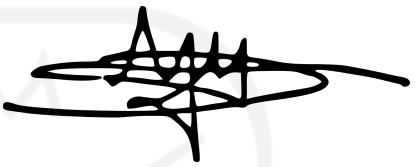
1. Allah SWT yang selalu melimpahkan rahmat, hidayah, dan nikmat Kesehatan hingga penulis dapat menyelesaikan laporan ini.
2. Kedua orang tua dan keluarga penulis atas segala doa dan dukungan hingga penulis dapat menyelesaikan laporan ini.
3. Hendrik, S.T., M.Eng, selaku Ketua Jurusan Informatika Fakultas Teknologi Industri Universitas Islam Indonesia.
4. Dr. Raden Teduh Dirgahayu, S.T., M.Sc. selaku Ketua Program Studi Informatika – Program Sarjana Fakultas Teknologi Industri Universitas Islam Indonesia.
5. Kholid Haryono, S.T., M.Kom., selaku dosen pembimbing yang dengan sabar membimbing penulis sehingga penulis dapat menyelesaikan tugas akhir ini.
6. Teman-teman yang selalu memberikan bantuan, dukungan dan semangat sehingga penulis dapat mengerjakan tugas akhir ini.
7. Semua pihak yang telah banyak membantu penulis yang tidak dapat penulis sebutkan satu persatu.

Penulis menyadari bahwa laporan tugas akhir ini masih belum sempurna, karna keterbatasan kemampuan dan pengalaman penulis. Oleh karna itu penulis mengharapkan kritik dan saran

yang membangun demi kesempurnaan laporan tugas akhir ini. Akhir kata, penulis berharap agar laporan ini dapat bermanfaat bagi semua pihak.

Wassalamu'alaikum Warahmatullahi Wabarakatuh

Yogyakarta, 14 Juni 2021



(Pujiarto Tri Atmaja)

SARI

OKR (*Objective and Key Results*) adalah alat penetapan tujuan kolaboratif yang digunakan oleh tim atau individual untuk menetapkan tujuan yang menantang dan ambisius dengan hasil yang bisa diukur. Zaman ini kita dituntut untuk bekerja secara cepat, tepat dan efisien agar dapat menghemat waktu dan *resource*. Untuk mencapai sebuah *objective* dengan cepat dan tepat maka metode dan praktik tim *Agile Development*, terutama metode *Scrum* banyak digunakan di komunitas pengembang perangkat lunak.

Untuk memantau kinerja semua tim *Scrum*, kita membutuhkan sebuah sistem informasi berbasis *website* yang dapat membantu memonitor kinerja setiap tim *Scrum* berdasarkan *objective* dan *key result* yang telah dicapai, hal ini dilakukan agar semua tim dapat bekerja lebih mudah, dan lebih cepat. *Website* yang dibuat diharapkan mampu memberikan informasi yang cukup terhadap pengguna.

Sistem informasi berbasis *website* ini dibuat menggunakan *framework Laravel* dengan *sintax PHP* dan *HTML*. Dengan bantuan *software Docker* sebagai *Container, MySql* sebagai *database* dan *Visual Studio Code* sebagai *code* editor teksnya. Setelah melalui proses pengembangan, pengujian serta revisi-revisi yang sudah dilakukan, maka tersusunlah sebuah sistem *monitoring* berbasis *website* yang dapat dimanfaatkan secara baik.

Kata kunci: efisien, *Scrum*, website, monitor, *okr*

GLOSARIUM

Website	Halaman informasi yang diakses menggunakan internet.
Framework	Kerangka untuk mengembangkan sesuatu.
Interface	Tampilan visual sebuah produk.
Database	Kumpulan data yang disimpan dan dikelola.
Scrum	Sebuah kerangka yang membantu tim untuk bekerja sama.
OKR	*Objective* dan *Key Result*.
Objective	Tujuan yang ingin dicapai.
Key Result	Indikator yang menggambarkan keberhasilan dalam proses mewujudkan *objective*.
Initiative	Semua *task* yang dapat membantu mencapai *Key Result*.

DAFTAR ISI

HALAMAN JUDUL i
HALAMAN PENGESAHAN DOSEN PEMBIMBING ii
HALAMAN PENGESAHAN DOSEN PENGUJI iii
HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR iv
HALAMAN PERSEMBAHAN v
HALAMAN MOTO vi
KATA PENGANTAR vii
SARI. ix
GLOSARIUM x
DAFTAR ISI xi
DAFTAR TABEL xiii
DAFTAR GAMBAR xiv
BAB I PENDAHULUAN 15
1.1 Latar Belakang Masalah 15
1.2 Rumusan Masalah 2
1.3 Batasan Masalah 2
1.4 Tujuan Penelitian 2
1.5 Manfaat Penelitian 2
1.6 Sistematika Penulisan 3
BAB II LANDASAN TEORI 4
2.1 Objective and Key Result 4
2.1.1 Monitoring and Tracking OKR 6
2.2 Prototype 6
2.3 Unified Modelling Language (UML) 7
2.3.1 *Use Case Diagram* 8
2.3.2 *Activity Diagram* 9
2.4 Black Box Testing 10
2.4.1 *Functional Testing* 10
2.5 Skala Likert 12
BAB III METODOLOGI PENELITIAN 14
3.1 Planing 14
3.1.1 *Initial Requirement* 14

3.2 Desain 18
3.2.1 *Use Case Diagram* 18
3.2.2 *Activity Diagram* 19
3.2.3 *Database Ralation Design* 26
3.3 Prototyping 30
BAB IV IMPLEMENTASI DAN PENGUJIAN 36
4.1 Implementasi 36
4.1.1 Skenario 36
4.1.2 Implementasi Sistem 37
4.2 Pengujian 41
4.2.1 *Functional Testing* 41
4.2.2 Pengujian Usabilitas 42
BAB V KESIMPULAN DAN SARAN 43
5.1 Kesimpulan 43
5.2 Saran 43
DAFTAR PUSTAKA 44
LAMPIRAN 45

DAFTAR TABEL

Tabel 2.1 Komponen Use Case....8
Tabel 2.2 Komponen Activity Diagram....10
Tabel 2.3 *Skala Likert*....12
Tabel 3.1 *User Requirement*....14
Tabel 3.2 Tabel User....26
Tabel 3.3 Tabel Team....27
Tabel 3.4 Tabel Objective....27
Tabel 3.5 Tabel Keyresult....28
Tabel 3.6 Tabel Task....28
Tabel 3.7 Tabel Deadline....29
Tabel 4.4.1 Skenario Penggunaan Sistem....36
Tabel 4.2 *Funtional Testing*....41
Tabel 4.3 Hasil Kuesioner Pengujian Sistem....42

DAFTAR GAMBAR

Gambar 2.1 Contoh Program *Weekdone* 6
Gambar 2.2 Tahapan Model *Prototype* 7
Gambar 2.3 Contoh Activity Diagram 9
Gambar 2.4 Cara Melakukan *Functional Testing* 11
Gambar 3.1 Diagram Proses Bisnis 17
Gambar 3.2 *Use Case Diagram* 18
Gambar 3.3 *Activity Diagram Login* 19
Gambar 3.4 *Activity Diagram* Membuat Tim Baru 20
Gambar 3.5 *Activity Diagram* Membuat *Objective* Baru 21
Gambar 3.6 *Activity Diagram* Membuat *Key Result* atau *Initiative* Baru 22
Gambar 3.7 *Activity Diagram* Mengubah *Objective* 23
Gambar 3.8 *Activity Diagram* Mengubah *Key Result* atau *Initiative* 24
Gambar 3.9 *Activity Diagram* Mengupdate *Progress Key Result* atau *Initiative* 25
Gambar 3.10 Desain Relasi Database 26
Gambar 3.11 Desain Halaman *Login* 30
Gambar 3.12 Desain Halaman Team List 30
Gambar 3.13 Desain Halaman *Add Team* 31
Gambar 3.14 Desain Halaman *Monitoring* 32
Gambar 3.15 Desain Halaman *New Objective* 32
Gambar 3.16 Desain Halaman *Objective Details.* 33
Gambar 3.17 Desain Halaman *Edit Objective* 33
Gambar 3.18 Desain Halaman *New Key Result* 34
Gambar 3.19 Desain Halaman *New Initiative* 34
Gambar 3.20 Desain Halaman *Update Key Result* atau *Initiative* 35
Gambar 4.1 *Form* Pembuatan Tim Baru 37
Gambar 4.2 Implementasi Tim List 38
Gambar 4.3 Form Pembuatan *Objective* 38
Gambar 4.4 *Form* Pembuatan *Key Result.* 39
Gambar 4.5 *Form* Pembuatan *Intitiatve* 39
Gambar 4.6 Halaman Update Progress *Key Result* 40
Gambar 4.7 Halaman *Update Initiative* 40
Gambar 4.8 Halaman *Monitoring* Tim 41

BAB I
PENDAHULUAN

1.1 Latar Belakang Masalah

Pada abad ini kita dituntut untuk bekerja secara cepat, tepat dan efisien agar dapat menghemat waktu. Untuk mencapai sebuah *objective* dengan cepat dan tepat maka metode dan praktik tim *Agile Development* banyak digunakan di komunitas pengembang perangkat lunak (Mahnic & Zabkar, 2012).

Menurut hasil survei yang dilakukan metode *Agile* yang banyak digunakan yaitu metode *Scrum* (Mahnic & Zabkar, 2012). *Scrum* adalah sebuah *framework* yang di dalamnya kita bisa menyelesaikan masalah adaptif yang kompleks selagi menghasilkan produk dengan nilai yang tinggi secara produktif dan kreatif (Ken Schwaber & Sutherland, 2017).

Scrum digunakan untuk memberikan perangkat lunak kepada *end user* dengan cepat, dan lebih baik (K. Schwaber & Sutherland, 2011), akan tetapi salah satu kekhawatiran terbesar pada metode *Scrum* ini adalah kehilangan *management control* yang dapat mengakibatkan terlambat atau bahkan gagalnya suatu proyek (Mahnic & Zabkar, 2012). Tim *Scrum* lazimnya memiliki OKR (*Objective and Key Results*) yang digunakan sebagai tujuan utama sebuah proyek.

OKR (*Objective and Key Results*) adalah alat penetapan tujuan kolaboratif yang digunakan oleh tim atau individual untuk menetapkan tujuan yang menantang dan ambisius dengan hasil yang bisa diukur (Ryan Panchadsaram & Sam Prince). OKR terdiri dari *Objective*, dan *Key Result*. *Objective* adalah sebuah pernyataan tujuan yang umum atau luas yang biasanya bersifat kualitatif, *Key Result* adalah sebuah *statement* terukur yang mendemonstrasikan pencapaian sebuah *objective* (Niven & Lamorte, 2016). Keuntungan menggunakan OKR adalah bertambah baiknya komunikasi, fokus, dan mendorong hasil bisnis yang lebih baik karena OKR mendefinisikan tujuan dengan jelas, memperjelas apa yang penting di seluruh tim, dan memfokuskan upaya pada *Key Result* yang memberikan atau menyelesaikan *Objective*.

BSI (Badan Sistem Informasi) Universitas Islam Indonesia menggunakan metode OKR untuk membuat target atau tujuan yang ingin dicapai dalam kurun waktu yang sudah ditentukan. Pada saat ini BSI masih menggunakan *Excel spreadsheet* untuk mengelola dan memonitor pencapaian OKR yang dimiliki oleh masing-masing tim *Scrum*. BSI memonitor OKR dengan cara melakukan rapat atau *meeting* per bulan dan per triwulan yang membahas

tentang tujuan yang telah di capai, masalah-masalah yang ditemukan pada saat pengembangan dan bagaimana cara mengatasi atau menghindari masalah-masalah tersebut.

Karena BSI diharapkan akan terus berkembang maka tentu saja jumlah tim *Scrum* yang ada di BSI juga akan ikut berkembang, perkembangan ini akan mempersulit pengelolaan *Excel spreadsheet* yang digunakan untuk mengelola dan memonitor pencapaian OKR karena banyaknya jumlah tim *Scrum* dan OKR milik masing-masing tim *Scrum* tersebut, oleh karena itu penulis mengajukan tugas akhir pembuatan sistem *monitoring* dengan judul "Sistem Monitoring Kinerja Pencapaian Objectiv*e* Tim Scrum Menggunakan Objective dan Key Result". Sistem ini juga diharapkan untuk menghindari terlambat atau gagalnya suatu proyek karena sistem ini mampu memantau kinerja tim *Scrum* berdasarkan pencapaian pada *Objective and Key Result (OKR)* yang sudah di tentukan oleh setiap tim *Scrum*

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, rumusan masalah dari penelitian ini adalah bagaimana cara untuk meningkatkan aktivitas dan produktivitas *objective* dan *key result* pada tim *Scrum.*

1.3 Batasan Masalah

Dalam pengerjaan tugas akhir ini terdapat beberapa batasan masalah agar pengerjaan lebih terarah:

1. Sistem yang dibangun berbasis *website.*
2. Sistem hanya memonitor tim berdasarkan pencapaian *OKR.*

1.4 Tujuan Penelitian

Adapun tujuan yang ingin dicapai dari penelitian ini adalah pembuatan sebuah sistem yang dapat memonitor kinerja pencapaian tim *Scrum* berdasarkan *objective* yang dimiliki oleh setiap tim *Scrum.*

1.5 Manfaat Penelitian

Sistem ini diharapkan untuk membantu *product owner* untuk memantau kinerja dari tim *Scrum* berdasarkan pada *OKR* yang telah dicapai

1.6 Sistematika Penulisan

Sistematika penulisan laporan digunakan untuk mempermudah pembaca dalam memahami laporan Tugas Akhir. Berikut garis besar sistematika penulisan laporan :

BAB I PENDAHULUAN

Bagian ini mengandung latar belakang masalah, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, dan sistematika penulisan.

BAB II LANDASAN TOERI

Bagian ini berisi penjelasan teori-teori yang digunakan sebagai landasan untuk menyelesaikan permasalahan yang dibahas pada penelitian ini.

BAB III METODOLOGI PENELITIAN

Bagian ini berisi tentang penjelasan dari metode yang digunakan penulis untuk melakukan penelitian.

BAB IV IMPLEMENTASI DAN PENGUJIAN

Bagian ini berisi tentang hasil dari implementasi penelitian berdasarkan perancangan yang telah dibuat dan pengujian hasil dari penelitian.

BAB V KESIMPULAN DAN SARAN

Bagian ini berisi tentang kesimpulan yang didapat dari penelitian tugas akhir beserta saran yang diperlukan agar hasil yang dicapai ke depannya menjadi lebih baik.

BAB II
LANDASAN TEORI

2.1 Objective and Key Result

OKRs (Objective and Key Result) adalah *framework* berpikir kritis dan disiplin berkelanjutan yang berusaha membuat pekerja atau karyawan bekerja sama, memfokuskan upaya untuk memberikan kontribusi yang terukur untuk mendorong organisasi menjadi lebih maju (Niven & Lamorte, 2016). *OKRs* terdiri dari:

a. *Objective*

   Objective adalah sebuah pernyataan tujuan yang luas biasanya bersifat kualitatif.

b. *Key Result.*

   Key Result adalah sebuah pernyataan terukur yang dapat menunjukkan pencapaian sebuah *Objective.*

Keuntungan menggunakan *OKR*s yaitu *OKR*s dapat meningkatkan komunikasi, fokus dan keselarasan untuk mendorong hasil bisnis yang lebih baik, ada 10 karakteristik yang menunjukan bahwa *OKRs* efektif

a. *Results Not Task*

   Key Results adalah hasil, bukan *tasks*, karena *tasks* muncul di dalam *"to-do list"* dan biasanya dapat diselesaikan dengan cepat, sedangkan *Key Results* mencerminkan kemajuan atau perkembangan yang terukur dengan target tanggal dan tidak bisa diselesaikan dalam satu hari.

b. *SMART*

   Key Results yang efektif itu *Specific, Measurable, Achivable, Relevant, and Time-bound (SMART). Key Results* yang tidak dapat diukur itu tak berarti.

c. *FOCUSED*

   Setiap set *OKRs* seharusnya mempunyai paling banyak 5 *Objectives* yang fokus di 1-3 *Key Results* pada setiap *Objective.*

d. *OWNED*

   Objective berasal dari diri sendiri, pemilik *OKRs*, bukan mandat perusahaan, kita harus mempertimbangkan *Objective* yang berada dalam kendali kita atau organisasi kita.

e. *ALIGNED FOR SUCCESS*

OKRs yang efektif mengukur keberhasilan. meskipun menyusun *OKRs* mencerminkan definisi sukses milik *owner, OKRs* juga harus disempurnakan berdasarkan masukan dari anggota tim yang lain untuk menjamin dan memastikan keselarasan dan kesepakatan tentang apa yang dianggap sukses.

f. *FREQUENTLY UPDATED*

Saat mengatur atau membuat *OKRs* personal untuk kuartal ke depan, pertimbangkan seberapa sering anda berencana untuk memperbarui *progress* pada setiap *Key Results*.

g. *CLEAR*

Komunikasi yang efektif bisa menjadi salah satu manfaat utama menggunakan *OKRs. OKRs* yang efektif ditulis dengan jelas, ringkas dan hanya mempunyai akronim standar yang mudah dimengerti

h. POSITIVE

Membuat sebuah metrik di mana *"Bigger is better"* jika dimungkinkan. *OKRs* Biasanya melaporkan diri sendiri (*self-reported*), dan kita ingin semua orang di dalam tim melihat ke depan untuk membuat *progress* yang positif daripada mencatat dan mengingat peristiwa buruk.

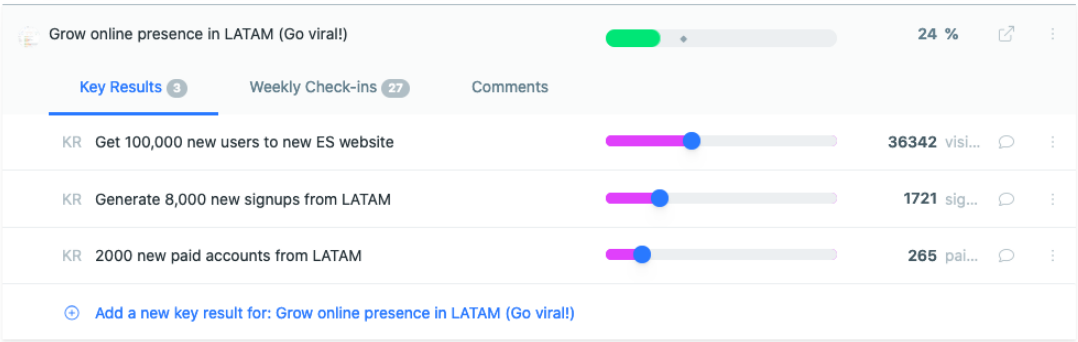
i. GRADED

Key Results yang efektif mencakup nilai yang dapat mengkomunikasikan tujuan dengan jelas dan dapat mengelola harapan.

j. ASPIRATIONAL

Menurut penelitian tentang *goal science,* pekerja dapat mencapai tujuan yang lebih tinggi saat dituntut untuk mencapai tujuan yang tinggi.

Salah satu contoh program yang banyak digunakan untuk memonitor *OKRs* adalah *Weekdone. Weekdone* adalah program pelaporan mingguan dan OKR yang digunakan untuk memanajemeni *team*, penetapan tujuan dan meningkatkan komunikasi *team*. Tampilan dari program *Weekdone* dapat dilihat pada Gambar 2.1 Contoh Program *Weekdone.*



Gambar 2.1 Contoh Program *Weekdone*

Sumber : https://weekdone.com/

2.1.1 Monitoring and Tracking OKR

Untuk memonitor *progress* OKR, diperlukan *Team Initiative*, OKR *check-ins* dan *Weekly, Monthly, and Quarterly retrospective reviews.*

Team Initiative adalah aktivitas (*action plans*) yang dikerjakan oleh tim untuk menghasilkan hasil yang sudah ditentukan dalam *Key Result.* Saat mendiskusikan *initiatives*, tim harus mempertimbangkan apakah mereka memerlukan bantuan dari fungsi lain dan memprioritaskan rencana mereka berdasarkan sumber daya yang tersedia.

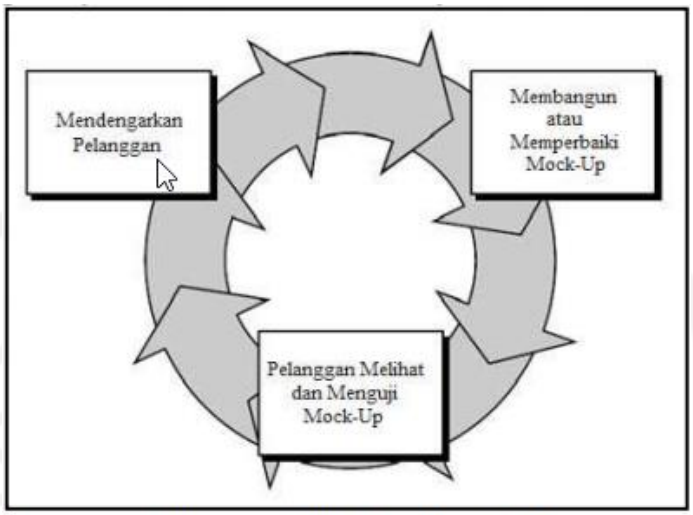
Weekly OKR *Chek-ins* adalah pertemuan rutin yang dihadiri oleh orang-orang yang mengerjakan OKR yang sama. Terlepas dari *progress update* rutin dan berusaha untuk tetap singkron antara satu dengan yang lainnya, inti dari pertemuan ini adalah untuk memastikan pertukaran informasi secara teratur yang akan meningkatkan pengertian satu dengan yang lain dan membangun kepercayaan.

Meninjau *progress* dalam rapat *retrospective* yang dilakukan *weekly*, *monthly*, dan *quarterly* dapat membantu menyesuaikan fokus area untuk siklus target atau tujuan berikutnya.

2.2 Prototype

Prototype adalah model awal dari produk yang akan dibangun untuk menguji desainnya. Kata *Prototype* berasal dari Bahasa Yunani yaitu *"primitive form"*. *Prototype* banyak digunakan di desain dan *engineering* untuk menyempurnakan produk dan proses pembuatan sebelum produk tersebut di implementasi (Amy et al., 2019). Dalam pengembangan perangkat lunak, *Prototyping* merupakan salah satu metode yang digunakan untuk mengembangkan perangkat lunak yang banyak digunakan. Dengan metode *prototyping* ini pengembang dan *user* dapat saling berkomunikasi dan memberikan informasi selama proses pembuatan produk.

Model *prototype* memiliki beberapa tahapan. Tahapan model *prototype* dapat dilihat pada Gambar 2.1 Tahapan Model *Prototype*



Sumber: (Sukamto & Shalahuddin, 2015:32)

Gambar 2.2 Tahapan Model *Prototype*

a. Mendengarkan Pelanggan

   Pengembang sistem dan objek penelitian atau *user* bertemu dan menentukan tujuan umum dan kebutuhan dasar.

b. Membangun atau Memperbaiki *Mock-Up*

   Perancangan sistem dapat dikerjakan apabila semua data dan informasi yang berkaitan telah dikumpulkan. Rancangan ini menjadi dasar pembuatan *prototype.*

c. Pelanggan Melihat dan Menguji *Mock-Up*

   Objek penelitian atau *user* mengevaluasi *prototype* yang telah dibuat dan pergunakan untuk memperjelas kebutuhan sistem.

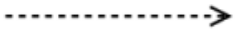
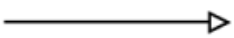
2.3 Unified Modelling Language (UML)

Unified Modelling Language (UML) adalah sebuah Bahasa pemodelan di bidang pengembangan perangkat lunak yang dimaksudkan untuk menyediakan cara standar untuk memvisualisasikan desain dari sebuah sistem (Wesley, 2015)

2.3.1 *Use Case Diagram*

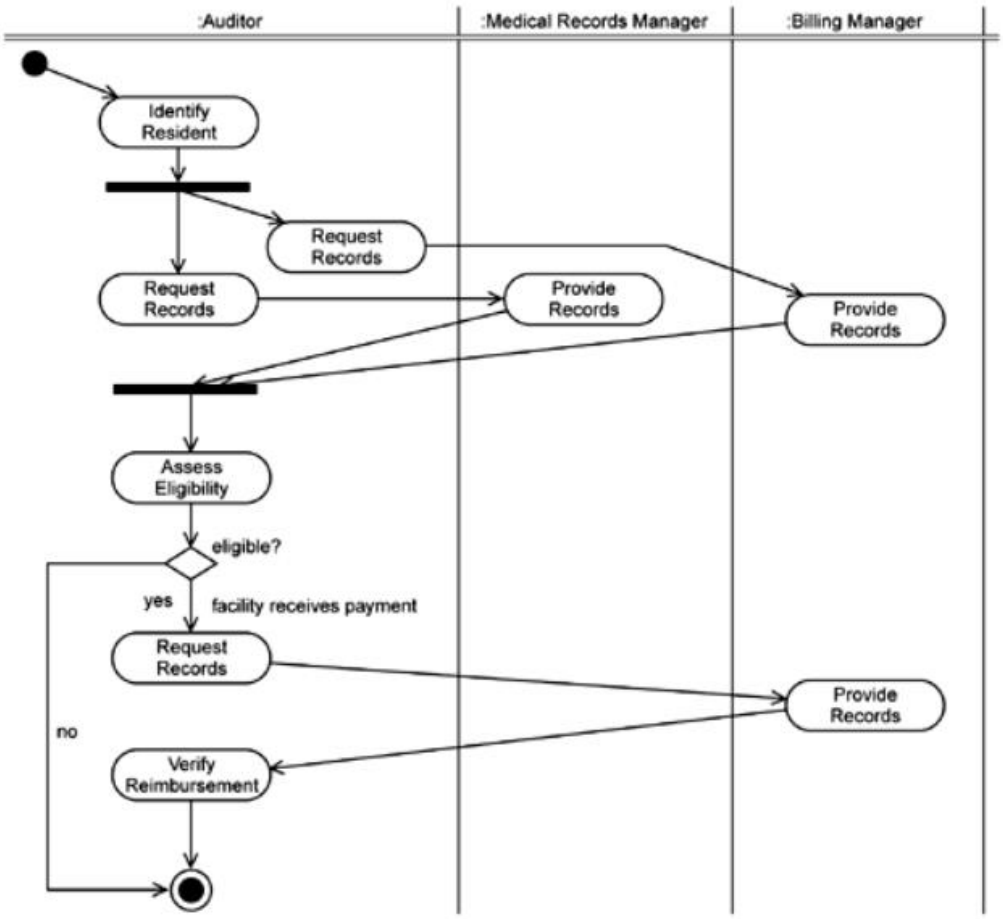
Use Case Diagram adalah sebuah model dari fungsi yang diharapkan pada sistem dan lingkungan yang mendukung proses bisnisnya. Model ini berfungsi sebagai kontrak antara pelanggan atau *user* dan para pengembang (*Eric J*, n.d.). *Use Case Diagram* memiliki beberapa komponen yang dapat dilihat pada Tabel 2.1 Komponen Use Case Diagram.

Tabel 2.1 Komponen Use Case

Simbol	Deskripsi
Actor 	*Actor* adalah segala hal ada di luar sistem (orang ataupun sistem lain) yang akan menggunakan atau berinteraksi dengan sistem.
Use Case 	Use case merupakan daftar aksi atau tahapan lengkap yang mendefinisikan interaksi antara aktor dan sistem untuk mencapai suatu tujuan
Association 	Sebuah koneksi yang menunjukkan komunikasi atau interaksi antara aktor dan *use case*
Dependency 	*Dependency* terbagi menjadi 2 macam yaitu *include* yang berfungsi untuk mengidentifikasi hubungan antara 2 *use case* di mana *use case* yang satu akan memanggil *use case* lainnya, dan *extend* apabila pemanggilan memerlukan kondisi tertentu maka akan berlaku dependensi
Generalization 	*Generalization* mendefinisikan relasi antara dua aktor atau dua *use case* yang mana salah satunya mewarisi dan menambahkan atau mengesampingkan sifat dari yang lainnya

2.3.2 *Activity Diagram*

Activity diagram adalah diagram yang menunjukkan bagaimana sebuah proses atau aktivitas berjalan, diagram ini juga bisa digunakan untuk menunjukkan proses bisnis tingkat tinggi dan bagaimana bisnis tersebut bekerja (*Esric J,* n.d.). Contoh dari *Activity Diagram* dapat dilihat pada Gambar 2.2 Contoh *Activity Diagram.*



Gambar 2.3 Contoh Activity Diagram

Activity Diagram memiliki beberapa komponen, komponen dapat dilihat pada tabel 2.2 Tabel komponen *Activity Diagram*

Tabel 2.2 Komponen Activity Diagram

Simbol	Deskripsi
Initial Node 	Menggambarkan awal dari serangkaian kegiatan atau Tindakan,
Activity 	Digunakan untuk menggambarkan serangkaian aktivitas dan biasanya diawali dengan kata kerja.
Decision 	Percabangan di mana ada lebih dari satu pilihan aktivitas.
Join	Penggabungan 2 atau lebih aktivitas digabungkan menjadi satu.
Activity final node 	Status akhir dari sebuah diagram aktivitas .
Swimlane 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang dilakukan.

2.4 Black Box Testing

Black Box Testing adalah sebuah metode pengujian perangkat lunak di mana fungsionalitas dari aplikasi perangkat lunak tersebut diuji tanpa mengetahui kode internal dari aplikasi perangkat lunak tersebut (Jerry Zeyu Gao, H.-S. Jacob Tsao, 2003)

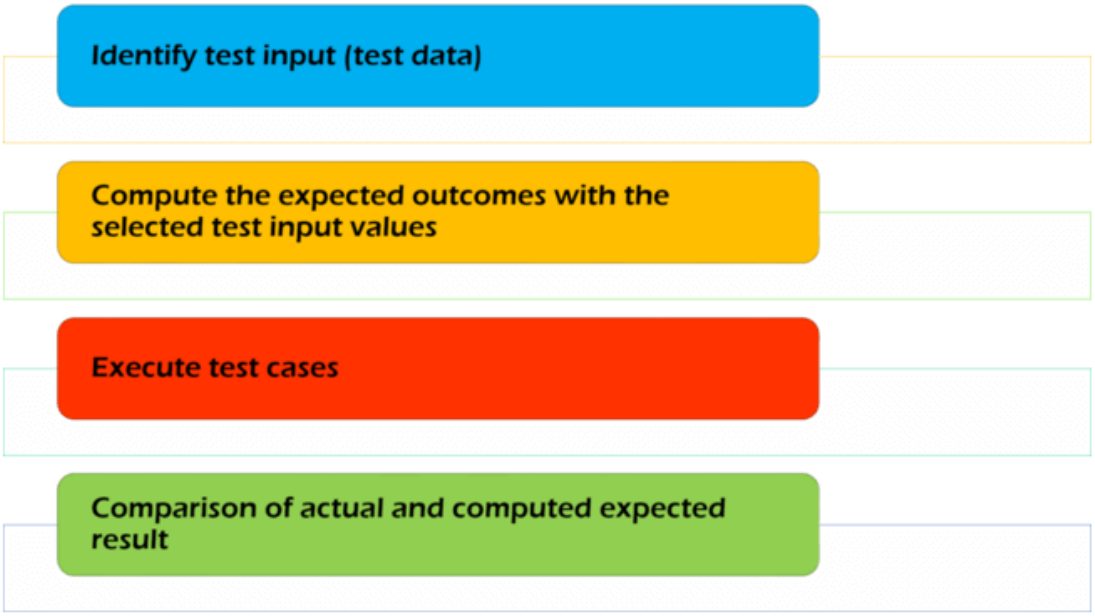
2.4.1 *Functional Testing*

Functional Testing merupakan tipe pengujian perangkat lunak yang memvalidasi sistem perangkat lunak terhadap persyaratan atau spesifikasi fungsi sistem perangkat lunak. Tujuan dari *Funtional Testing* ini untuk menguji setiap fungsi yang ada di dalam sistem dengan

memberikan *input* yang tepat, lalu memverifikasi *output* dan membandingkannya terhadap spesifikasi atau persyaratan dari fungsi perangkat lunak.

Langkah-langkah yang dilakukan untuk melakukan *functional testing* adalah sebagai berikut:

a. *Indentify test input*,
   mengidentifikasi data atau masukan yang akan digunakan untuk menguji fungsionalitas sistem.
b. *Compute the expected outcomes*
   Menghitung atau memprediksi hasil yang diharapkan dari data atau masukan yang digunakan untuk menguji sistem tadi.
c. *Execute test case*
   Menjalankan atau mengeksekusi data atau masukan yang digunakan untuk menguji sistem ke dalam sistem.
d. *Comparison of actual and computed expected result*
   Membandingkan hasil yang dari data yang dikelola oleh sistem dengan data yang diprediksi atau dihitung, apakah data sesuai atau tidak.



Gambar 2.4 Cara Melakukan *Functional Testing*

Sumber : https://www.guru99.com/functional-testing.html

2.5 Skala Likert

Skala Likert merupakan suatu metode penghitungan terhadap penelitian, sikap, pendapat dan persepsi seseorang dalam sebuah variabel penelitian. Variabel yang diukur menjadi sebuah indikator yang kemudian dijadikan sebagai titik tolak untuk menyusun item instrumen berupa pernyataan atau pertanyaan(Sugiyono, 2008).

Jawaban dari *Skala Likert* mempunyai gradasi dari sangat positif sampai sangat negatif, setiap jawaban tersebut dapat diberikan skor, misalnya Sangat Setuju (SS) dengan skor 5, Setuju (S) dengan skor 4, Ragu-ragu (R) dengan skor 3, Tidak Setuju (TS) dengan skor 2, Sangat Tidak Setuju (TST) dengan skor 1(Sugiyono, 2008). Contoh kuesioner sebuah *Skala Likert* dapat dilihat pada Tabel 2.3 *Skala Likert*

Tabel 2.3 *Skala Likert*

No	Pernyataan	Jawaban				
		SS	S	R	TS	STS
1	UII akan tetap melakukan kuliah daring selama masa pandemi	50	20	20	5	5
2	……					

Selanjutnya kuesioner akan diberikan kepada 100 mahasiswa dengan hasil 50 mahasiswa menjawab SS, 20 mahasiswa menjawab S, 20 mahasiswa menjawab R, 5 mahasiswa menjawab TS, dan 5 mahasiswa menjawab STS. Untuk dapat menghitung rata-rata pada jawaban responden maka perlu mengetahui jumlah skor dari masing-masing pertanyaan dengan perhitungan sebagai berikut :

Jumlah Skor SS $= 50 \text{ x } 5 = 250$
Jumlah Skor S $= 20 \text{ x } 4 = 80$
Jumlah Skor R $= 20 \text{ x } 3 = 60$
Jumlah Skor TS $= 5 \text{ x } 2 = 10$
Jumlah Skor STS $= 5 \text{ x } 1 = 5$

Total $= 405$ (2.1)

Jika seluruh responden menjawab SS maka skor maksimal akan bernilai 5 x 100 = 500, sedangkan total skor yang diperoleh adalah 405. Berdasarkan pada perhitungan dari data tersebut maka tingkat persetujuan responden terhadap pertanyaan dari kuesioner di atas dapat di hitung dengan cara:

$$Persentase\ Skor = \frac{Total\ Skor}{Skor\ Maksimal}\ x\ 100\%$$

$$\frac{405}{500}\ x\ 100\% = 81\% \quad (2.2)$$

Jadi tingkat persetujuan responden terhadap pertanyaan dari kuesioner adalah 81%

BAB III
METODOLOGI PENELITIAN

3.1 Planing

Dalam mengerjakan tugas akhir ini Adapun tahapan pengerjaan penulis awali dari menentukan *Initial Requirement* untuk mengetahui kebutuhan sistem berdasarkan pengguna atau *user*. Setelah penulis mengetahui semua kebutuhan dari pengguna atau *user*, tahapan selanjutnya adalah untuk mengetahui proses bisnis yang akan dikembangkan. Jika proses bisnis sudah terpenuhi, dilanjutkan dengan membuat desain *interface* dan *database*.

3.1.1 *Initial Requirement*

a. *User Requirement*

Untuk mengetahui kebutuhan dari *user* yang akan menggunakan sistem, diperlukan *user requirement* yang di dapatkan dari hasil diskusi dengan *user* dan Dosen Pembimbing Tugas akhir. Tabel *user requirement* dapat dilihat pada Tabel 3.1 *User Requirement*.

Tabel 3.1 *User Requirement*

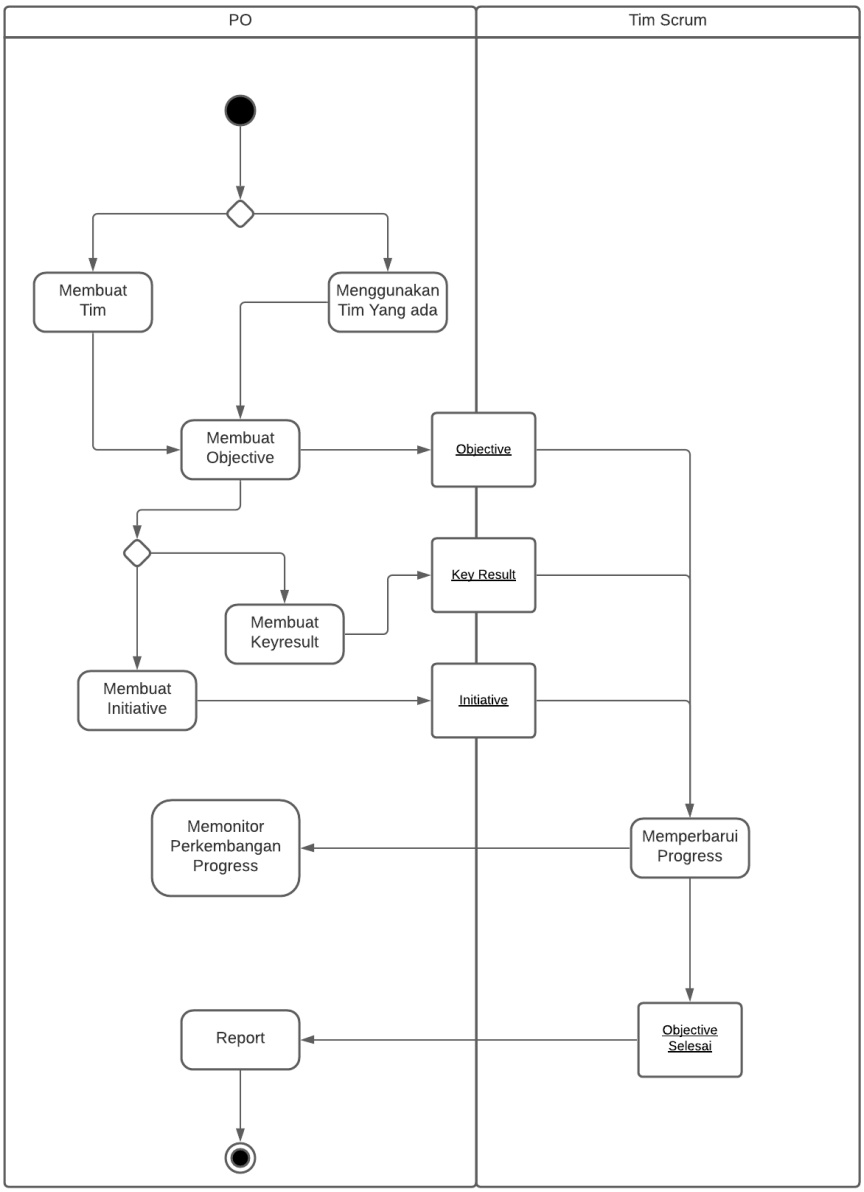
No	Kebutuhan	Skenario
1	Sebagai *user* saya ingin bisa mendaftarkan diri ke dalam sistem agar saya dapat *login*	- Memasukkan nama, *email* dan *password* - Menggunakan tombol *Register* untuk mendaftarkan diri ke dalam sistem
2	Sebagai *user* saya ingin bisa melakukan *login* agar saya bisa masuk ke dalam sistem menggunakan data yang sudah saya daftarkan sebelumnya	- Memasukkan email dan *password* di halaman *login* - Menggunakan tombol *Login*
3	Sebagai *user* saya ingin mempunyai halaman utama untuk menampilkan menu utama yang ada di dalam sistem	-Setelah user melakukan *login*, user akan di arahkan ke halaman *dashboard* -Halaman *dashboard* menampilkan menu utama yang ada di dalam sistem
4	Sebagai *Product Owner* saya ingin bisa melihat semua tim *Scrum* yang saya miliki	-Pada halaman *dashboard* terdapat menu tim *list* yang dapat menampilkan semua tim *Scrum* yang ada -Menggunakan tombol tim *list* untuk menuju ke halaman tim *list*

5	Sebagai *Product Owner* saya ingin membuat sebuah tim baru untuk mengerjakan proyek baru	-Pada halaman tim list, *Product owner* dapat membuat sebuah tim baru. -Menggunakan tombol *New team* untuk membuat sebuah tim baru akan diarahkan ke halaman pembuatan tim -Memasukkan nama tim yang diinginkan. -Menggunakan tombol *save* untuk menyimpan tim yang sudah dibuat.
6	Sebagai *Product Owner* saya ingin melihat dan memantau bagaimana perkembangan progress dari tim *Scrum*	-Untuk melakukan *monitoring* sistem mempunyai halaman monitor yang menampilkan *objectives key result dan initiative* setiap tim -Pada halaman tim *list*, nama dari tim bisa digunakan untuk menuju halaman monitor dari tim tersebut.
7	Sebagai *Product Owner* saya ingin bisa menambahkan sebuah objective baru	-Pada halaman *monitoring*, terdapat tombol untuk menambahkan *objective* baru -Menggunakan tombol *New Objective* untuk masuk ke halaman pembuatan *objective* baru. -pada halaman *New Objective*, *Product owner* memasukkan nama *objective* baru, memasukkan deskripsi dari *objective* tersebut, dan menentukan kapan *objective* tersebut harus dikerjakan dan diselesaikan. -Menggunakan tombol *save* untuk menyimpan *objective* yang sudah dibuat.
8	Sebagai *Product Owner* saya ingin melihat objective lebih detail seperti deskripsi dari objective tersebut.	-Pada halaman *monitoring*, nama dari *objective* yang ada bisa digunakan untuk menuju ke halaman *objective details.*
9	Sebagai *Product Owner* saya ingin bisa mengubah atau menghapus nama atau tanggal pengerjaan dari sebuah *Objective, key result,* maupun *inisiative,*	-Pada halaman *objective details,* terdapat tombol *Edit*. -Menggunakan tombol *Edit.* untuk menuju ke halaman edit. -Memasukkan nama baru untuk *Objective, key result* atau *Inisiative*. Menentukan tanggal mulai pengerjaan dan kapan *OKRs* harus sudah diselesaikan. -Menggunakan tombol *save* untuk menyimpan perubahan yang sudah dilakukan. -Untuk menghapus sebuah *objective, Key result* atau *inisiative,*

		menggunakan tombol *Delete* yang ada di dalam halaman *Objective Details*
10	Sebagai *Product Owner* saya ingin bisa melihat laporan mengenai kinerja tim *Scrum* berdasarkan *OKRs* yang telah dicapai	-Pada halaman Tim *List* terdapat tombol *details* di setiap tim yang ada. -Menggunakan tombol detail untuk menuju ke halaman *Team Details.* -Halaman *team details* akan menampilkan grafik kinerja tim.
11	Sebagai tim *Scrum* saya ingin mengupdate progress atau dari suatu *objective, Key result,* atau *Insiative*	-Pada halaman *objective details* terdapat tombol *update.* -Menggunakan tombol *update* untuk menuju ke halaman update *progress* -Menggunakan *slider* yang disediakan untuk *mengupdate* sejauh mana perkembangan *progress* dari *objective, key result,* atau *inisiative* yang sedang dikerjakan

b. Proses bisnis

Untuk menggambarkan Langkah-langkah dan logika dari proses bisnis pada sistem *monitoring* ini menggunakan *Unified Modelling Language* (UML). Diagram dari proses bisnis dapat dilihat pada Gambar 3.1 Diagram Proses Bisnis.



Gambar 3.1 Diagram Proses Bisnis

Pada diagram proses bisnis ini terdapat 1 *start event*, 1 *end event*¸ 1 *decision*, 8 activity, dan 3 *object.* Proses dimulai saat *Product owner* membuat atau memilih sebuah tim untuk memulai sebuah proyek. Selanjutnya *Product owner* membuat *Objectives* yang digunakan

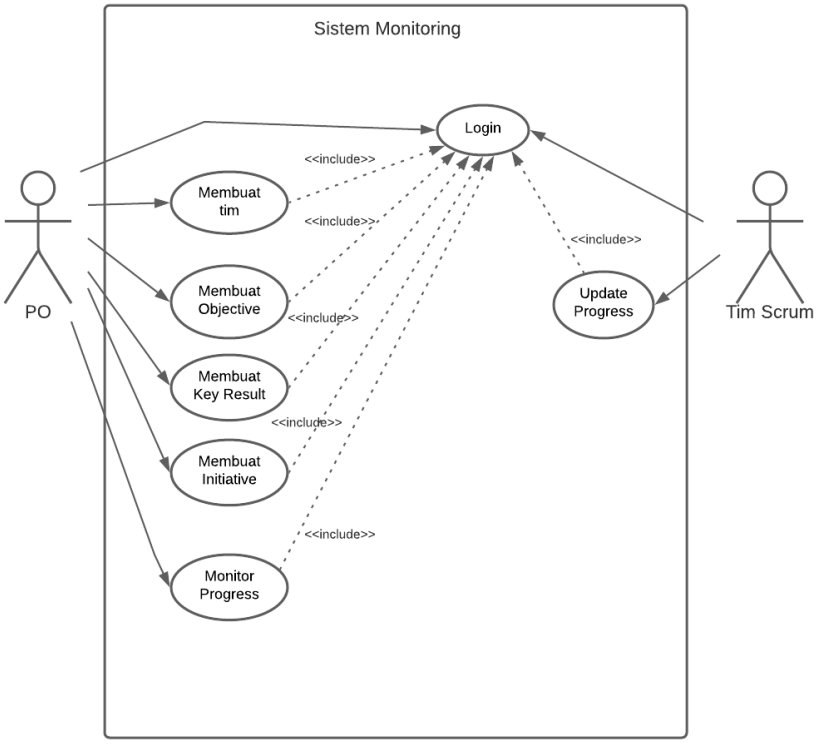
untuk menyelesaikan proyek yang dibuat, kemudian *Product owner* menambahkan *Key Result* atau *Initiative* untuk dapat mencapai *Objectives* yang telah dibuat. Tim *Scrum* mengerjakan *Key Result* dan *Initiative* yang sudah dibuat dan ditentukan oleh *Product owner* dan terus melakukan *update* sejauh mana *progress* yang telah dicapai. *Product owner* dapat memantau perkembangan *progress* berdasarkan *OKRs* yang telah dicapai oleh Tim *Scrum*. Setelah Tim *Scrum* sudah memenuhi semua *Objectives* yang dibutuhkan, *Tim Scrum* dapat memasukkan *report* untuk menjadi bukti bahwa *OKRs* sudah diselesaikan.

3.2 Desain

Untuk membuat desain dari sistem yang akan dibuat, dibutuhkan *use case diagram* dan *activity diagram* untuk mengetahui bagaimana sistem akan digunakan nantinya.

3.2.1 *Use Case Diagram*

Untuk dapat mengetahui fungsi dan interaksi apa saja yang ada pada sistem *monitoring* yang akan dibuat maka diperlukan desain *Use Case Diagram*. Gambar *Use Case Diagram* dari sistem ini dapat dilihat pada Gambar 3.2 *Use Case Diagram*



Gambar 3.2 *Use Case Diagram*

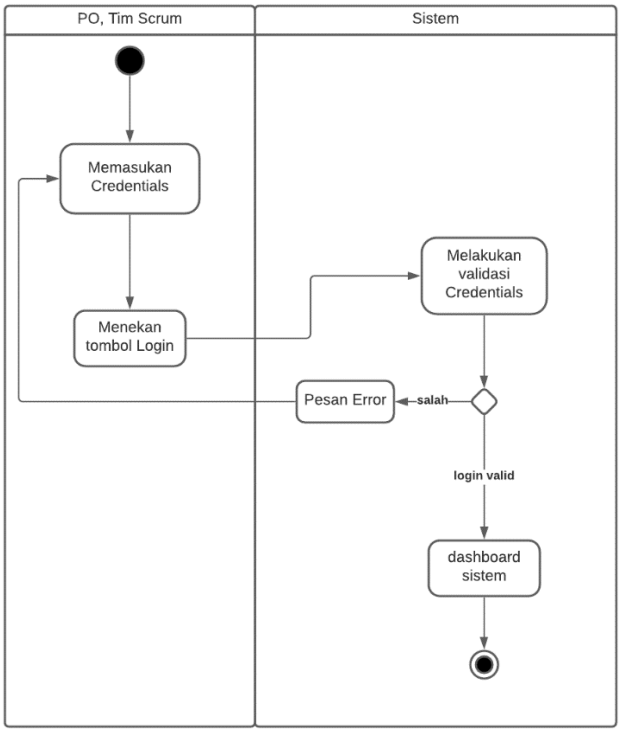
Use Case Diagram pada sistem ini memiliki 2 aktor yang berinteraksi dengan sistem yaitu *Product Owner*, dan Tim *Scrum.* Kedua *actor* dapat melakukan *Login, Product Owner* dapat melakukan proses membuat tim, membuat *objective*, membuat *key result*, membuat *initiative*, dan *memonitor progress* pengerjaan proyek berdasar pada *OKRs.* Tim *Scrum* dapat melakukan proses *update progress* dari *OKRs.*

3.2.2 *Activity Diagram*

Untuk mengetahui detail dari interaksi atau aktivitas yang terjadi pada setiap fungsi yang ada pada sistem maka dibutuhkan *Activity Diagram.* Berikut adalah desain dari *Activity Diagram.*

a. Fungsi dari *Login*

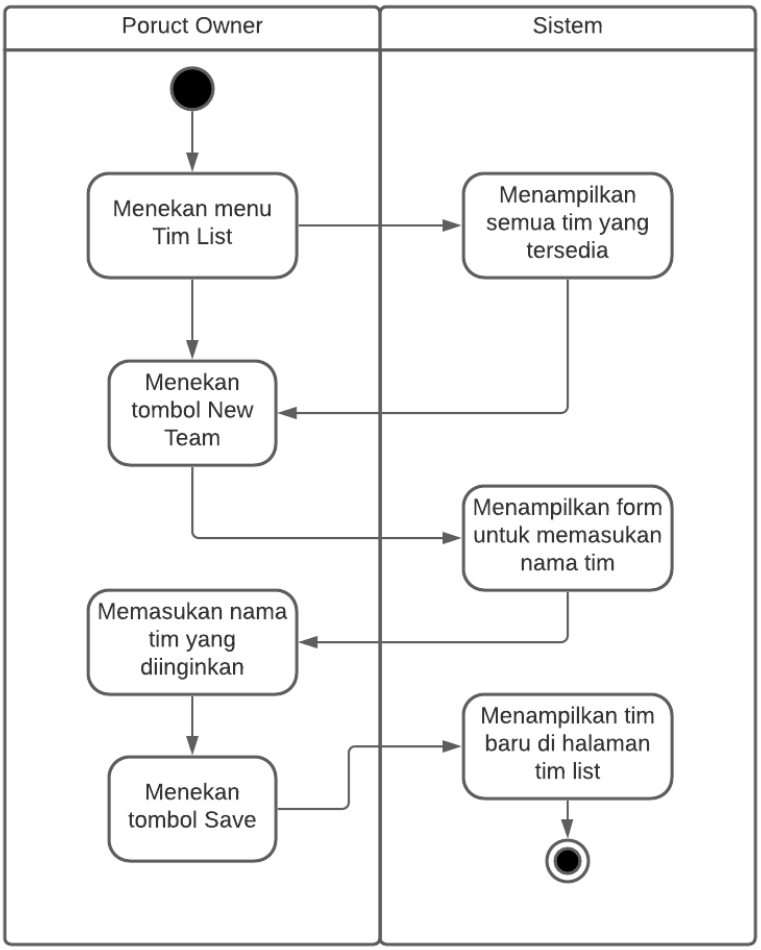
Semua pengguna sistem dapat melakukan fungsi *login* diawali dengan memasukan *credentials* di halaman *login*, kemudian menekan tombol *login.* Sistem akan melakukan validasi *credentials* apabila data yang dimasukkan salah sistem akan memunculkan pesan *error,* apabila data valid maka pengguna akan diteruskan ke halaman *dashboard*. Diagram dari aktivitas ini dapat dilihat pada Gambar 3.3 *Activity Diagram Login*



Gambar 3.3 *Activity Diagram Login*

b. Fungsi Membuat Tim Baru

Fungsi membuat tim baru di mana *Product Owner* menekan menu tim *list* untuk menunjukkan semua tim yang tersedia, lalu *Product Owner* menekan tombol New Team, sistem akan menampilkan *form* untuk memasukkan nama tim baru, selanjutnya *Product owner* memasukkan nama tim yang diinginkan dan menekan tombol *save*. Sistem akan menyimpan tim baru tersebut dan *Product owner* akan diarahkan Kembali ke halaman tim *list*. Aktivitas ini dapat dilihat pada Gambar 3.4 *Activity Diagram* Membuat Tim Baru

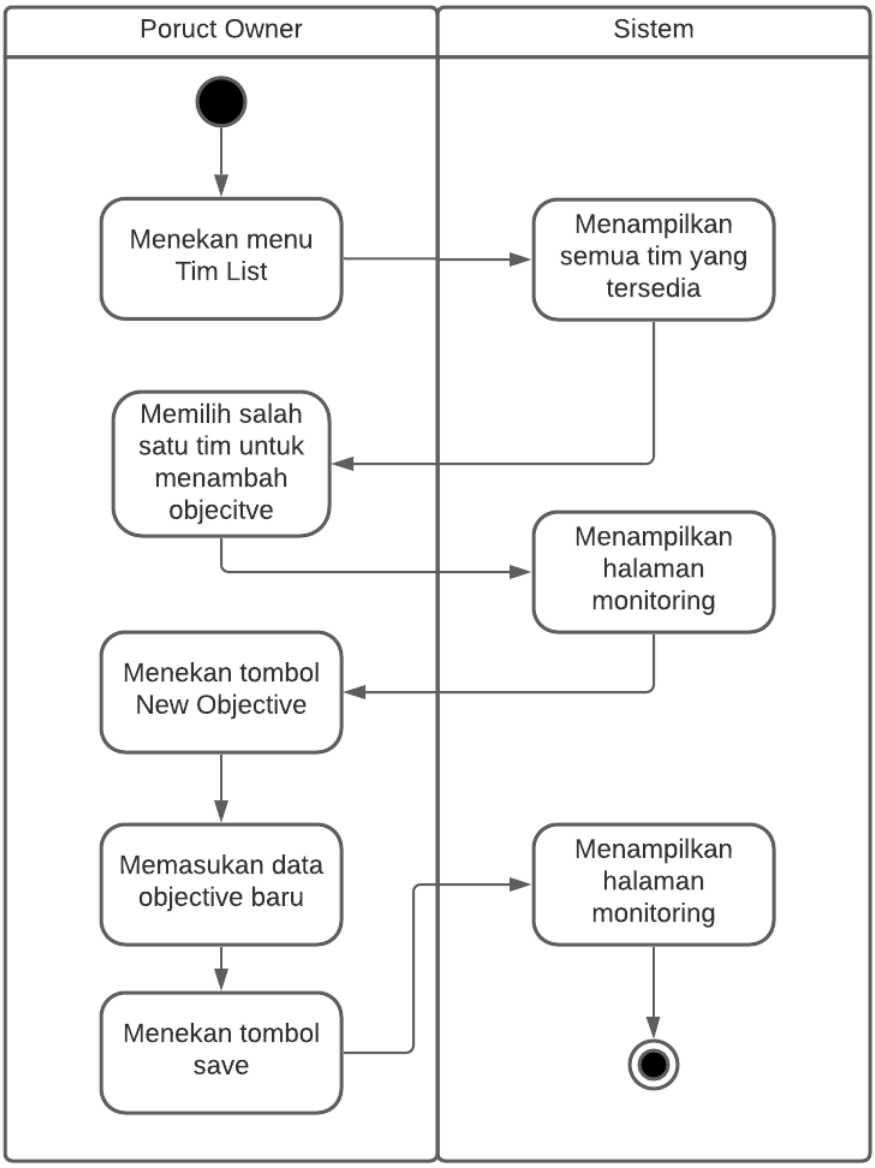


Gambar 3.4 *Activity Diagram* Membuat Tim Baru

c. Fungsi Membuat *Objective* Baru

Fungsi untuk membuat *objective* baru, *Product Owner* menekan tim *list* dan sistem akan menampilkan semua tim milik *Product Owner* yang tersedia, lalu *Product Owner* memilih salah satu tim untuk menambah *Objective* yang perlu diselesaikan. Sistem akan menampilkan halaman *monitoring* dari tim yang dipilih. *Product Owner* menekan tombol

New Objective yang tersedia di halaman *monitoring*, *Product Owner* dapat memasukkan data *Objective* baru dan menyimpannya dengan menekan tombol *save*, sistem akan menyimpan *Objective* baru yang sudah dibuat dan *Product Owner* akan diarahkan Kembali ke halaman *monitoring*. Aktivitas ini dapat dilihat pada Gambar 3.5 *Activity Diagram* Membuat *Objective* Baru.

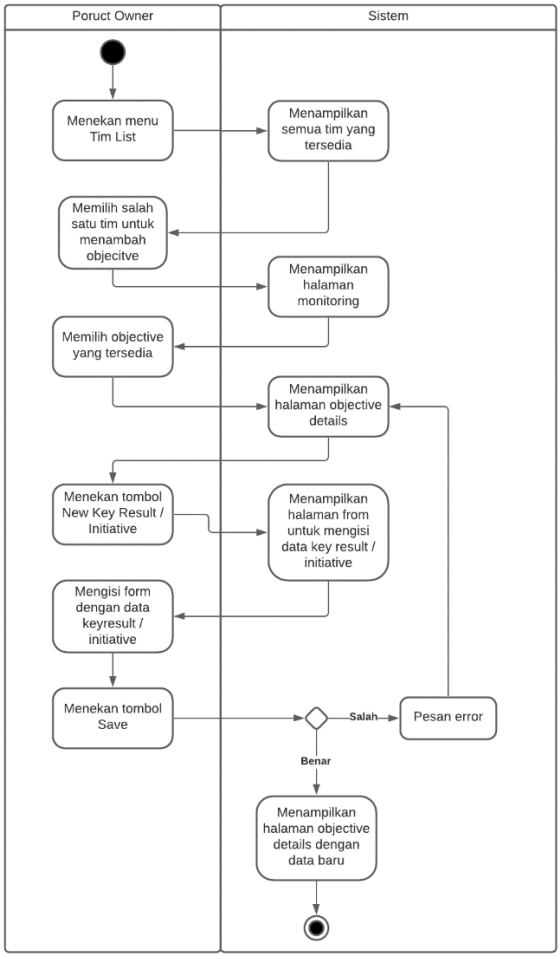


Gambar 3.5 *Activity Diagram* Membuat *Objective* Baru.

d. Fungsi Membuat *Key Result* atau *Initiative* Baru

Fungsi untuk membuat *Key Result* atau *Initiative* baru, *Product Owner* menekan tim *list* dan sistem akan menampilkan semua tim milik *Product Owner* yang tersedia, lalu *Product*

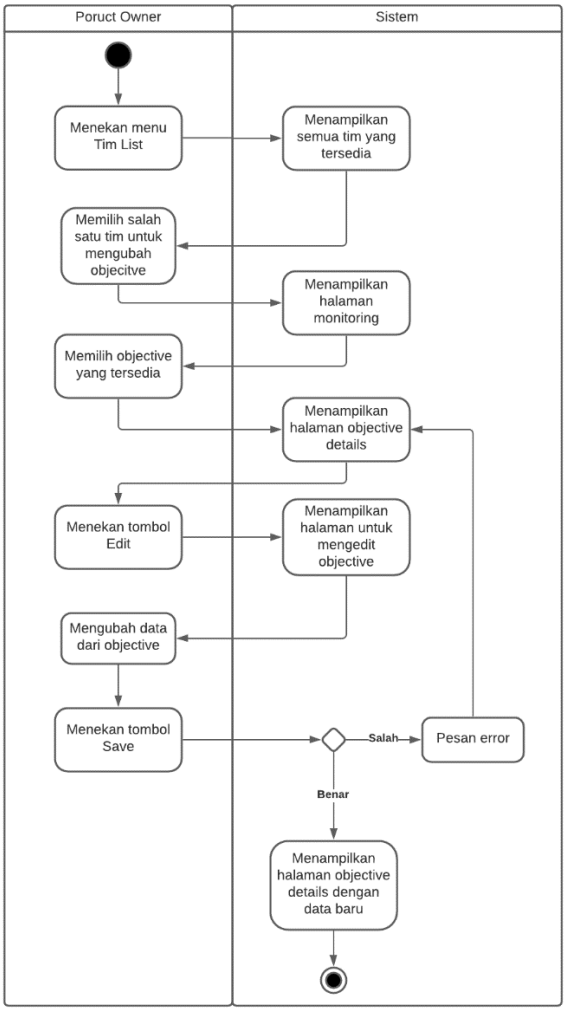
Owner memilih salah satu tim untuk menambah *Key Result* atau *Initiative* yang perlu diselesaikan. Sistem akan menampilkan halaman *monitoring* dari tim yang dipilih. *Product Owner* memilih salah satu *objectives* yang ada untuk ditambahkan *Key Result* atau *Initiative*-nya. Sistem akan menampilkan halaman *Objective Details*, pada halaman ini *Product Owner* dapat menekan tombol *New Key Result* atau *New Initiative.* Sistem akan menampilkan halaman form untuk menambahkan *Key Result* atau *Initiative* baru, *Product owner* memasukkan data untuk *Key Result* atau *Initiative* baru dan menekan tombol *save*, jika data valid sistem akan menyimpan *Key Result* atau *Initiative* baru dan mengarahkan *Product Owner* ke halaman *Objective Details.* Jika data tidak valid sistem akan mengirimkan pesan *error* dan mengarhkan *Product owner* ke halaman *Objective Detail*. Aktivitas ini dapat dilihat pada Gambar 3.6 *Activity Diagram* Membuat *Key Result* atau *Initiative* Baru



Gambar 3.6 *Activity Diagram* Membuat *Key Result* atau *Initiative* Baru

e. Fungsi Mengubah *Objective*

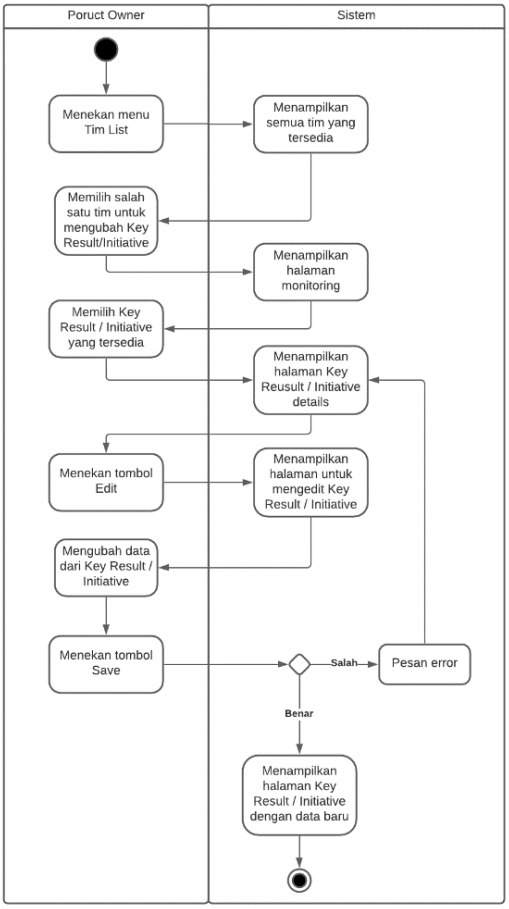
Fungsi untuk mengubah *objective*, *Product Owner* menekan tim list dan sistem akan menampilkan semua tim milik *Product Owner* yang tersedia, lalu *Product Owner* memilih salah satu tim untuk mengubah *Objective*. Sistem akan menampilkan halaman *monitoring* dari tim yang dipilih. *Product Owner* memilih *Objective* yang akan diubah lalu sistem akan menampilkan halaman detail dari *objective* tersebut. *Product Owner* menekan tombol Edit dan sistem akan menampilkan halaman untuk mengubah *objective*, *Product Owner* dapat memasukkan data baru dan menyimpan perubahan yang dilakukan dengan menekan tombol *save.* Jika data tidak valid sistem akan mengirimkan pesan *error* dan mengarahkan *Product owner* ke halaman *Objective Detail*. Aktivitas ini dapat dilihat pada Gambar 3.7



Gambar 3.7 *Activity Diagram* Mengubah *Objective*

f. Fungsi Mengubah *Key Result* atau *Inititaive*

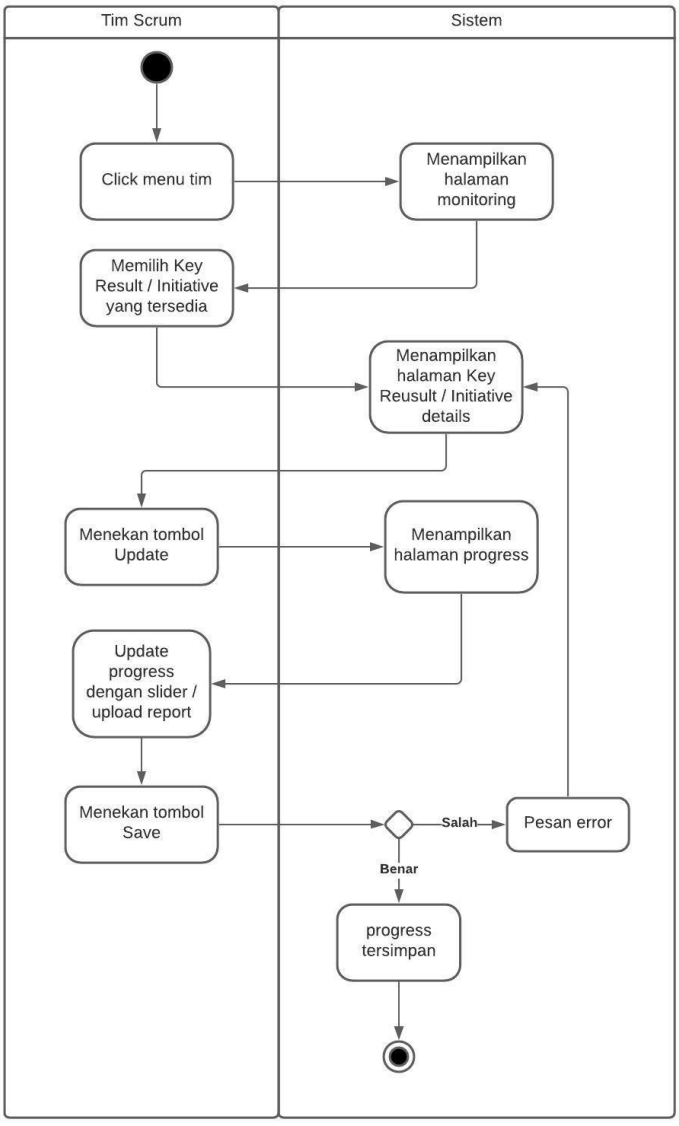
Fungsi untuk mengubah *Key Result* atau *Initiative*, *Product Owner* menekan tim *list* dan sistem akan menampilkan semua tim yang tersedia, lalu *Product Owner* memilih salah satu tim untuk mengubah *Key Result* atau *Initiative* yang dimiliki, sistem akan menampilkan halaman *monitoring* dari tim yang dipilih. *Product Owner* memilih salah satu *Key Result* atau *Initiative* yang ingin diubah, dan sistem akan menampilkan halaman detail dari *Key Result* atau *Initiative* tersebut, *Product Owner* menekan tombol edit dan memasukkan data baru untuk *Key Result* atau *Initiative*, lalu menekan tombol *save*, jika data valid sistem akan menyimpan *Key Result* atau *Initiative* baru dan mengarahkan *Product Owner* ke halaman *Key Result* atau *Initiative Details.* Jika data tidak valid sistem akan mengirimkan pesan *error* dan mengarahkan *Product owner* ke halaman *Key Result* atau *Initiative Detail*. Aktivitas ini dapat dilihat pada Gambar 3.8 *Activity Diagram* Mengubah *Key Result* atau *Initiative*



Gambar 3.8 *Activity Diagram* Mengubah *Key Result* atau *Initiative*

g. Fungsi *Mengupdate Progress Key Result* atau *Initiative*

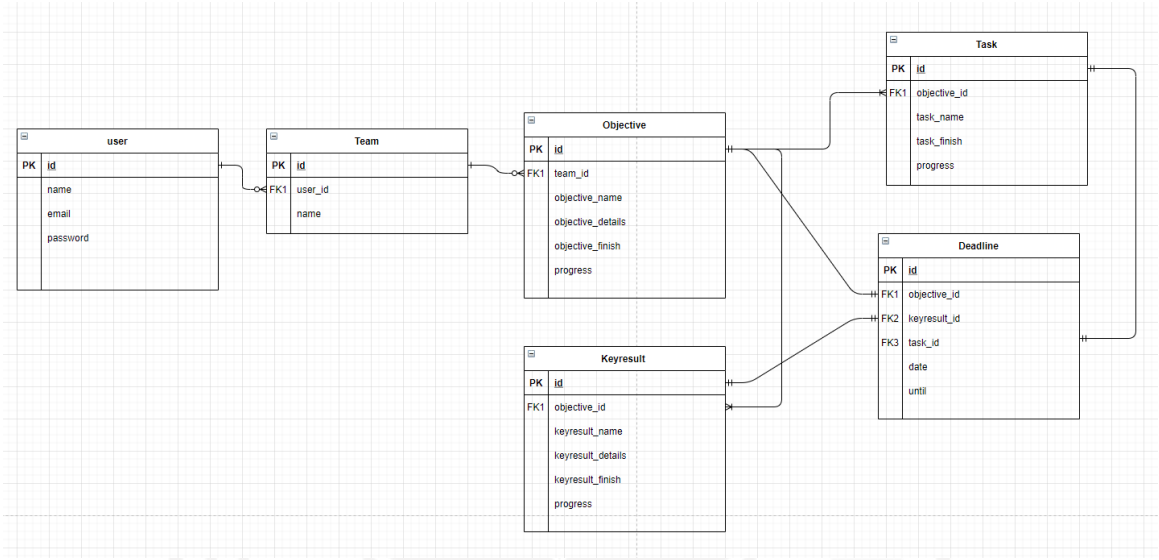
Fungsi untuk *Mengupdate Progress Key Result* atau *Initiative*, Tim *Scrum* memilih menu tim dan sistem akan menampilkan halaman *monitoring*, selanjutnya Tim *Scrum* memilih *key results* atau *initiative* yang tersedia, kemudian tim *Scrum* menekan tombol *update*, sistem akan menampilkan halaman *progress*. Tim *Scrum* dapat *mengupdate progress* menggunakan *slider* yang tersedia dan upload report jika dibutuhkan, lalu menekan tombol *save* untuk menyimpan *progress*.



Gambar 3.9 *Activity Diagram* Mengupdate *Progress Key Result* atau *Initiative*

3.2.3 *Database Ralation Design*

Pembuatan desain *database* sistem ini bertujuan untuk memberikan gambaran bagaimana *database* dan relasi dari setiap tabel yang ada dari sistem yang akan dibuat. Relasi tabel dapat dilihat pada Gambar 3.10 Desain Relasi *Database*



Gambar 3.10 Desain Relasi Database

Untuk mengetahui struktur tabel secara keseluruhan maka setiap tabel yang ada di dalam desain akan dijelaskan sebagai berikut:

Tabel 3.2 Tabel User

No	Atribut	Constraint	Keterangan
1	id	*Primary Key*	*Auto Increment, Not null*
2	name		*Not null*
3	email		*Unique, Not null*
4	password		*Not null*

Tabel *user* digunakan untuk menyimpan data dari pengguna sistem *monitoring*, seperti Tabel 3.2 Tabel User, tabel ini terdiri dari 4 kolom atribut. Atribut dari tabel ini adalah:

id : tipe data integer, *value* 255, *Primary key, Auto Increment*

name : tipe data *varchar*, *value* 255, *not null*

email : tipe data *varchar*, *value* 255, *not null*

password : tipe data *varchar*, *value* 255, *not null*

Tabel 3.3 Tabel Team

No	Atribut	Constraint	Keterangan
1	id	*Primary Key*	*Auto Increment, Not null*
2	user_id	*Foreign Key*	*Tabel user, Not null*
3	name		*Not null*

Tabel *team* digunakan untuk menyimpan nama dari semua tim yang dimiliki oleh *user.* Pada Tabel 3.3 Tabel Team terdiri dari 3 kolom atribut yaitu :

id : Tipe data *integer, value* 255, *Primary Key, Auto Increment*

user_id : Tipe data *integer, value* 255, *Foreign Key* memiliki relasi dari tabel user

name : tipe data *varchar, value* 255,

Tabel 3.4 Tabel Objective

No	Atribut	Constraint	Keterangan
1	id	*Primary Key*	*Auto Increment, Not null*
2	team_id	*Foreign Key*	*Tabel Team, Not null*
3	objective_name		*Not null*
4	objective_details		
5	objective_finish		
6	progress		

Tabel *Objective* digunakan untuk menyimpan *objectives* yang dimiliki oleh tim *Scrum.* Pada Tabel 3.4 Tabel *Objective* terdapat 6 Atribut yaitu :

id : Tipe data *integer, value* 255, *Primary Key, Auto Increment*

team_id : Tipe data *integer, value* 255, *Foreign Key* memiliki relasi dari tabel team

objective_name : tipe data *varchar, value* 255

objective_details : tipe data *varchar, value* 255

objective_finish : tipe data *Boolean*

objective_progress : tipe data *integer, value* 100

Tabel 3.5 Tabel Keyresult

No	Atribut	Constraint	Keterangan
1	id	*Primary Key*	*Auto Increment, Not null*
2	objective_id	*Foreign Key*	*Tabel Objective, Not null*
3	keyresult_name		*Not null*
4	keyresult_details		
5	keyresult_finish		
6	progress		

Tabel *Keyresult* digunakan untuk menyimpan *key result* yang dimiliki oleh tim *Scrum* tabel ini memiliki relasi *many to one* dengan tabel *Objective* Pada Tabel 3.5 Tabel Keyresult terdapat 6 atribut yaitu :

id : Tipe data *integer, value* 255, *Primary Key, Auto Increment*

objective_id : Tipe data *integer, value* 255, *Foreign Key* memiliki relasi dari tabel Objective

keyresult_name : tipe data *varchar, value* 255

keyresult_details : tipe data *varchar, value* 255

keyresult_finish : tipe data *Boolean*

keyresult_progress : tipe data *integer, value* 100

Tabel 3.6 Tabel Task

No	Atribut	Constraint	Keterangan
1	id	*Primary Key*	*Auto Increment, Not null*
2	objective_id	*Foreign Key*	*Tabel Objective, Not null*
3	task_name		*Not null*
4	task_finish		
5	progress		

Tabel *Taskt* digunakan untuk menyimpan *Initiative* yang dimiliki oleh tim *Scrum* tabel ini memiliki relasi *many to one* dengan tabel *Objective.* Pada Tabel 3.6 Tabel *Task* terdapat 5 atribut yaitu :

id : Tipe data *integer, value* 255, *Primary Key, Auto Increment*

objective_id : Tipe data *integer, value* 255, *Foreign Key* memiliki relasi dari tabel Objective

task_name : tipe data *varchar, value* 255

task_finish : tipe data *Boolean*

task_progress : tipe data *integer, value* 100

Tabel 3.7 Tabel Deadline

No	Atribut	Constraint	Keterangan
1	id	*Primary Key*	*Auto Increment, Not null*
2	objective_id	*Foreign Key*	*Tabel Objective*
3	keyresult_id	*Foreign Key*	*Tabel Keyresult*
4	task_id	*Foreign Key*	*Tabel Task*
5	date		
6	until		

Tabel *Deadline* digunakan untuk menyimpan *Deadline* yang dimiliki oleh tim *Scrum* tabel ini memiliki relasi *one to one* dengan tabel *Objective*, Tabel *Keyresult*, dan Tabel *Task*. Pada Tabel 3.7 Tabel Keyresult terdapat 6 atribut yaitu :

id : Tipe data *integer, value* 255, *Primary Key, Auto Increment*

objective_id : Tipe data *integer, value* 255, *Foreign Key* memiliki relasi dari tabel Objective

keyresult_id : Tipe data *integer, value* 255, *Foreign Key* memiliki relasi dari tabel Keyresult

task_id : Tipe data *integer, value* 255, *Foreign Key* memiliki relasi dari tabel Task

date : tipe data *date*

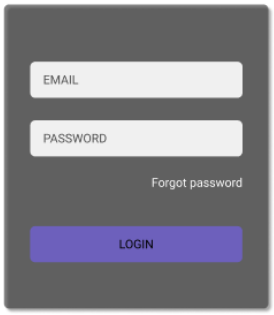
until : tipe data *date*

3.3 Prototyping

Setelah mengetahui bagaimana aktivitas yang akan dilakukan di dalam sistem Langkah selanjutnya adalah membuat sebuah *prototype* sistem, prototype ini berguna untuk memvisualisasikan bagaimana bentuk *interface* sistem ke depannya.

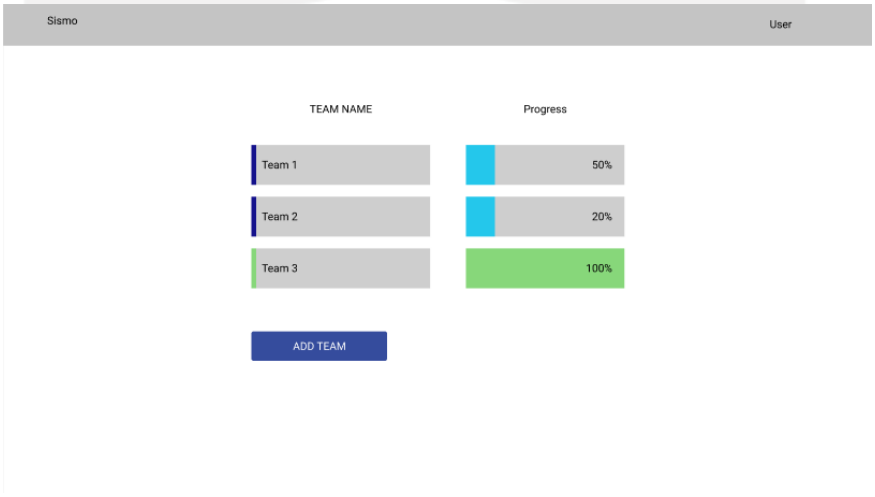
a. Desain Tampilan halaman *login*

Halaman *login* ini merupakan halaman awal sistem. Pada halaman ini pengguna dapat memasukkan email dan *password* untuk dapat melakukan *login* ke dalam sistem. Tampilan dari halaman *login* dapat dilihat pada Gambar 3.11 Desain Halaman *Login.*



Gambar 3.11 Desain Halaman *Login*

b. Desain Tampilan halaman *Team List*

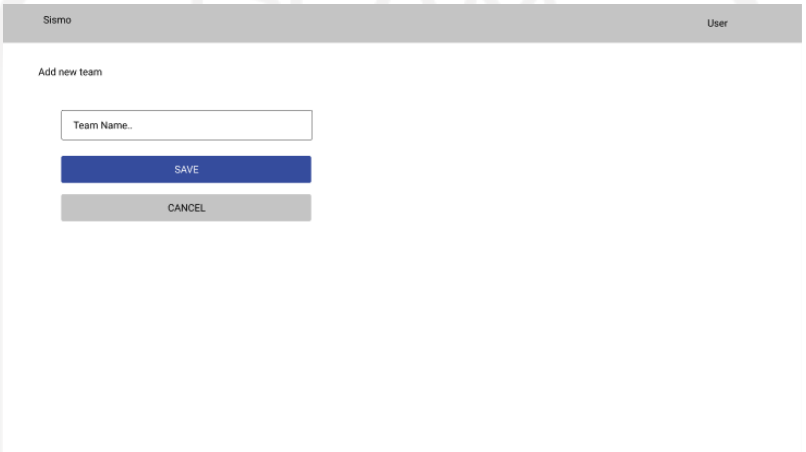


Gambar 3.12 Desain Halaman Team List

Pada Gambar 3.12 Desain Halaman *Team List*, Halaman ini merupakan halaman yang menampilkan semua tim yang dimiliki oleh *Product Owner*, halaman ini juga menunjukkan *progress OKRs* dari masing-masing tim yang ada.

c. Desain Tampilan halaman *add team*

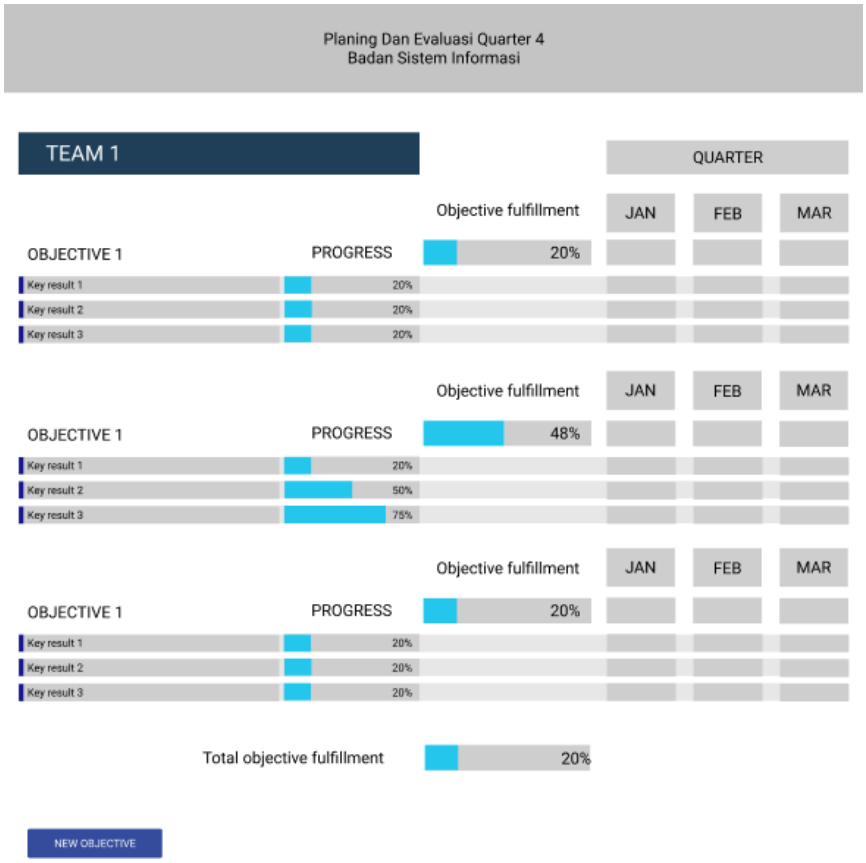
Pada desain halaman *add team* atau *new team* ini terdapat sebuah *form* untuk memasukkan sebuah nama untuk tim yang akan dibuat. Tampilan dari desain halaman ini dapat dilihat pada Gambar 3.13 Desain Halaman *Add Team*.



Gambar 3.13 Desain Halaman *Add Team*

d. Desain Halaman *Monitoring*.

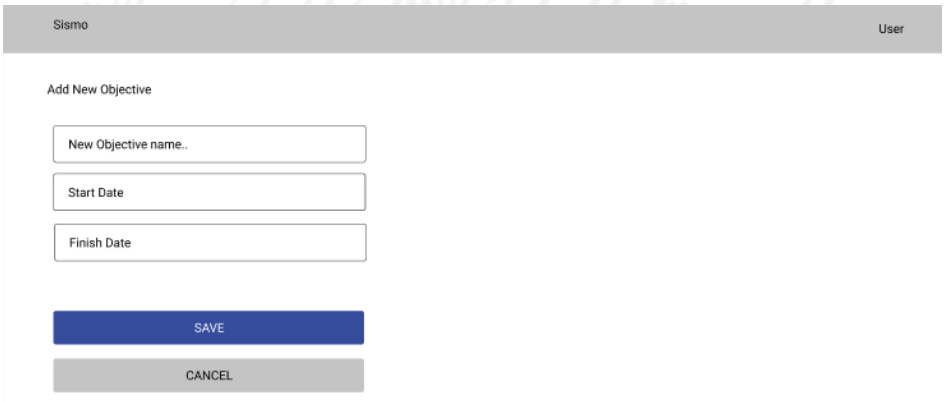
Pada desain halaman *monitoring* ini akan menampilkan semua *objective, key result* dan *initiative* beserta progres dan *quarter* dari setiap *objective, key result, dan initiative* yang dimiliki oleh sebuah tim, halaman *monitoring* ini juga mempunyai *objective fullfilment* dan *total objective fullfilment* yang digunakan untuk mengukur progres kerja tim, halaman *monitoring* ini dapat pada Gambar 3.14 Design Halaman *Monitoring*.



Gambar 3.14 Desain Halaman *Monitoring*

e. Desain Halaman *New Objective*

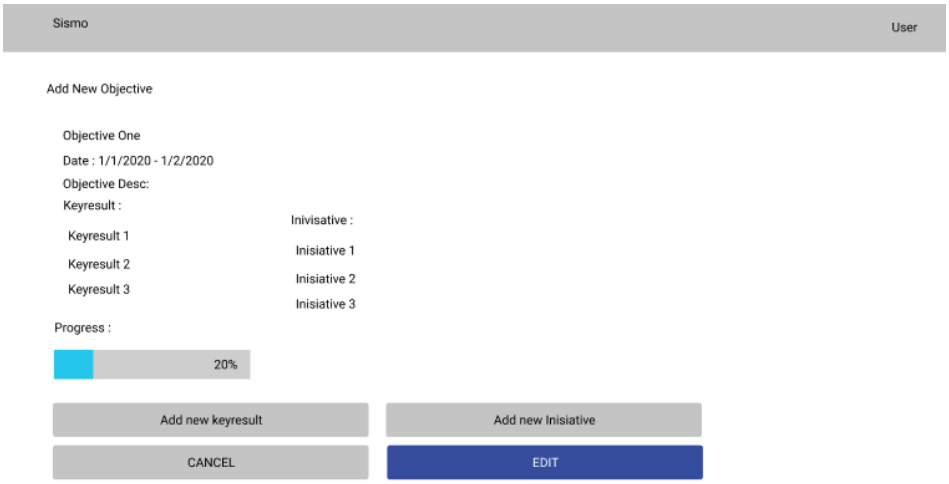
Pada desain halaman *New Objective* ini terdapat *form* yang digunakan untuk memasukkan nama *objective* baru dan menentukan tanggal mulai dan selesainya. Tampilan dari desain halaman new *objective* dapat dilihat pada Gambar 3.15 Desain Halaman *New Objective.*



Gambar 3.15 Desain Halaman *New Objective*

f. Desain Halaman Objective Details

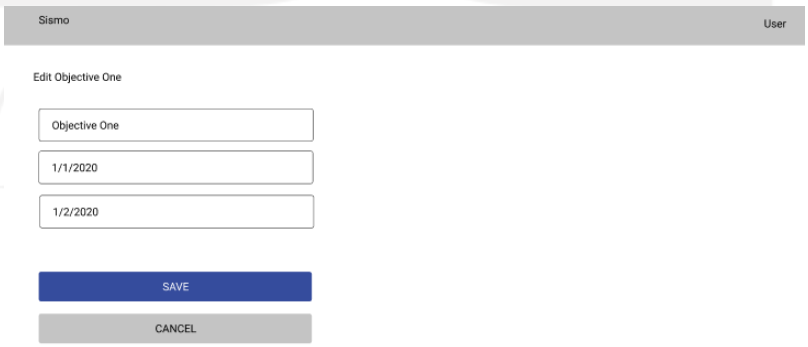
Pada desain halaman *objective details*, halaman ini menampilkan deskripsi dari *objective* dan semua *key result dan initiative* yang dimiliki oleh sebuah *objective*, halaman ini juga menampilkan progress dari *objective* tersebut. Tampilan halaman ini dapat dilihat pada Gambar 3.16 Desain Halaman *Objective Details.*



Gambar 3.16 Desain Halaman *Objective Details.*

g. Desain Halaman Edit Objective

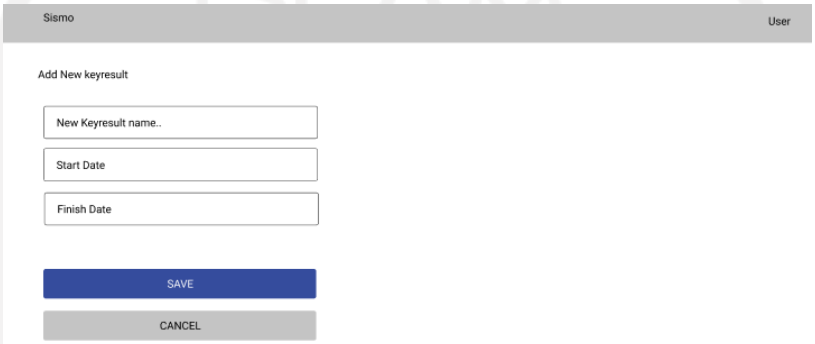
Pada desain halaman *edit objective* ini terdapat *form* yang digunakan untuk mengubah data dari *objective* yang ingin diubah. Tampilan dari desain halaman *edit objective* dapat dilihat pada Gambar 3.15 Desain Halaman *Edit Objective*



Gambar 3.17 Desain Halaman *Edit Objective*

h. Desain Halaman *New Key Result*

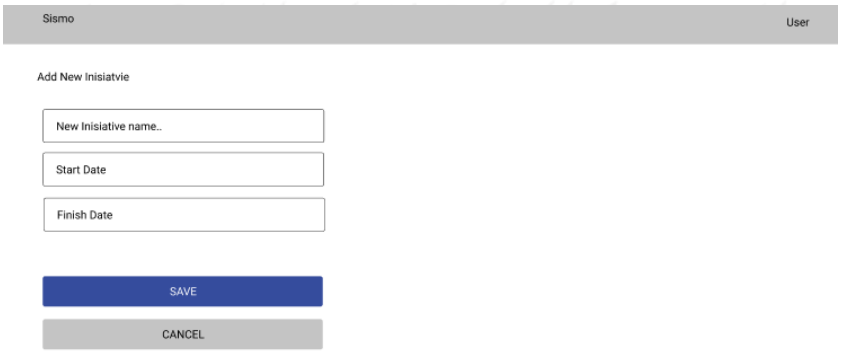
Pada desain halaman *New Key Result* ini terdapat *form* yang digunakan untuk memasukkan nama *Key Result* baru dan menentukan tanggal mulai dan selesainya. Tampilan dari desain halaman *new key result* dapat dilihat pada Gambar 3.18 Desain Halaman *New Key Result.*



Gambar 3.18 Desain Halaman *New Key Result*

i. Desain Halaman *New Initiative*

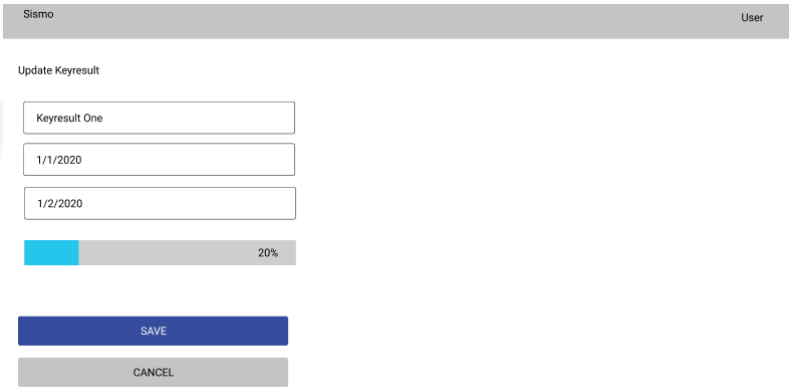
Pada desain halaman *new initiative* ini terdapat *form* yang digunakan untuk memasukkan nama *initiative* baru dan menentukan tanggal mulai dan selesainya. Tampilan dari desain halaman *new initiative* dapat dilihat pada Gambar 3.19 Desain Halaman *New Initiative.*



Gambar 3.19 Desain Halaman *New Initiative*

j. Desain Halaman Update Keyresult atau Initiative

Pada desain halaman *update key result* atau *initiative* ini terdapat *from* dan *slider* yang digunakan untuk *mengupdate progress* dari pengerjaan *key result* atau *initiative* yang sudah dikerjakan sejauh ini. Tampilan desain halaman ini dapat dilihat pada Gambar 3.20 Desain Halaman *Update Key Result* atau *Initiative*



Gambar 3.20 Desain Halaman *Update Key Result* atau *Initiative*

BAB IV
IMPLEMENTASI DAN PENGUJIAN

Bab ini membahas tentang implementasi dan pengujian sistem, untuk menerapkan rancangan sistem yang telah dibuat pada bab sebelumnya. Setelah sistem diimplementasikan akan dilakukan pengujian sistem untuk memastikan semua fungsi dari sistem bekerja sebagaimana mestinya.

4.1 Implementasi

Untuk memudahkan pemahaman implementasi sistem penulis menggunakan pendekatan Skenario dan Implementasi Sistem. Pada skenario akan dijelaskan bagaimana proses bisnis pada sistem berjalan, lalu skenario tersebut akan divisualisasikan pada bagian Implementasi sistem.

4.1.1 Skenario

Skenario ini memiliki 2 aktor yaitu, *Product owner* dan Tim *Scrum*. Pada skenario ini *Product owner* berperan sebagai penanggung jawab dan pemilik proyek, sekaligus yang menentukan apa saja *Objectives* dan *Key Results* yang dikerjakan untuk menyelesaikan proyek yang dibuat, Tim *Scrum* berperan sebagai pengembang yang bertujuan untuk memenuhi *Objective* dengan cara menyelesaikan *Key Result* dan *Initiative* dari *Objective.* Skenario dapat dilihat pada Tabel 4.1 Skenario Penggunaan Sistem

Tabel 4.4.1 Skenario Penggunaan Sistem

No	Aktor	Tahapan	Skenario
1	*Product Owner*	Membuat tim baru	-*Product owner* membuat sebuah tim baru untuk mengerjakan suatu proyek
2	*Product Owner*	Membuat *OKRs* baru untuk tim tersebut	-*Product owner* membuat *Objective* dan *key result* yang akan digunakan menjadi acuan penyelesaian proyek

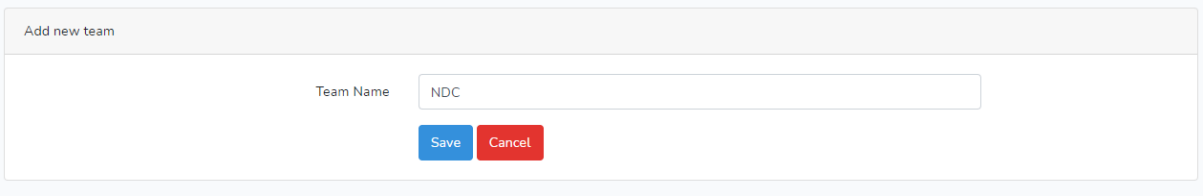
3	Tim *Scrum*	*Mengupdate Progress* pengerjaan *OKRs*	-Tim *Scrum* akan terus *mengupdate progress* dari *Key result* dan *initiative* yang dikerjakan selama masa proyek berlangsung
4	*Product Owner*	Memonitor progress dari proyek berdasarkan *OKRs*	-*Product Owner* akan terus memantau kinerja dan *progress* dari tim *scrum* berdasarkan *OKRs* yang dicapai selama pengerjaan proyek

4.1.2 Implementasi Sistem

Implementasi sistem ini akan memvisualkan bagaimana sistem bekerja berdasarkan skenario yang telah dibuat dengan tahapan sebagai berikut:

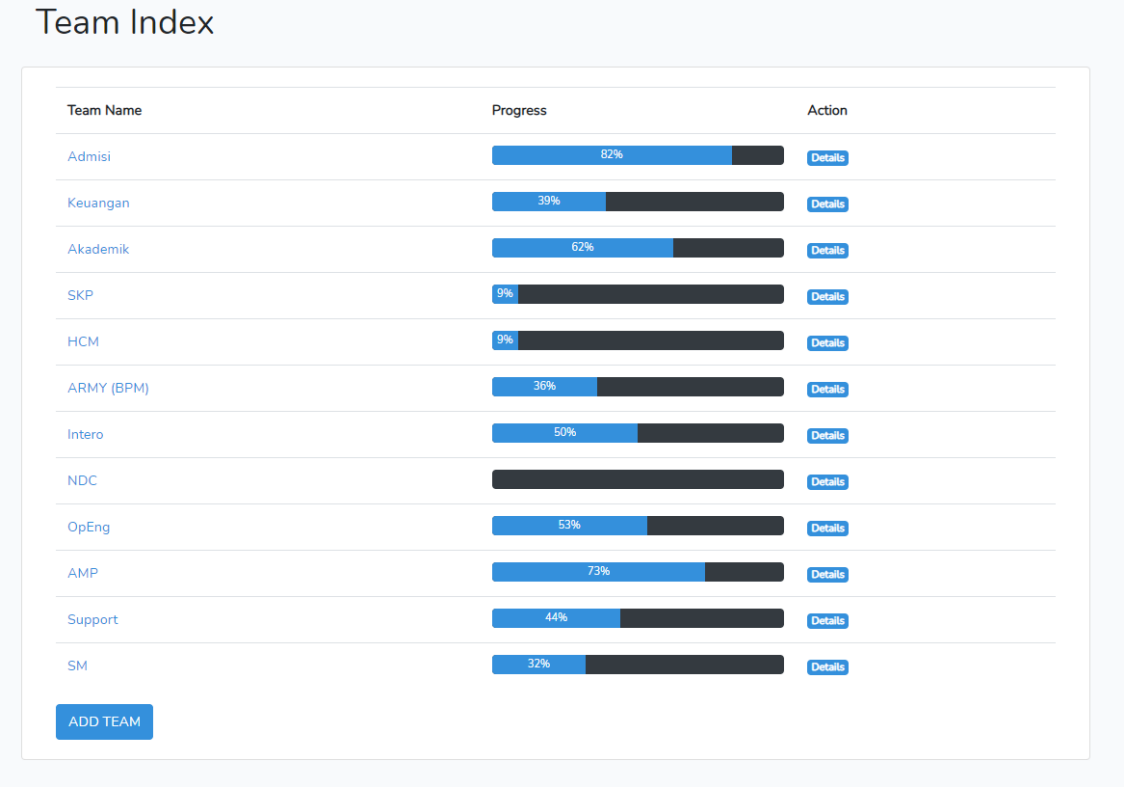
a. *Product Owner* Membuat sebuah tim

Pada Implementasi sistem ini, *Product Owner* membuat sebuah tim untuk mengerjakan sebuah proyek, pada proses pembuatan tim ini *Product owner* dapat memasukkan nama tim yang diinginkan. Tampilan implementasi sistem untuk pembuatan tim baru dapat dilihat pada Gambar 4.1 *Form* Pembuatan Tim Baru.



Gambar 4.1 *Form* Pembuatan Tim Baru

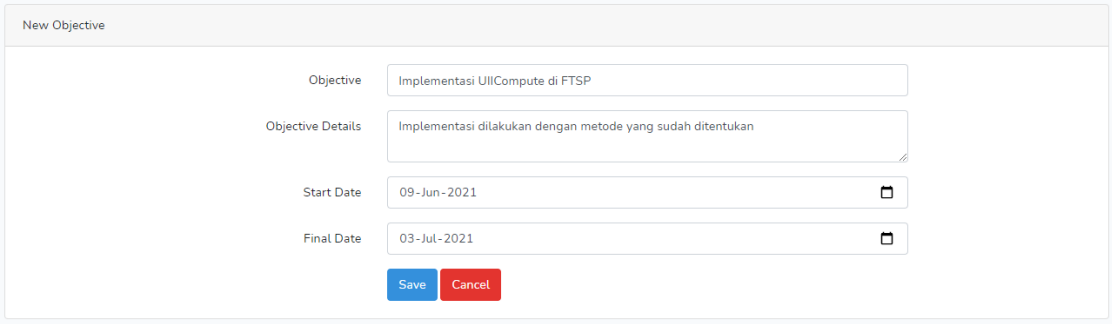
Setelah *Product Owner* menekan tombol *save,* maka tim yang sudah dibuat tadi akan muncul di halaman tim *list* seperti pada Gambar 4.2 Implementasi Tim *List,* halaman ini menampilkan semu tim yang tersedia atau yang sudah dibuat sebelumnya.



Gambar 4.2 Implementasi Tim List

b. *Product Owner* Membuat *OKRs* baru untuk tim tersebut

Setelah membuat sebuah tim untuk mengerjaka suatu proyek, *Product Owner* harus membuat *Objectives* yang akan diguanakan sebagai acuan pencapaian proyek data yang dimasukan yaitu *Objective name, Objective Details, Start Date,* dan *Final Date.* Pembuatan *Objectives* dapat dilihat pada Gambar 4.3 Form Pembuatan *Objective*



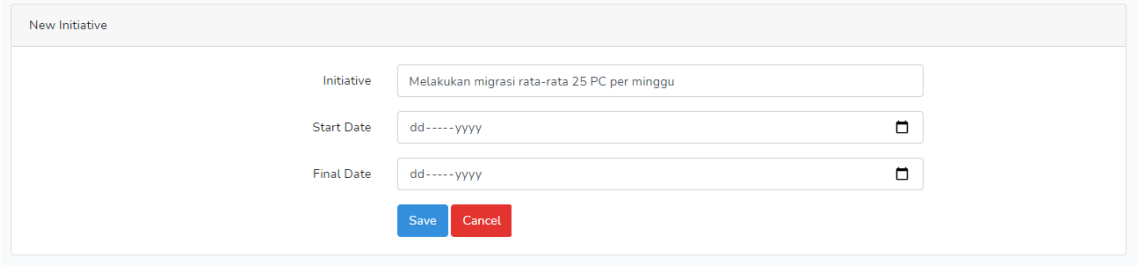
Gambar 4.3 Form Pembuatan *Objective*

Setelah *Objective* dibuat *Product Owner* juga akan membuat *Key Result* dan *Intiative* yang digunakan untuk memenuhi atau mewujudkan *objective*. Untuk pembuatan *Key result* data yang dimasukkan yaitu *Key Result name, Key Result Detail, Start Date,* dan *Final date,* seperti pada Gambar 4.4 *Form* Pembuatan *Key Result.*



Gambar 4.4 *Form* Pembuatan *Key Result.*

Selanjutnya *Product Owner* menambahkan *Initiative* dari *objective* untuk membantu memenuhi *Key Result.* Untuk pembuatan *initiative* data yang dimasukan adalah *intitaive name, Start Date,* dan *Finish Date,* seperti pada Gambar 4.5 *Form* Pembuatan *Initiative.*



Gambar 4.5 *Form* Pembuatan *Intitiatve*

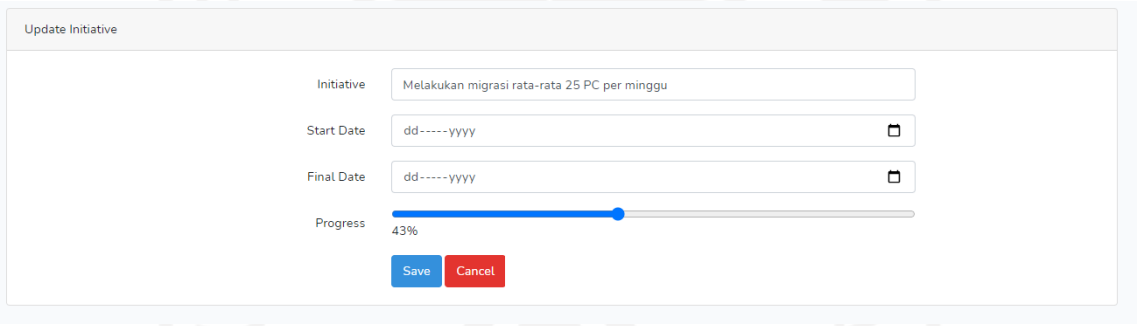
c. *Tim Scrum mengupdate progress* dari pengerjaan *OKRs*

Untuk memenuhi atau mewujudkan sebuah *objective, progress* dari *OKRs* tim *Scrum* perlu mengerjakan *Key Result.* Lalu untuk menunjukkan *progress* dari pekerjaan *Key result* tersebut, Tim *Scrum* dapat menggunakan *slider progress* pada halaman *Update key result.* Tampilan implementasi ini dapat dilihat pada Gambar 4.6 Halaman *Update* Progress *Key Result.*



Gambar 4.6 Halaman Update Progress *Key Result*

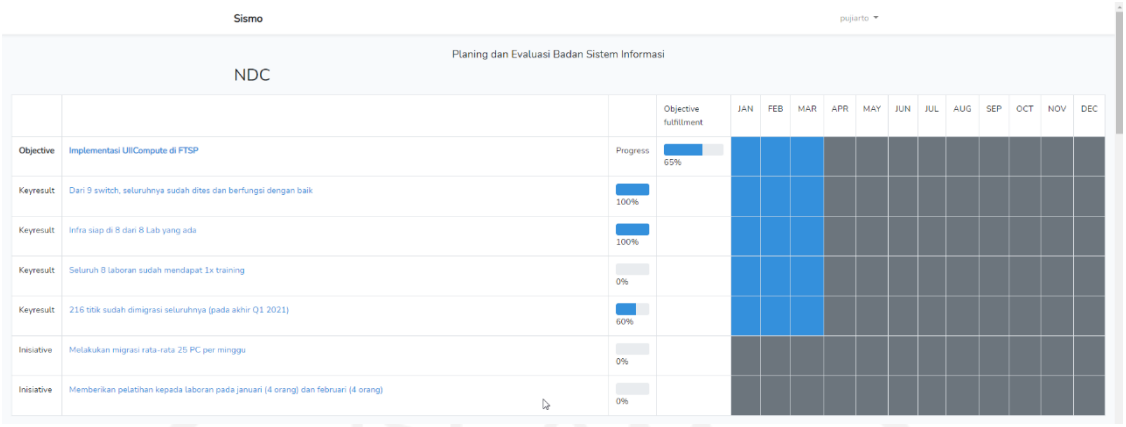
Selanjutnya Tim *Scrum* juga dapat *mengupdate progress Initiative* yang sudah dilakukan untuk membantu mencapai *key result* dan *objective*. Tim *Scrum* dapat menggunakan *slider progress* pada halaman *Update Initiatives* untuk *mengupdate progress* yang sudah dicapai sejauh ini. Tampilan dari implementasi ini dapat dilihat pada Gambar 4.7 Halaman *Update Initiatives*.



Gambar 4.7 Halaman *Update Initiative*

d. *Product Owner* Memonitor *progress* dari proyek berdasarkan *OKRs*

Untuk mengetahui bagaimana pencapaian atau perkembangan *objective* dari Tim *Scrum* berdasar pada *OKRs*, *Product owner* dapat melihat pencapaian *objective* melalui halaman *monitoring*. Halaman ini dapat dilihat pada Gambar 4.8 Halaman *Monitoring* Tim, halaman ini menampilkan semua *OKRs* dan *quarter* pengerjaan proyek.



Gambar 4.8 Halaman *Monitoring* Tim

4.2 Pengujian

Bagian ini akan menjelaskan pengujian sistem yang dilakukan yaitu dengan menggunakan metode pengujian *functional testing* yang memanfaatkan *use case*. Pengujian ini dilakukan untuk memastikan sistem dapat berjalan dengan baik. Pengujian dilakukan terhadap fungsi sistem.

4.2.1 *Functional Testing*

Pengujian ini dilakukan berdasarkan *use case* untuk melihat apakah fitur yang dimiliki oleh sistem dapat bekerja sebagaimana mestinya. Hasil dari pengujian yang dilakukan dapat dilihat pada Tabel 4.2 *Functional Testing.*

Tabel 4.2 *Funtional Testing*

No	Aktivitas	Hasil	Kesimpulan
1	Pengguna *Login*	Pengguna berhasil *login* jika data yang dimasukkan valid	Berhasil
2	Melihat daftar tim	Sistem Menampilkan daftar tim yang tersedia	Berhasil
3	Membuat Sebuah tim	Pembuatan tim baru berhasil disimpan ke dalam sistem	Berhasil
4	Membuat *Objective*	Pembuatan *objective* baru berhasil disimpan	Berhasil

5	Membuat *Key Result*	Pembuatan *key result* baru berhasil disimpan	Berhasil
6	Membuat *Initiative*	Pembuatan initiative baru berhasil disimpan	Berhasil
7	Melihat data monitor tim	Data *monitoring* tim dapat ditampilkan	Berhasil
8	*Update progress* tim *Scrum*	Sistem berhasil menyimpan dan menampilkan *progress OKRs* dari Tim *Scrum*	Berhasil

4.2.2 Pengujian Usabilitas

Pada pengujian ini akan menggunakan *Skala Likert*, dengan *Skala Likert* ini responden diminta untuk melengkapi kuesioner. Berikut hasil dari penghitungan nilai menggunakan *Skala Likert* dapat dilihat pada Tabel 4.3 Hasil Kuesioner Pengujian Sistem.

Tabel 4.3 Hasil Kuesioner Pengujian Sistem

No	Pertanyaan	Jawaban					Persentase
		STS	TS	R	S	SS	
1	Sistem Memiliki alur dan fungsi yang mudah di pahami				3	2	88%
2	Sistem berjalan dengan baik tanpa adanya error			1	4		76%
3	Sistem mampu memonitor tim berdasarkan *OKRs*				4	1	84%
4	Sistem memiliki tampilan yang menarik			2	3		72%
Rata Rata Persentase							80%

Berdasarkan dari hasil kuesioner pengujian usabilitas pada tabel 4.3 diperoleh rata-rata skor dengan persentase 80%, Maka hasil pengujian sistem ini berada pada ketegori sangat setuju.

BAB V
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan pada penelitian dan pembahasan tugas akhir mengenai Sistem Monitoring Kinerja Pencapaian Objective Tim Scrum Menggunakan OKR,maka dapat diambil kesimpulan bahwasanya:

a. Implementasi sistem yang dibuat bisa berjalan dengan baik tanpa kendala, serta memiliki alur dan fungsi yang jelas dan mudah dipahami
b. Sistem dapat memberikan informasi kinerja Tim *Scrum* berdasarkan pada pencapaian *objective*

5.2 Saran

Saran untuk pengembangan sistem lebih lanjut, diharapkan Sistem Monitoring Kinerja Pencapaian Objective Tim Scrum Menggunakan OKR ini ke depannya dapat dikembangkan sebagai berikut :

a. Menambahkan fitur *export* dan *import* menggunakan Microsoft Excel untuk melakukan dokumentasi data.
b. Menambahkan fitur untuk menampilkan data kinerja Tim *Scrum* dari seluruh tim yang ada.

DAFTAR PUSTAKA

Amy, E., Blackwell, H., & Manar, E. (2019). *Editors: Amy Hackney Blackwell and Elizabeth Manar Date: 2015 From: UXL Encyclopedia of Science(3*. *June*, 2019.

Eric J. (n.d.).

Jerry Zeyu Gao, H.-S. Jacob Tsao, Y. W. (2003). *Testing and quality assurance for component-based software*.

Mahnic, V., & Zabkar, N. (2012). Measuring progress of scrum-based software projects. *Elektronika Ir Elektrotechnika*, *18*(8), 73–76. https://doi.org/10.5755/j01.eee.18.8.2630

Niven, P. R., & Lamorte, B. (2016). Objectives and Key Results. *Objectives and Key Results*. https://doi.org/10.1002/9781119255543

Schwaber, K., & Sutherland, J. (2011). The Scrum Papers: Nut, Bolts, and Origins of an Agile Framework. *SCRUM Inc.*, 224. http://jeffsutherland.com/ScrumPapers.pdf

Schwaber, Ken, & Sutherland, J. (2017). The Scrum Guide: The Definitive The Rules of the Game. *Scrum.Org and ScrumInc*, *November*, 19. http://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-US.pdf

Sugiyono. (2008). *Metode penelitian pendidikan:(pendekatan kuantitatif, kualitatif dan R \& D)*. Alfabeta.

Wesley, P. A. (2015). Unified Modeling Language User Guide , The (2nd Edition) (Addison-Wesley Object Technology Series) Unified Modeling Language User Guide , The Unified Modeling Language User Guide , The Many of the designations used by manufacturers and sellers to dist. In *ResearchGate* (Vol. 2nd, Issue August).

LAMPIRAN