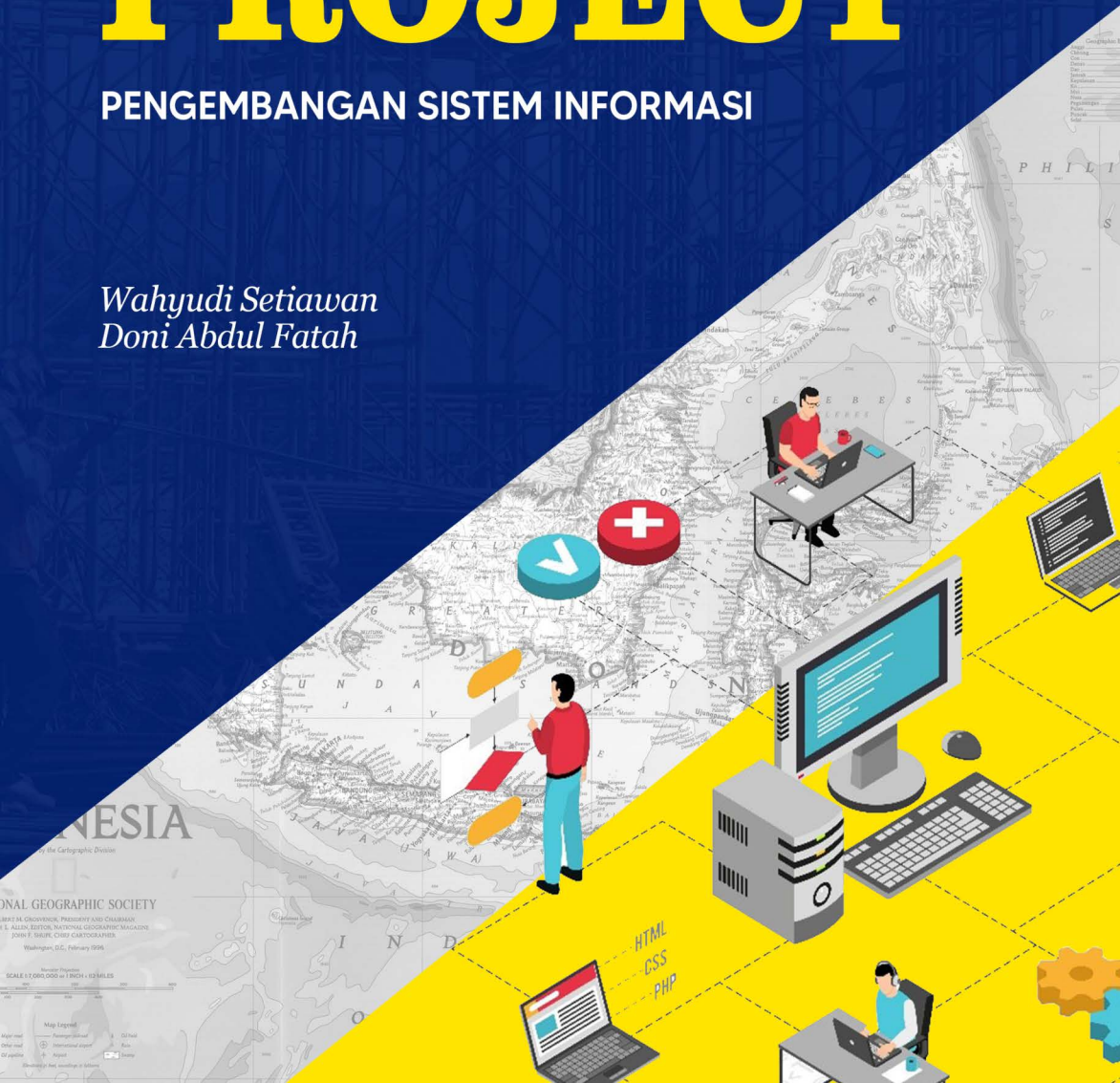




Wahyudi Setiawan
Doni Abdul Fatah



CAPSTONE PROJECT
PENGEMBANGAN SISTEM
INFORMASI

CAPSTONE PROJECT
PENGEMBANGAN SISTEM INFORMASI

Wahyudi Setiawan
Doni Abdul Fatah



CAPSTONE PROJECT

PENGEMBANGAN SISTEM INFORMASI

Penulis:
Wahyudi Setiawan
Doni Abdul Fatah

Editor:
Putri Jasmine

Layouter :
Tim Kreatif PRCI

Cover:
Rusli

Cetakan Pertama : Juni 2023

Hak Cipta 2022, pada Penulis. Diterbitkan pertama kali oleh:

Perkumpulan Rumah Cemerlang Indonesia
ANGGOTA IKAPI JAWA BARAT
Pondok Karisma Residence Jalan Raflesia VI D.151
Panglayungan, Cipedes Tasikmalaya – 085223186009

Website : www.rcipress.rcipublisher.org
E-mail : rumahcemerlangindonesia@gmail.com

Copyright © 2022 by Perkumpulan Rumah Cemerlang Indonesia
All Right Reserved

- Cet. I – : Perkumpulan Rumah Cemerlang Indonesia, 2022
; 15,5 x 23 cm
ISBN : 978-623-448-557-8

Hak cipta dilindungi undang-undang
Dilarang memperbanyak buku ini dalam bentuk dan dengan
cara apapun tanpa izin tertulis dari penulis dan penerbit

Undang-undang No.19 Tahun 2002 Tentang
Hak Cipta Pasal 72

Undang-undang No.19 Tahun 2002 Tentang Hak Cipta
Pasal 72

Barang siapa dengan sengaja melanggar dan tanpa hak melakukan perbuatan sebagaimana dimaksud dalam pasal ayat (1) atau pasal 49 ayat (1) dan ayat (2) dipidana dengan pidana penjara masing-masing paling sedikit 1 (satu) bulan dan/atau denda paling sedikit Rp.1.000.000,00 (satu juta rupiah), atau pidana penjara paling lama 7 (tujuh) tahun dan/atau denda paling banyak Rp.5.000.000.000,00 (lima miliar rupiah).

Barang siapa dengan sengaja menyiarkan, memamerkan, mengedarkan, atau menjual kepada umum suatu ciptaan atau barang hasil pelanggaran hak cipta terkait sebagai dimaksud pada ayat (1) dipidana dengan pidana penjara paling lama 5 (lima) tahun dan/atau denda paling banyak Rp.500.000.000,00 (lima ratus juta rupiah)

KATA PENGANTAR

Puji syukur kehadirat Allah SWT atas rahmat serta karunia- Nya, sehingga buku ajar ini telah dapat diselesaikan. Buku ini sebagai pedoman bagi Mahasiswa Sistem Informasi. Dalam menunjang pembelajaran mata kuliah.

Penulis menyadari apabila dalam penyusunan buku ini masih terdapat kekurangan, tetapi penulis meyakini sepenuhnya bahwa sekecil apapun buku ini tetap memberikan manfaat.

Akhir kata untuk menyempurnakan buku ini, kritik dan saran dari pembaca sangatlah berguna untuk penulis kedepannya. penulis berharap buku ini dapat memberikan manfaat bagi para pembacanya.

Penulis

DAFTAR ISI

KATA PENGANTAR	I
DAFTAR ISI	II
DAFTAR GAMBAR	IV
DAFTAR TABEL	VI
BAB I KONSEP <i>CAPSTONE PROJECT</i>	1
A. Definisi dan Konsep <i>Capstone Project</i>	1
B. <i>Capstone Project</i> pada bidang Sistem Informasi	2
C. Pentingnya <i>Capstone Project</i>	4
D. Tujuan dan manfaat <i>Capstone Project</i>	6
BAB II METODOLOGI PENGEMBANGAN <i>CAPSTONE PROJECT</i> SISTEM INFORMASI	9
A. Metodologi Pengembangan Proyek Sistem Informasi	9
1. Metodologi <i>Waterfall</i>	10
2. Metodologi <i>Agile</i>	13
3. Metodologi <i>Spiral</i>	31
4. Metodologi RAD (<i>Rapid Application Development</i>)	34
B. Tahapan Pengembangan Proyek	39
BAB III KONSEP PENELITIAN PROYEK SISTEM INFORMASI	41
A. Dasar Penelitian	41
B. Teknik penyusunan laporan proyek	42
C. Sistematika penulisan proyek	44
BAB IV STUDI KASUS <i>CAPSTONE PROJECT</i> SISTEM INFORMASI	48
A. Studi Kasus 1	48
B. Latar Belakang	48
C. Kualitas Daging Sapi	49
a) Daging Sapi Segar (<i>Fresh</i>)	50
b) Daging Sapi Busuk (<i>Spoiled</i>)	50
D. <i>Convolutional Neural Network</i> (CNN)	51
E. Perancangan Sistem	54
a) Pengumpulan Data	54
b) Tahap <i>preprocessing</i>	55
c) Tahap Training dan Testing CNN	56
F. Tahap Pembangunan Program	58

G.	Hasil Analisis	68
1.	Studi Kasus 2	72
2.	Latar Belakang	72
3.	Perancangan Sistem	74
	<i>a. Desain Thinking</i>	74
	<i>b. User Experience Questionnaire (UEQ)</i>	76
	c. Penentuan Sampel Responden	79
4.	Tahapan Pengembangan	80
	a) Tahap <i>Emphthize</i>	80
	b) Tahapan <i>Define</i>	88
	c) Tahapan <i>Ideate</i>	90
	d) Tahapan <i>Prototype</i>	92
	e) Tahapan <i>Testing</i>	99
	DAFTAR PUSTAKA	104

DAFTAR GAMBAR

Gambar 1. 1. Ilustrasi Capstone Project	1
<i>Gambar 2. 1. System Development Life Cycle (SDLC) (Santander 2018)</i>	9
Gambar 2. 2. Metodologi <i>Waterfall</i>	11
Gambar 2. 3. <i>Agile</i> SDLC Models	13
Gambar 2. 4. Tahapan pengembangan proyek metode <i>Scrum</i>	16
Gambar 2. 5. Visualisasi scrum task board	20
Gambar 2. 6. Tahapan <i>Extreme Programming</i>	23
Gambar 2. 7. Model <i>Lean Software Development</i>	25
Gambar 2. 8. fase dalam Metodologi Spiral	32
Gambar 2. 9. Tahapan Metodologi RAD (Rapid Application Development)	35
Gambar 4. 1. Daging sapi segar (<i>fresh</i>)	50
Gambar 4. 2. Daging sapi busuk (<i>spoiled</i>)	51
Gambar 4. 3. Arsitektur MLP Sederhana	52
Gambar 4. 4. Proses Konvolusion pada CNN	52
Gambar 4. 5. Operasi Konvolusion	53
Gambar 4. 6. Operasi Max Pooling	54
Gambar 4. 7. Perancangan sistem	54
Gambar 4. 8. Citra asli	55
Gambar 4. 9. <i>Cropping</i> dan <i>resize</i>	56
Gambar 4. 10. Proses Training	57
Gambar 4. 11. Proses Testing	57
Gambar 4. 12. Source code membuat Data Set	59
Gambar 4. 13. Kode Program Resize Citra	61
Gambar 4. 14. Kode Program Pembagian Data Set Training dan Test	62
Gambar 4. 15. Kode Program Membuat Model	63
Gambar 4. 16. Kode Program Tahap Evaluasi	65
Gambar 4. 17. Kode Program Matplotlib	66
Gambar 4. 18. Kode Program Result Model	67
Gambar 4. 19. Hasil Program Analisis	68
Gambar 4. 20. Hasil Program Epoch	69

Gambar 4. 21. Halaman Depan	70
Gambar 4. 22. Halaman Deteksi Daging	70
Gambar 4. 23. Halaman Proses Training	71
Gambar 4. 24. Proses Design Thinking	75
Gambar 4. 25. Halaman Utama	81
Gambar 4. 26. Halaman Berita	82
Gambar 4. 27. Halaman Utama Digilab	82
Gambar 4. 28. Halaman Profile	83
Gambar 4. 29. Struktur Organisasi	83
Gambar 4. 30. Denah UPT	84
Gambar 4. 31. Halaman Detail Informasi Koleksi	84
Gambar 4. 32. Halaman Self Upload	85
Gambar 4. 33. Halaman Utama Katalog	85
Gambar 4. 34. Halamann Pencarian Katalog	86
Gambar 4. 35. Uji Reliability	88
Gambar 4. 36. <i>Journey map</i>	91
Gambar 4. 37. Site Map perbaikan	92
Gambar 4. 38. Halaman <i>Homepage</i>	93
Gambar 4. 39. <i>Dashboard</i> OPAC	94
Gambar 4. 40. Dashboard Digilib	95
Gambar 4. 41. <i>Dashboard E-megazine</i>	96
Gambar 4. 42. Halaman Desain <i>Homepage</i>	97
Gambar 4. 43. Halaman Desain Dahboard OPAC	98
Gambar 4. 44. <i>E-Megazine Dashboard</i>	99
Gambar 4. 45. <i>Scales Mean and Variance</i>	102
Gambar 4. 46. Hasil Perbandingan Dengan <i>Skala Benchmark</i>	103

DAFTAR TABEL

Tabel 4. 1. Pertanyaan UEQ	77
Tabel 4. 2. <i>Interval Benchmark</i>	79
Tabel 4. 3. Rata-rata Impresi dan Varians Skala	87
Tabel 4. 4. <i>How Might We</i>	89
Tabel 4. 5. Sampel Data Hasil Kuisioner UEQ	100
Tabel 4. 6. Konversi Sampel Data Hasil Kuisioner UEQ	101

BAB I

KONSEP *CAPSTONE PROJECT*

Konsep *Capstone Project* mengacu pada sebuah proyek akhir yang melibatkan penerapan pengetahuan dan keterampilan yang telah diperoleh selama masa studi mulai dari semester 1 hingga semester 6. Proyek ini dirancang untuk mengintegrasikan berbagai aspek pembelajaran menjadi suatu karya yang komprehensif dan relevan dengan bidang studi sistem informasi.



Gambar 1. 1. Ilustrasi Capstone Project
[Gambar Ilustrasi *Capstone Project*] Sumber: Freepik
(Freepik.com)

A. Definisi dan Konsep *Capstone Project*

Capstone Project dalam konteks mata kuliah ini merujuk pada sebuah proyek akhir atau paling tidak dapat dilanjutkan menjadi sebuah skripsi yang bertujuan untuk mengintegrasikan

pengetahuan dan keterampilan yang telah diperoleh selama masa studi dalam bidang sistem informasi. Proyek ini merupakan proses perjalanan akademik mahasiswa yang mencakup penerapan konsep-konsep teoritis dalam suatu proyek yang relevan sesuai dengan bidang minat dan juga merupakan penerapan atau implementasi dari mata kuliah-mata kuliah yang telah ditempuh pada semester sebelumnya.

Capstone Project dapat didefinisikan sebagai “sebuah proyek penutup yang mengharuskan mahasiswa untuk mengintegrasikan, menerapkan, dan mendemonstrasikan pemahaman mereka terhadap pengetahuan dan keterampilan yang telah diperoleh selama masa studi, melalui pengembangan dan implementasi suatu solusi dalam bidang sistem informasi”.

Pada *Capstone Project*, mahasiswa dihadapkan pada berbagai tugas, termasuk perencanaan, analisis, desain, implementasi, dan evaluasi proyek. Metodologi yang relevan digunakan untuk memandu mahasiswa dalam mengembangkan proyek secara sistematis dan efektif. *Capstone Project* juga melibatkan kolaborasi, pemecahan masalah, dan inovasi. Mahasiswa ditantang untuk mengidentifikasi dan mengatasi masalah nyata dalam bidang studi dengan solusi yang kreatif dan relevan.

Tujuan utama dari *Capstone Project* adalah menghadirkan suatu solusi nyata yang dapat memecahkan masalah atau memenuhi kebutuhan dalam konteks sistem informasi serta dapat mengembangkan kemampuan analitis, pemikiran kritis, dan keterampilan komunikasi yang mendalam.

B. *Capstone Project* pada bidang Sistem Informasi

Capstone Project pada bidang Sistem Informasi merupakan komponen integral dalam kurikulum pendidikan tinggi yang bertujuan untuk menggabungkan pengetahuan teoritis dan penerapan praktis dalam konteks Sistem Informasi. *Capstone Project* memungkinkan mahasiswa untuk mengembangkan pemahaman mendalam tentang konsep dan prinsip yang terkait

dengan Sistem Informasi serta mengaplikasikannya dalam situasi dunia nyata.

Dalam konteks ini, *Capstone Project* pada bidang Sistem Informasi bertujuan untuk melatih mahasiswa dalam mengidentifikasi, menganalisis, dan memecahkan masalah yang kompleks yang terkait dengan sistem informasi organisasi. *Capstone Project* pada bidang Sistem Informasi memungkinkan mahasiswa untuk mengembangkan keterampilan pemecahan masalah, pemikiran kritis, dan kemampuan beradaptasi dalam menghadapi tantangan yang kompleks dalam industri teknologi informasi.

Capstone Project pada bidang Sistem Informasi juga memberikan kesempatan bagi mahasiswa untuk berkolaborasi dengan mitra industri. Hal ini memungkinkan mahasiswa untuk memahami lebih dalam kebutuhan dan tantangan yang dihadapi oleh organisasi dalam mengimplementasikan sistem informasi yang efektif. Kolaborasi ini juga dapat memberikan wawasan praktis dan peluang kerja bagi mahasiswa setelah lulus.

Komponen-komponen *Capstone Project* pada bidang Sistem Informasi secara umum, diantaranya:

1. Identifikasi masalah atau kebutuhan: Tahap awal dalam *Capstone Project* adalah mengidentifikasi masalah atau kebutuhan yang ingin diselesaikan. Dalam konteks Sistem Informasi, ini melibatkan pemahaman tentang tantangan atau peluang dalam pengelolaan, analisis, atau pengembangan sistem informasi.
2. Analisis dan perancangan sistem: Setelah masalah atau kebutuhan diidentifikasi, tahap berikutnya adalah melakukan analisis sistem yang meliputi pemahaman tentang persyaratan bisnis, kebutuhan pengguna, dan lingkungan organisasi. Selanjutnya, perancangan sistem dilakukan untuk merancang solusi yang sesuai dengan kebutuhan tersebut.
3. Implementasi sistem: Komponen ini melibatkan implementasi solusi yang telah dirancang.

Pengembangkan sistem informasi yang relevan dengan kebutuhan dan tujuan proyek. Pada tahap ini, pengujian dan debugging juga dilakukan untuk memastikan sistem berfungsi dengan baik.

4. Evaluasi dan analisis hasil: Setelah sistem diimplementasikan, tahap evaluasi dilakukan untuk mengukur efektivitas dan efisiensi solusi yang telah dikembangkan. Evaluasi ini dapat melibatkan pengumpulan data, pengujian fungsionalitas, serta analisis hasil yang mencakup penilaian terhadap keberhasilan proyek dalam memenuhi tujuan awal.
5. Penyusunan laporan dan presentasi: Bagian terakhir dari *Capstone Project* adalah menyusun laporan tertulis yang menjelaskan seluruh proses proyek, termasuk masalah yang diidentifikasi, solusi yang dikembangkan, dan hasil evaluasi. Selain itu, presentasi proyek juga dilakukan di hadapan dosen pembimbing dan pihak terkait lainnya untuk memaparkan temuan dan hasil proyek.

C. Pentingnya Capstone Project

Capstone Project memegang peranan yang sangat penting dalam mengaplikasikan pengetahuan dan keterampilan yang telah diperoleh mahasiswa dalam bidang Sistem Informasi. Melalui *Capstone Project*, mahasiswa memiliki kesempatan untuk menerapkan teori yang mereka pelajari dalam konteks nyata, menghadapi tantangan praktis, dan mengembangkan solusi yang inovatif.

Capstone Project memberikan kesempatan bagi mahasiswa untuk menerapkan pemahaman mereka tentang konsep dan prinsip Sistem Informasi dalam situasi dunia nyata. Hal ini penting karena Sistem Informasi menjadi bagian integral dalam operasional organisasi modern. Dengan menerapkan pengetahuan dan keterampilan Sistem Informasi dalam *Capstone Project*, mahasiswa dapat mengasah kemampuan mereka untuk

memecahkan masalah yang kompleks dan memenuhi kebutuhan bisnis.

Selain itu, *Capstone Project* juga memberikan kesempatan bagi mahasiswa untuk berkolaborasi dengan mitra industri. Melalui kolaborasi ini, mahasiswa dapat memperoleh wawasan praktis tentang kebutuhan dan tantangan yang dihadapi oleh organisasi dalam mengimplementasikan sistem informasi yang efektif. Interaksi langsung dengan mitra industri membantu mahasiswa memahami konteks bisnis yang lebih luas dan memperoleh pengalaman kerja yang berharga.

Selain meningkatkan keterampilan teknis, *Capstone Project* juga melatih mahasiswa dalam hal kemampuan analitis, pemikiran kritis, dan komunikasi. *Capstone Project* memungkinkan mahasiswa untuk mengembangkan kemampuan pemecahan masalah yang kritis dan beradaptasi dengan cepat terhadap perubahan teknologi dan tuntutan pasar.

Pentingnya *Capstone Project*, sebagai berikut:

1. Aplikasi Praktis Pengetahuan dan Keterampilan

Capstone Project memungkinkan mahasiswa untuk menerapkan pengetahuan dan keterampilan yang telah mereka pelajari dalam situasi dunia nyata. Dengan melakukan proyek yang relevan dengan bidang Sistem Informasi, mahasiswa dapat mengaplikasikan konsep dan prinsip yang telah mereka pelajari dalam solusi yang praktis dan terukur.

2. Pemecahan Masalah Kompleks

Capstone Project sering melibatkan pemecahan masalah yang kompleks dan nyata yang dihadapi oleh organisasi dalam konteks Sistem Informasi. Hal ini membantu mahasiswa mengasah kemampuan analitis dan pemikiran kritis dalam mengidentifikasi dan merancang solusi yang efektif.

3. Kolaborasi dengan Mitra Industri

Capstone Project sering melibatkan kolaborasi dengan mitra industri, seperti perusahaan atau organisasi yang relevan dengan bidang Sistem Informasi. Melalui kolaborasi ini, mahasiswa dapat memperoleh wawasan langsung tentang

kebutuhan industri, tantangan yang dihadapi, dan praktik terbaik dalam pengembangan sistem informasi.

4. Pengembangan Keterampilan Komunikasi

Capstone Project melibatkan penyusunan laporan tertulis dan presentasi proyek kepada dosen pembimbing dan pihak terkait lainnya. Hal ini membantu mahasiswa mengembangkan keterampilan komunikasi yang penting dalam menyampaikan ide, temuan, dan hasil proyek secara efektif.

5. Persiapan Karir Profesional

Capstone Project membantu mahasiswa dalam mempersiapkan diri untuk karir profesional dalam bidang Sistem Informasi. Melalui pengalaman praktis dalam menghadapi tantangan dan menerapkan solusi, mahasiswa menjadi lebih siap untuk memasuki dunia kerja dan berkontribusi dalam pengembangan sistem informasi yang relevan dengan kebutuhan industri.

D. Tujuan dan manfaat *Capstone Project*

Capstone Project memiliki tujuan yang jelas dan memberikan manfaat yang signifikan bagi mahasiswa dan industri terkait. Melalui *Capstone Project*, mahasiswa dapat mengembangkan keterampilan dan pengetahuan yang relevan dengan bidang studi mereka, sementara industri dapat memperoleh manfaat langsung dari hasil kerja mahasiswa.

Tujuan *Capstone Project* adalah sebagai berikut:

1. Pengaplikasian Pengetahuan

Capstone Project bertujuan untuk memberikan kesempatan bagi mahasiswa untuk mengaplikasikan pengetahuan yang telah mereka peroleh selama masa studi. Melalui proyek ini, mahasiswa dapat menghubungkan teori dengan praktik nyata dalam konteks industri yang relevan.

2. Pengembangan Keterampilan

Capstone Project bertujuan untuk mengembangkan keterampilan yang penting dalam dunia kerja, seperti keterampilan pemecahan masalah, analitis, komunikasi, kerja tim, dan kepemimpinan. Proyek ini memberikan platform untuk

mahasiswa memperkuat keterampilan ini melalui tugas-tugas yang melibatkan penelitian, analisis, desain, implementasi, dan evaluasi.

3. Kolaborasi Industri

Capstone Project bertujuan untuk membangun hubungan erat antara universitas dan industri. Melalui proyek ini, mahasiswa dapat bekerja sama dengan perusahaan atau organisasi terkait dalam mengatasi masalah nyata yang dihadapi oleh industri. Hal ini memungkinkan mahasiswa untuk memperoleh wawasan langsung tentang industri tersebut dan mengembangkan pemahaman mendalam tentang kebutuhan dan tantangan di lapangan.

Manfaat *Capstone Project* bagi mahasiswa adalah sebagai berikut:

1. Penerapan Pengetahuan

Capstone Project memberikan kesempatan bagi mahasiswa untuk menerapkan pengetahuan yang telah mereka pelajari dalam konteks nyata. Melalui pengalaman praktis ini, mahasiswa dapat menggabungkan teori dengan pengalaman langsung, memperkuat pemahaman mereka, dan mengembangkan kepercayaan diri dalam menerapkan pengetahuan dalam situasi yang kompleks.

2. Pengembangan Keterampilan Praktis

Capstone Project membantu mahasiswa mengembangkan keterampilan praktis yang relevan dengan industri. Misalnya, kemampuan analitis, pemecahan masalah, pemikiran kritis, komunikasi efektif, dan kemampuan beradaptasi dengan teknologi dan perubahan pasar.

3. Peluang Karir

Capstone Project memberikan mahasiswa kesempatan untuk memperluas jaringan profesional mereka dan mendapatkan pengalaman kerja yang berharga. Melalui kolaborasi dengan industri, mahasiswa dapat membangun hubungan dengan praktisi di bidang Sistem Informasi, meningkatkan peluang untuk mendapatkan pekerjaan atau kesempatan magang setelah lulus.

Manfaat *Capstone Project* bagi industri adalah sebagai berikut:

1. Inovasi dan Solusi Praktis

Capstone Project memberikan kesempatan bagi industri untuk mendapatkan solusi inovatif dan praktis untuk masalah atau tantangan yang dihadapi. Dengan melibatkan mahasiswa yang memiliki pemahaman teoritis dan keterampilan praktis, industri dapat memperoleh pemikiran segar dan pendekatan baru dalam mengatasi masalah yang ada.

2. Rekrutmen Mahasiswa Berpotensi

Capstone Project memungkinkan industri untuk melihat potensi mahasiswa yang akan segera lulus dan memiliki minat dalam bidang Sistem Informasi. Industri dapat mengidentifikasi mahasiswa yang memiliki keterampilan dan pengetahuan yang relevan, sehingga memudahkan proses rekrutmen.

3. Kolaborasi dengan Perguruan Tinggi

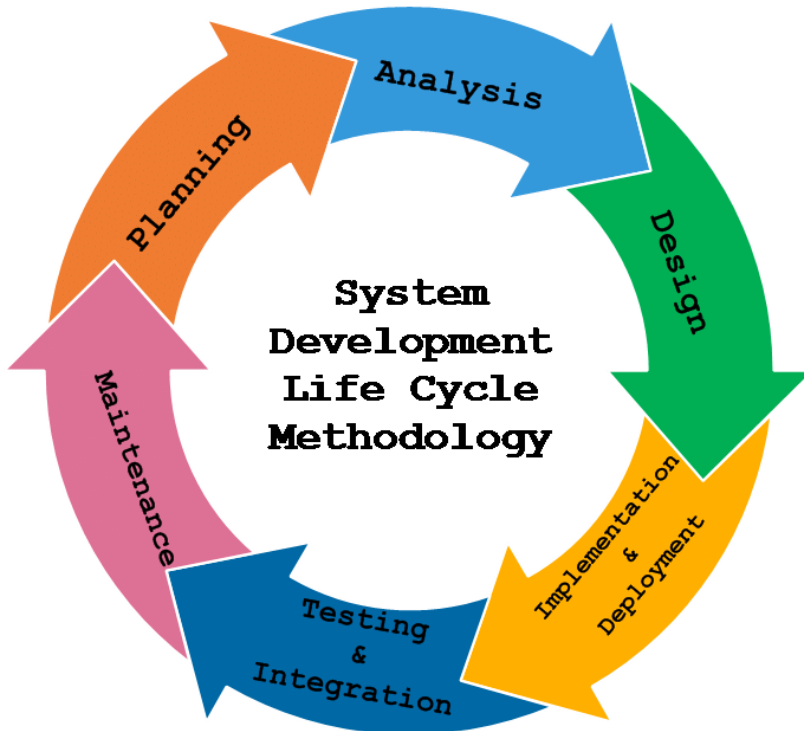
Capstone Project membangun hubungan kolaboratif antara industri dan perguruan tinggi. Melalui kolaborasi ini, industri dapat berkontribusi dalam pengembangan kurikulum yang relevan dengan kebutuhan industri, memberikan wawasan tentang tren terkini, dan berpartisipasi dalam mengarahkan pendidikan yang lebih terhubung dengan dunia industri.

BAB II

METODOLOGI PENGEMBANGAN *CAPSTONE PROJECT* SISTEM INFORMASI

A. Metodologi Pengembangan Proyek Sistem Informasi

Metodologi Pengembangan Proyek Sistem Informasi (SDLC - *System Development Life Cycle*) adalah kerangka kerja yang digunakan untuk merancang, mengembangkan, mengimplementasikan, dan memelihara sistem informasi. Metodologi ini membantu mengatur langkah-langkah yang diperlukan dalam setiap fase pengembangan proyek, mulai dari perencanaan awal hingga penyelesaian dan pemeliharaan sistem.



Gambar 2. 1. System Development Life Cycle (SDLC)
(Santander 2018)

Pengembangan proyek sistem informasi adalah proses yang kompleks yang memerlukan pendekatan sistematis dan metodologi yang tepat. Metodologi pengembangan proyek sistem informasi bertujuan untuk mengatur langkah-langkah yang diperlukan dalam merancang, mengembangkan, dan mengimplementasikan sistem informasi yang efektif dan efisien.

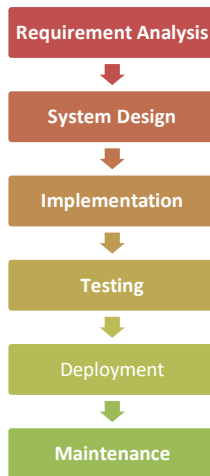
Kebanyakan kasus, pengembangan sistem informasi diperlukan untuk menyelesaikan masalah yang tidak dapat diselesaikan oleh sistem lama. Sebuah rumah sakit harus melakukan perombakan pada SIMRS (Sistem Informasi Manajemen Rumah Sakit) karena aplikasi sebelumnya tidak dapat melakukan bridging dengan BPJS karena pemerintah mewajibkannya. Akibatnya, pihak RS harus menyesuaikan SIMRS yang sudah mereka miliki.

Pengembangan sistem informasi, tim akan terdiri dari beberapa orang, seperti koordinator proyek (*project coordinator*), analis sistem dan desain (*system analyst and design*), desainer jaringan (*network designer*), *programmer*, teknisi hardware (*technician hardware*), *administrator*, penguji software (*software tester*), desain grafis (*graphic designer*), dan dokumentasi (*documenter*).

Terdapat beberapa pendekatan yang umum digunakan dalam metodologi pengembangan proyek sistem informasi, di antaranya:

1. Metodologi *Waterfall*

Metodologi ini mengikuti pendekatan linear dan berurutan, di mana setiap fase pengembangan sistem informasi dilakukan secara berurutan, mulai dari analisis kebutuhan, desain, pengembangan, pengujian, hingga implementasi. Setiap fase dimulai setelah fase sebelumnya selesai. Pendekatan ini cocok untuk proyek yang memiliki kebutuhan yang jelas dan stabil. Berikut adalah langkah-langkah yang umum terdapat dalam Metodologi *Waterfall*:



Gambar 2. 2. Metodologi *Waterfall*

Pada gambar di atas merupakan fase yang terjadi pada waterfall yaitu:

a) Analisis Kebutuhan

Tahap analisis kebutuhan merupakan langkah awal dalam Metodologi *Waterfall*. Pada tahap ini, tim pengembang akan berinteraksi dengan pemangku kepentingan (*stakeholders*) untuk mengidentifikasi dan memahami kebutuhan sistem yang akan dikembangkan. Proses ini melibatkan pengumpulan dan analisis kebutuhan pengguna, pemetaan kebutuhan ke dalam spesifikasi sistem, dan perumusan persyaratan proyek yang jelas.

b) Perancangan

Tahap perancangan bertujuan untuk merancang struktur dan arsitektur sistem yang akan dibangun. Pada tahap ini, tim pengembang akan membuat desain sistem secara rinci, termasuk desain basis data, desain antarmuka pengguna, dan desain komponen-komponen lainnya. Tujuan dari tahap perancangan adalah menciptakan panduan yang jelas bagi implementasi sistem.

c) Implementasi

Tahap implementasi melibatkan pengkodean atau pembangunan sistem berdasarkan desain yang telah dibuat. Tim pengembang akan melakukan pemrograman, pengujian unit, dan integrasi komponen-komponen sistem. Implementasi yang sukses

melibatkan penggunaan bahasa pemrograman dan alat pengembangan yang sesuai dengan kebutuhan proyek.

d) Pengujian

Tahap pengujian dilakukan untuk memastikan bahwa sistem yang dikembangkan berfungsi sesuai dengan spesifikasi dan memenuhi kebutuhan pengguna. Pengujian dapat meliputi pengujian fungsional, pengujian keamanan, pengujian performa, dan pengujian lainnya. Tujuan dari tahap ini adalah untuk mengidentifikasi dan memperbaiki cacat (*bug*) serta memastikan kualitas keseluruhan sistem.

e) Penerapan dan Penyampaian

Tahap ini melibatkan penerapan sistem yang telah diuji dan dinyatakan siap untuk digunakan ke dalam lingkungan produksi. Proses ini meliputi instalasi sistem, migrasi data, dan pelatihan pengguna. Pada tahap ini, sistem akan secara resmi diserahkan kepada pemangku kepentingan dan digunakan dalam operasional sehari-hari.

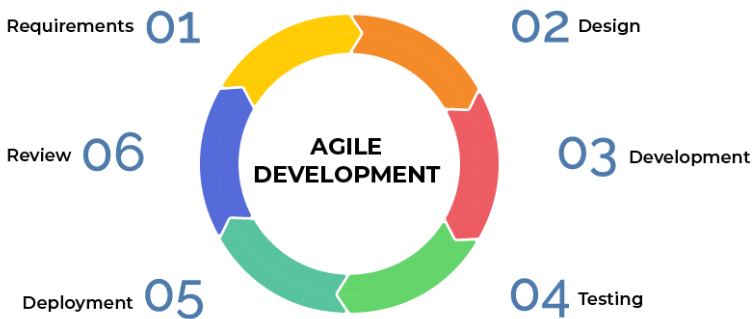
f) Pemeliharaan

Tahap pemeliharaan berfokus pada pemantauan, pemeliharaan, dan perbaikan sistem setelah implementasi. Tim pengembang akan mengatasi masalah atau kegagalan yang muncul, menjalankan pembaruan sistem, serta memberikan dukungan teknis kepada pengguna. Pemeliharaan yang efektif dapat meningkatkan kinerja sistem dan memperpanjang masa pakai sistem.

Metodologi *Waterfall* memiliki keunggulan dalam menyediakan tahapan yang terstruktur dan terurut secara linier, sehingga memudahkan pemahaman dan pengendalian proyek. Namun, pendekatan ini kurang fleksibel terhadap perubahan kebutuhan dan sulit untuk melakukan koreksi jika terdapat kesalahan pada tahap-tahap sebelumnya. Oleh karena itu, perlu diingat bahwa pemilihan metodologi pengembangan harus disesuaikan dengan kebutuhan dan karakteristik proyek yang sedang dilakukan.

2. Metodologi Agile

Metodologi ini menekankan fleksibilitas, kolaborasi tim dan responsif terhadap perubahan. Pada pendekatan ini, pengembangan sistem informasi dilakukan dalam iterasi pendek yang disebut “*sprint*”, di mana setiap sprint berfokus pada tujuan tertentu. Metodologi *Agile* yang populer adalah *Scrum* dan *Kanban*.



Gambar 2. 3. Agile SDLC Models

[Gambar Agile SDLC Models] Sumber: <https://agiletech.vn>

Tujuan Pengembangan metode *Agile*, terdiri dari tujuh kategori, diantaranya:

1. *High-value and working App system*

Tujuan pertama adalah menghasilkan perangkat lunak yang memiliki nilai jual tinggi dan dapat menekan biaya pembuatan. Yang terpenting adalah kemampuan untuk menghasilkan produk berkualitas tinggi.

2. *Iterative, incremental, evolutionary*

Model pengembangan *agile* berulang dan dapat diubah sesuai kebutuhan. Diakui bahwa pendekatan ini sangat fleksibel dan cocok untuk proyek pengembangan jangka pendek.

3. *Cost control and value-driven development*

Pengembangan perangkat lunak dapat disesuaikan dengan kebutuhan pengguna. Tim pengembang kemudian dapat

mengontrol biaya dan waktu yang diperlukan untuk proses pengembangan *software*.

4. *High-quality production*

Produk perangkat lunak tetap berkualitas tinggi meskipun jumlah biaya dan waktu yang diperlukan sedikit.

5. *Flexible and risk management*

Fleksibel berarti bahwa pertemuan dengan klien dapat dilakukan kapan saja untuk menjaga kinerja perangkat lunak. Terutama, dapat mengurangi kesalahan pada produk dan program sebelum proses penginstalan aplikasi dimulai.

6. *Collaboration*

Setiap tim pengembang bekerja sama untuk membahas masukan klien. Jadi, tim pengembang harus berkomunikasi dan bekerja sama dengan baik.

7. *Self-organizing and self-managing teams*

Tujuan terakhir dari pendekatan *Agile* ini adalah agar pengembang dapat mengelola proses pengembangan *software* secara mandiri. Sebagai cara untuk mengurangi kesalahan komunikasi, manajer bertanggung jawab untuk berinteraksi dengan developer dan klien.

Metode *Agile* memiliki 12 dasar dalam penggunaannya, yaitu:

1. Kepuasan pelanggan adalah prioritas utama.
2. Adaptif dengan perubahan untuk menciptakan keuntungan yang kompetitif bagi klien
3. Mengembangkan perangkat lunak secara konsisten dalam waktu yang singkat
4. Kerja sama tim berlangsung setiap waktu
5. Mengembangkan proyek yang memotivasi secara pribadi
6. Memberikan informasi melalui komunikasi langsung
7. Perangkat lunak yang berfungsi dengan baik merupakan ukuran kemajuan.
8. Pengembangan yang terus menerus
9. Proses, rancangan, dan teknik yang berkelanjutan
10. Memanfaatkan sumber daya yang tersedia sebaik mungkin

11. Anggota tim menciptakan desain, persyaratan, dan ide terbaik.
12. Refleksi berkala tentang tim untuk mengubah dan menyesuaikan cara kerja yang efisien dan efektif.

Tahapan-tahapan metode *Agile*

Dalam metode *Agile*, terdapat serangkaian langkah yang berbeda dan lebih spesifik. Beberapa langkah yang umum dalam metode *Agile* adalah:

1. Perencanaan Produk dan Pembuatan Daftar *Backlog*: Tim melakukan perencanaan produk dan membuat daftar backlog yang berisi fitur-fitur yang akan dikembangkan.
2. Perencanaan *Sprint*: Tim memilih item dari backlog dan merencanakan sprint berikutnya, termasuk menetapkan sasaran sprint dan memperkirakan waktu dan sumber daya yang dibutuhkan.
3. Pengembangan Iteratif: Tim bekerja dalam sprint, melakukan pengembangan perangkat lunak dalam iterasi pendek (biasanya 1-4 minggu) dengan fokus pada fitur-fitur yang telah dipilih.
4. Pengujian Iteratif: Selama pengembangan iteratif, pengujian terus dilakukan untuk memastikan kualitas perangkat lunak yang dikembangkan.
5. Review dan Retrospektif *Sprint*: Setelah selesai sebuah sprint, tim melakukan review untuk mengevaluasi hasil pekerjaan yang telah dilakukan, serta retrospektif untuk mengevaluasi proses dan melakukan perbaikan untuk sprint berikutnya.
6. Penyerahan Produk dan Umpan Balik: Produk yang telah dikembangkan dalam sprint diserahkan kepada pengguna atau pemangku kepentingan untuk mendapatkan umpan balik yang berharga untuk perbaikan selanjutnya.

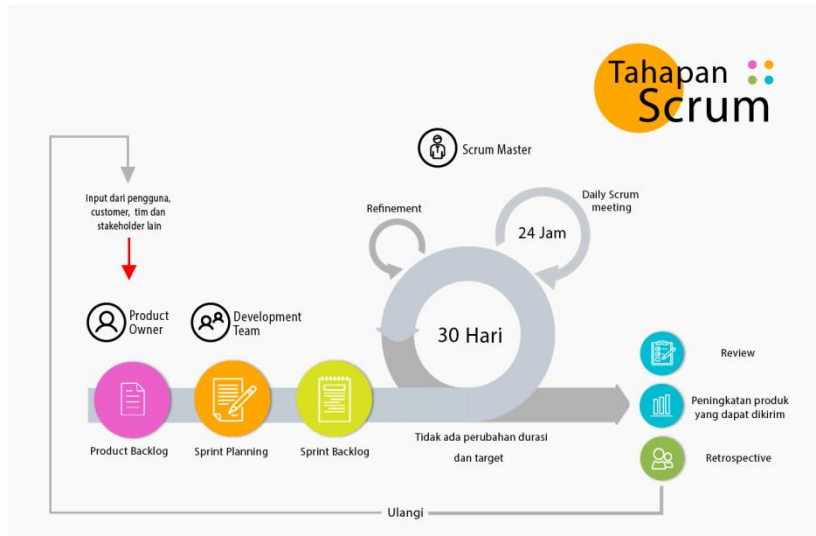
Tentu saja, langkah-langkah tersebut dapat bervariasi tergantung pada metode *Agile* yang digunakan, seperti *Scrum*,

Kanban, atau *XP*. Penting untuk mengadaptasi metode *Agile* sesuai kebutuhan proyek.

Ada beberapa jenis *Agile Development* yang umum digunakan dalam pengembangan perangkat lunak. Berikut ini beberapa jenis *Agile Development*, yaitu:

a. *Scrum*

Scrum adalah metode *Agile* yang paling populer dan banyak digunakan. Dalam *Scrum*, proyek dibagi menjadi serangkaian sprint yang berdurasi pendek (biasanya 1-4 minggu). Setiap sprint memiliki tujuan yang jelas dan menghasilkan deliverables yang dapat diuji. Tim bekerja dalam *tim-boxed iterations* dan melakukan pertemuan harian untuk sinkronisasi.



Gambar 2. 4. Tahapan pengembangan proyek metode *Scrum*
[Gambar Tahapan Metodologi *Scrum*] Sumber: agus-hermanto.com

Pada gambar di atas, metode *scrum* adalah sebuah kerangka kerja pengembangan perangkat lunak yang berfokus pada pengelolaan proyek secara adaptif dan kolaboratif. Tahapan dalam *Scrum* terdiri dari:

1. *Product Backlog*: Tahap ini melibatkan membuat daftar kebutuhan atau fitur yang harus dikembangkan dalam bentuk *Product Backlog*, berisi elemen-elemen yang

dinyatakan dalam bentuk *User Story*, yang mendeskripsikan kebutuhan dari perspektif pengguna.

2. *Sprint Planning*: Tahap ini melibatkan pertemuan antara Tim Pengembang dan Pemilik Produk (*Product Owner*) untuk merencanakan Sprint berikutnya. Tim Pengembang memilih sejumlah *User Story* dari *Product Backlog* yang akan diambil dalam Sprint berikutnya dan merancang tugas-tugas yang diperlukan untuk menyelesaikan *User Story* tersebut.
3. *Sprint*: Tahap ini merupakan periode waktu tetap di mana Tim Pengembang bekerja untuk menyelesaikan *User Story* yang dipilih dalam Sprint Planning. Sprint memiliki durasi yang terbatas, biasanya antara satu hingga empat minggu. Selama *Sprint*, Tim Pengembang melakukan *Stand-up Daily Meeting* untuk berkoordinasi, berbagi perkembangan, dan mengidentifikasi hambatan yang mungkin muncul.
4. *Daily Scrum*: Ini adalah pertemuan harian yang dilakukan oleh Tim Pengembang untuk berkoordinasi dan mengupdate perkembangan pekerjaan yang telah disampaikan dari 24 jam terakhir dan merencanakan pertemuan berikutnya. Setiap anggota tim menjawab tiga pertanyaan sederhana: “Apa yang sudah dilakukan sejak pertemuan sebelumnya?”, “Apa yang akan dilakukan hari ini?”, dan “Apakah ada hambatan yang menghalangi?”.
5. *Sprint Review*: Setelah selesai *Sprint*, dilakukan *Sprint Review* yang melibatkan Tim Pengembang, Pemilik Produk, dan pemangku kepentingan lainnya. Pada tahap ini, Tim Pengembang memperlihatkan hasil pekerjaan yang telah selesai kepada pemangku kepentingan dan memperoleh umpan balik. Hasil pekerjaan dapat berupa produk yang berfungsi atau fitur yang dikembangkan selama *Sprint*.
6. *Retrospektif Sprint*: Ini adalah pertemuan di akhir *Sprint*

yang dilakukan oleh Tim Pengembang untuk mengevaluasi kinerja diri sendiri dan mencari cara untuk meningkatkan efektivitas pengembangan ke depan. Pada tahap ini, adalah mengidentifikasi apa yang telah berjalan dengan baik, apa yang harus diperbaiki, dan mengambil tindakan perbaikan untuk *Sprint* berikutnya.

Tahapan-tahapan ini berulang dalam siklus pengembangan perangkat lunak menggunakan pendekatan *Scrum*. Dengan setiap *Sprint*, proyek secara bertahap membangun dan meningkatkan produk yang dikembangkan.

Terdapat tiga peran dan tanggung jawab di dalam metode *Scrum*: *Scrum Master*, Pemilik Produk, dan Tim

1. *Scrum Master* memiliki tugas pertama untuk mengatur pertemuan selama proses *Scrum*. Tugas kedua adalah mengatur dan memantau tim untuk menghindari masalah selama proses pengembangan. Tugas ketiga adalah memastikan bahwa tim pengembangan menerapkan praktik dan prinsip *Scrum*. Tugas terakhir dari *Scrum Master* adalah memastikan bahwa tidak ada item tambahan selama kegiatan *sprint*.
2. *Product Owner* atau pemilik produk adalah orang yang menggunakan kemampuan bisnisnya untuk memprioritaskan item tertentu dalam *backlog* produk. Pemilik produk bertindak sebagai perwakilan perusahaan dan bertanggung jawab untuk menghasilkan nilai terbaik dari pekerjaan.
3. *Development Team* atau Tim Pengembangan bertanggung jawab untuk menghasilkan produk kerja. Tim ini harus bertindak cepat dalam menyelesaikan pekerjaan, oleh karena itu disarankan untuk membentuk tim dengan tiga hingga sembilan anggota. Aspek penting dari tim pengembangan adalah kemampuan mereka untuk bekerja lintas fungsi, mengatur diri sendiri, dan tidak memiliki hirarki.

b. *Kanban*

Kanban adalah metode *Agile* yang fokus pada visualisasi aliran kerja. Pada papan *Kanban*, tugas ditempatkan dalam kolom yang merepresentasikan statusnya, seperti *"To Do"*, *"In Progress"*, dan *"Done"*. Tim mengambil tugas baru saat mereka selesai dengan yang sedang mereka kerjakan. Pendekatan ini membantu mengoptimalkan aliran kerja dan mengidentifikasi hambatan atau *bottleneck*.

Kanban adalah teknik pengelolaan aliran kerja yang didasarkan pada visualisasi, pengurangan jumlah pekerjaan yang perlu dilakukan, dan adaptasi terhadap perubahan. Berikut ini adalah prosedur umum yang digunakan oleh program *Kanban*:

- Visualisasi Aliran Kerja: Pertama, tim menggunakan papan *Kanban* untuk menunjukkan aliran kerja mereka. Papan ini terdiri dari kolom yang menunjukkan langkah-langkah dalam proses, seperti *"To Do"*, *"In Progress"*, *"Review/QA"*, dan *"Done"*.
- Pembatasan *Work in Progress* (WIP): Pada papan *Kanban*, setiap kolom menunjukkan jumlah pekerjaan yang dapat dilakukan secara bersamaan. Ini disebut sebagai batasan WIP dan dimaksudkan untuk mencegah penumpukan pekerjaan pada tahap tertentu, mendorong tim untuk menyelesaikan pekerjaan yang sudah ada sebelum menambahkan yang baru.
- Tugas dalam Antrian: Setiap anggota tim dapat mengambil tugas yang tersedia dari antrian sesuai kapasitas dan keahlian mereka. Kolom *"To Do"* menunjukkan tugas-tugas yang belum dimulai atau yang belum diprioritaskan.
- Pindah Tahap: Tugas dipindahkan ke kolom *"In Progress"* ketika anggota tim mulai mengerjakannya. Setiap tugas harus memiliki pemilik yang bertanggung jawab untuk menyelesaikannya.
- *Review* dan *QA*: Setelah pekerjaan selesai, pekerjaan