

SPEKIFIKASI KEBUTUHAN PERANGKAT LUNAK

<QuickTurn>

untuk:

<nama pelanggan>

Dipersiapkan oleh:


<Nama Pelaksana Proyek>

Program Studi S1 Informatika

Fakultas Informatika

Universitas Telkom

2025

 UNIVERSITAS Telkom	Program Studi S1 Informatika - Fakultas Informatika	SKPL - Nomor Dokumen		Halaman
		Revisi	<i><nomor revisi></i>	<i>Tgl: <isi tanggal></i>

1. Daftar Perubahan

Revisi	Deskripsi
A	
B	
C	
D	
E	
F	
G	

INDEX	-	A	B	C	D	E	F	G
TGL								
Ditulis oleh								
Diperiksa oleh								
Disetujui oleh								

2. Daftar Halaman Perubahan

Halaman	Revisi	Halaman	Revisi

3. Daftar Isi

Daftar Perubahan	1
Daftar Halaman Perubahan	2
Daftar Isi	3
1. Pendahuluan	4
1.1 Tujuan Penulisan Dokumen	4
1.2 Ruang Lingkup / Cakupan Dokumen	4
1.3 Definisi, Singkatan, dan Akronim	4
1.4 Referensi	4
2. Deskripsi Global Perangkat Lunak	5
2.1 Statement of Objective Perangkat Lunak	5
2.2 Perspesktif dan Goal Perangkat Lunak	5
2.3 Profil dan Kelas Pengguna	
2.4 Lingkungan Operasi	5
2.5 Batasan Perangkat Lunak / Sistem	5
2.6 Asumsi dan Dependensi	6
3. Deskrpsi Rinci Perangkat Lunak	7
3.1 Deskripsi Kebutuhan	7
3.1.1 Kebutuhan Fungsional	7
3.1.2 Kebutuhan Non-Fungsional	7
3.2 Pemodelan Analisis	7
3.2.1 Usecase Diagram	7
3.2.2 Class Diagram:	8
4. Kebutuhan Lain - Lain	9
4.1 Antarmuka Pengguna	9
4.2 Antarmuka Perangkat Keras	9
4.3 Antarmuka Perangkat Lunak	9
4.4 Antarmuka Komunikasi	9
4.5 Sistem Cerdas	

1. Pendahuluan

1.1 Tujuan Penulisan Dokumen

Dokumen Spesifikasi Kebutuhan Perangkat Lunak (SKPL) ini bertujuan menjelaskan secara rinci kebutuhan perangkat lunak yang sedang di kembangkan. Penjabaran spesifikasi perangkat lunak meliputi deskripsi umum perangkat lunak dan deskripsi kebutuhan perangkat lunak. SKPL ini dapat digunakan sebagai acuan dalam setiap tahapan pengembangan perangkat lunak agar sesuai dengan kebutuhan yang diinginkan oleh pengguna dan tujuan perangkat lunak itu sendiri.

1.2 Ruang Lingkup / Cakupan Dokumen

Dokumen SKPL ini mengikuti beberapa standar dan konvensi penulisan untuk memudahkan pemahaman dalam penyampaian informasi. Standar penulisan yang digunakan mencakup font yang digunakan adalah Times New Roman 12 untuk isian utama dokumen dan Times New Roman 14 tebal untuk judul utama. Penyorotan italic digunakan untuk penekanan istilah teknis atau istilah penting.

1.3 Definisi, Singkatan, dan Akronim

Dalam penyusunan dokumen SKPL ini, beberapa istilah, singkatan, dan akronim digunakan untuk mempermudah pemahaman. Adapun singkatan dan Akronim yang terdapat dalam dokumen ini sebagai berikut :

1.4 Referensi

<Buat daftar dokumen lain atau alamat Web yang diacu SKPL ini. Termasuk panduan gaya antarmuka pengguna, kontrak, standar, sistem spesifikasi requirements, dokumen use case, atau dokumen visi dan ruang lingkup. Berikan informasi yang cukup sehingga pembaca bisa mengakses salinan setiap referensi, termasuk judul, penulis, nomor versi, tanggal, dan sumber atau lokasi.>

2. Deskripsi Global Perangkat Lunak

2.1 Statement of Objective Perangkat Lunak

QuickTurn adalah aplikasi berbasis web dan mobile yang berfungsi menghubungkan mahasiswa dan UMKM melalui sistem proyek digital (micro-internship). Sistem mendukung seluruh siklus proyek: mulai dari pendaftaran, posting proyek, pencarian & matching, komunikasi, kontrak digital, hingga pembayaran dan review.

2.2 Perspektif dan Goal Perangkat Lunak

Pembangunan perangkat lunak **QuickTurn** bertujuan untuk menciptakan sebuah platform digital yang menjembatani kolaborasi antara **Mahasiswa** dan **UMKM** dalam pengerjaan proyek secara efektif, aman, dan transparan. Secara umum, tujuan utama sistem ini adalah membantu UMKM mendapatkan sumber daya manusia berkualitas dengan cara yang efisien, sekaligus memberi kesempatan bagi mahasiswa untuk memperoleh pengalaman kerja nyata dan penghasilan yang terjamin.

Secara operasional, sistem QuickTurn dirancang untuk memfasilitasi proses interaksi antara pihak-pihak yang terlibat, mulai dari pendaftaran pengguna, publikasi proyek, pencarian dan seleksi mitra kerja, pelaksanaan proyek bersama, hingga penyelesaian dan evaluasi hasil kerja. Selain itu, sistem juga bertujuan menjaga keamanan data, kelancaran komunikasi, serta memastikan seluruh transaksi dan aktivitas berlangsung secara transparan dan dapat dipertanggungjawabkan. Dengan demikian, pemodelan goal QuickTurn berfokus pada terciptanya ekosistem kolaboratif yang saling menguntungkan, efisien, dan terpercaya bagi seluruh penggunanya.

2.3 Profil dan Kelas Pengguna

1. Mahasiswa (User Role: Talent)

Profil:

Mahasiswa merupakan pengguna utama yang menggunakan sistem untuk mencari, melamar, dan mengerjakan proyek dari UMKM. Mereka memiliki motivasi untuk memperoleh pengalaman kerja, membangun portofolio profesional, dan mendapatkan kompensasi yang aman. Umumnya memiliki kemampuan teknis dasar hingga menengah serta terbiasa menggunakan platform digital.

Karakteristik:

- Frekuensi penggunaan: tinggi (aktif dalam pencarian dan pengerjaan proyek).

- Bagian sistem yang digunakan: pencarian proyek, pengajuan proposal, komunikasi (chat), pelacakan status proyek, dan penarikan pembayaran.
- Tingkat keahlian teknis: menengah.
- Tingkat hak akses: pengguna umum dengan akses terbatas sesuai peran mahasiswa.

Kebutuhan terkait:

- Antarmuka yang mudah dipahami dan digunakan.
- Fitur portofolio, pengelolaan profil, serta pelacakan status proyek.
- Keamanan data pribadi dan transaksi pembayaran.

2. UMKM (User Role: Project Owner)

Profil:

UMKM merupakan pihak yang membuat dan mempublikasikan proyek serta mencari mahasiswa yang sesuai dengan kebutuhan bisnisnya. Pengguna dari kelas ini berasal dari latar belakang non-teknis, sehingga sistem harus mudah digunakan dalam proses pembuatan proyek dan pengelolaan pembayaran.

Karakteristik:

- Frekuensi penggunaan: sedang hingga tinggi (tergantung kebutuhan proyek).
- Bagian sistem yang digunakan: pembuatan proyek, seleksi pelamar, komunikasi, dan pembayaran.
- Tingkat keahlian teknis: dasar hingga menengah.
- Tingkat hak akses: pengguna umum dengan kontrol terhadap proyek yang dibuat.

Kebutuhan terkait:

- Kemudahan dalam pembuatan dan pengelolaan proyek.
- Fitur seleksi kandidat dan manajemen proyek.
- Sistem pembayaran yang aman dan terintegrasi.

3. Admin (User Role: System Administrator)

Profil:

Admin merupakan pihak yang bertanggung jawab dalam mengawasi aktivitas sistem dan pengguna. Mereka memastikan sistem berjalan dengan baik, aman, serta bebas dari aktivitas yang melanggar aturan. Admin memiliki hak akses tertinggi dan kemampuan teknis untuk melakukan moderasi, verifikasi, serta pelaporan.

Karakteristik:

- Frekuensi penggunaan: tinggi (pengawasan sistem dilakukan secara rutin).
- Bagian sistem yang digunakan: dashboard admin, moderasi laporan, verifikasi akun UMKM, analisis transaksi, dan manajemen konten.
- Tingkat keahlian teknis: tinggi.
- Tingkat hak akses: penuh terhadap seluruh modul sistem.

Kebutuhan terkait:

- Fitur verifikasi pengguna dan proyek.
- Sistem pelaporan, audit log, dan monitoring aktivitas.
- Kemampuan untuk mengekspor data dan melakukan tindakan administratif.

4. Pengguna Eksternal Pendukung (User Role: Service Integrator)**Profil:**

Terdiri dari pihak ketiga seperti penyedia layanan pembayaran (payment gateway) dan penyedia layanan OTP. Mereka tidak berinteraksi langsung dengan antarmuka pengguna, melainkan melalui integrasi sistem dan API.

Karakteristik:

- Frekuensi penggunaan: otomatis berdasarkan event tertentu.
- Bagian sistem yang digunakan: layanan transaksi dan autentikasi.
- Tingkat keahlian teknis: berbasis sistem, tanpa interaksi manual.
- Tingkat hak akses: terbatas pada endpoint API yang relevan.

Kebutuhan terkait:

- Keamanan integrasi API (token, enkripsi, validasi).
- Reliabilitas komunikasi antar sistem dan dokumentasi teknis yang jelas.

2.4 Lingkungan Operasi

< Gambarkan lingkungan di mana perangkat lunak ini akan beroperasi, termasuk platform perangkat keras, sistem operasi beserta versinya, dan setiap komponen perangkat lunak lain atau aplikasi lain yang berjalan bersamaan. Sebuah diagram sederhana yang menunjukkan komponen utama dari sistem secara keseluruhan, interkoneksi subsistem, dan antarmuka eksternal (API) dapat membantu menjelaskan bagian ini. Jika SKPL mendefinisikan komponen dari sistem yang lebih besar, hubungkan requirements sistem yang lebih besar dengan fungsi dari perangkat lunak ini dan identifikasi antarmuka (API) antara keduanya >

2.5 Batasan Perangkat Lunak / Sistem

< Jelaskan setiap item atau masalah yang akan membatasi fungsi yang tersedia untuk para developer. Hal yang mungkin termasuk dalam bagian ini antara lain: kebijakan atau peraturan perusahaan; keterbatasan perangkat keras (requirements waktu, requirements memori); antarmuka untuk aplikasi lain; teknologi tertentu, alat, dan database yang akan digunakan; operasi paralel; requirements bahasa; protokol komunikasi; pertimbangan keamanan; konvensi desain atau standar pemrograman (misalnya, jika organisasi pengguna akan bertanggung jawab untuk melakukan maintenance untuk perangkat lunak yang akan dikirimkan).>

2.6 Asumsi dan Dependensi

< Buatlah daftar faktor-faktor yang diasumsikan (sebagai lawan dari fakta yang telah diketahui) yang dapat mempengaruhi requirements dalam SKPL ini. Hal ini dapat mencakup pihak ketiga atau komponen komersial yang direncanakan untuk digunakan, isu seputar development atau lingkungan operasi, atau kendala yang akan dihadapi.

Proyek ini bisa terpengaruh jika asumsi ini tidak benar, tidak disebarluaskan, atau berubah. Juga identifikasi dependensi yang dimiliki proyek pada faktor-faktor eksternal, seperti komponen perangkat lunak yang Anda berniat untuk gunakan kembali dari proyek lain, kecuali jika komponen tersebut sudah didokumentasikan di tempat lain (misalnya, dalam dokumen visi dan ruang lingkup atau rencana proyek).>

3. Deskripsi Rinci Perangkat Lunak

<Template ini menggambarkan cara mengatur requirements fungsional untuk produk berdasar fitur sistem, layanan utama yang disediakan oleh produk. Anda dapat memilih untuk mengatur bagian ini dengan use case, mode operasi, kelas pengguna, kelas objek, hirarki fungsional, atau kombinasi dari itu semua, yang dapat membuat artian yang paling logis untuk produk Anda. Anda harus menggunakan use-case diagram>

3.1 Deskripsi Kebutuhan

3.1.1 Kebutuhan Fungsional

No.	Kode Kebutuhan	Deskripsi	Nama Kebutuhan
1.	FR-01	User (Mahasiswa/UMKM/Admin) dapat melakukan registrasi dan login menggunakan sistem berbasis JWT	Registrasi & Login
2.	FR-02	Sistem membedakan hak akses berdasarkan role (Mahasiswa, UMKM, Admin).	Role Management
3.	FR-03	User dapat mengubah profil (bio, skill, portofolio mahasiswa / profil bisnis UMKM).	Profile Management
4.	FR-04	UMKM dapat membuat posting project (judul, deskripsi, kategori, budget, deadline).	Posting Project
5.	FR-05	Mahasiswa dapat mencari dan memfilter project berdasarkan skill, kategori, budget.	Browsing & Search Project
6.	FR-06	Mahasiswa dapat mengajukan proposal untuk mengikuti project.	Apply Project
7.	FR-07	Sistem mengelola alur status project mulai dari OPEN, kemudian berubah menjadi ONGOING, dilanjutkan DONE, dan akhirnya CLOSED.	Project Status Flow
8.	FR-08	UMKM dapat memilih mahasiswa untuk project yang diposting.	Matching
9.	FR-09	Sistem menghasilkan kontrak digital sederhana ketika mahasiswa dipilih.	Kontrak Digital

10.	FR-10	User dapat memberikan rating dan review terhadap UMKM setelah selesai magang/kerja.	Reviews & Ratings
11.	FR-11	Admin dapat melakukan verifikasi UMKM, memoderasi users/jobs/reports, serta memantau aktivitas.	Admin Moderation
12.	FR-12	Admin dapat melihat dashboard metrik.	Reporting & Analytics
13.	FR-13	Mahasiswa & UMKM dapat berkomunikasi melalui chat.	Realtime Chat
14.	FR-14	User dapat mengunggah file (gambar, dokumen) di ruang chat.	Attachment Sharing
15.	FR-15	Sistem terhubung dengan payment gateway (Midtrans, Xendit, GoPay, DANA).	Payment Integration
16.	FR-16	Sistem menahan dana pembayaran hingga project selesai.	Escrow System
17.	FR-17	Admin dapat memantau transaksi dan membuat laporan.	Admin Monitoring
18.	FR-18	User dapat melakukan verifikasi email/nomor HP serta reset password via OTP. Masuk ke NFR	Email & OTP Verification
19.	FR-19	User dapat mengajukan dispute dan meminta mediasi dari admin.	Dispute & Mediation System
20.	FR-20	Sistem menyimpan versi file dan menyediakan preview (gambar/PDF).	File Versioning & Preview
21.	FR-21	Sistem menangani refund atau chargeback bila ada pembatalan project.	Refund & Chargeback Handling
22.	FR-22	Sistem menghasilkan invoice dan kwitansi dalam format PDF.	Invoicing & Receipt Generation
23.	FR-23	Mahasiswa dapat menarik saldo ke rekening bank/e-wallet.	Payout & Withdrawal System
24.	FR-24	User dapat melaporkan konten, project, atau user yang melanggar.	Report & Flag System

25.	FR-25	Sistem mengirim notifikasi (push/email/in-app) untuk event penting.	Notification Center
26.	FR-26	Mahasiswa dapat melihat statistik profil (views, acceptance rate).	Talent Insights Dashboard
27.	FR-27	UMKM dapat melihat performa posting (jumlah aplikasi, completion rate).	UMKM Insights Dashboard
28	FR-28	Admin dapat membuat pengumuman global atau banner sistem.	System Announcement
29.	FR-29	Admin dapat mengekspor laporan (CSV/PDF) terkait transaksi, project, dan user.	Data Export & Reporting

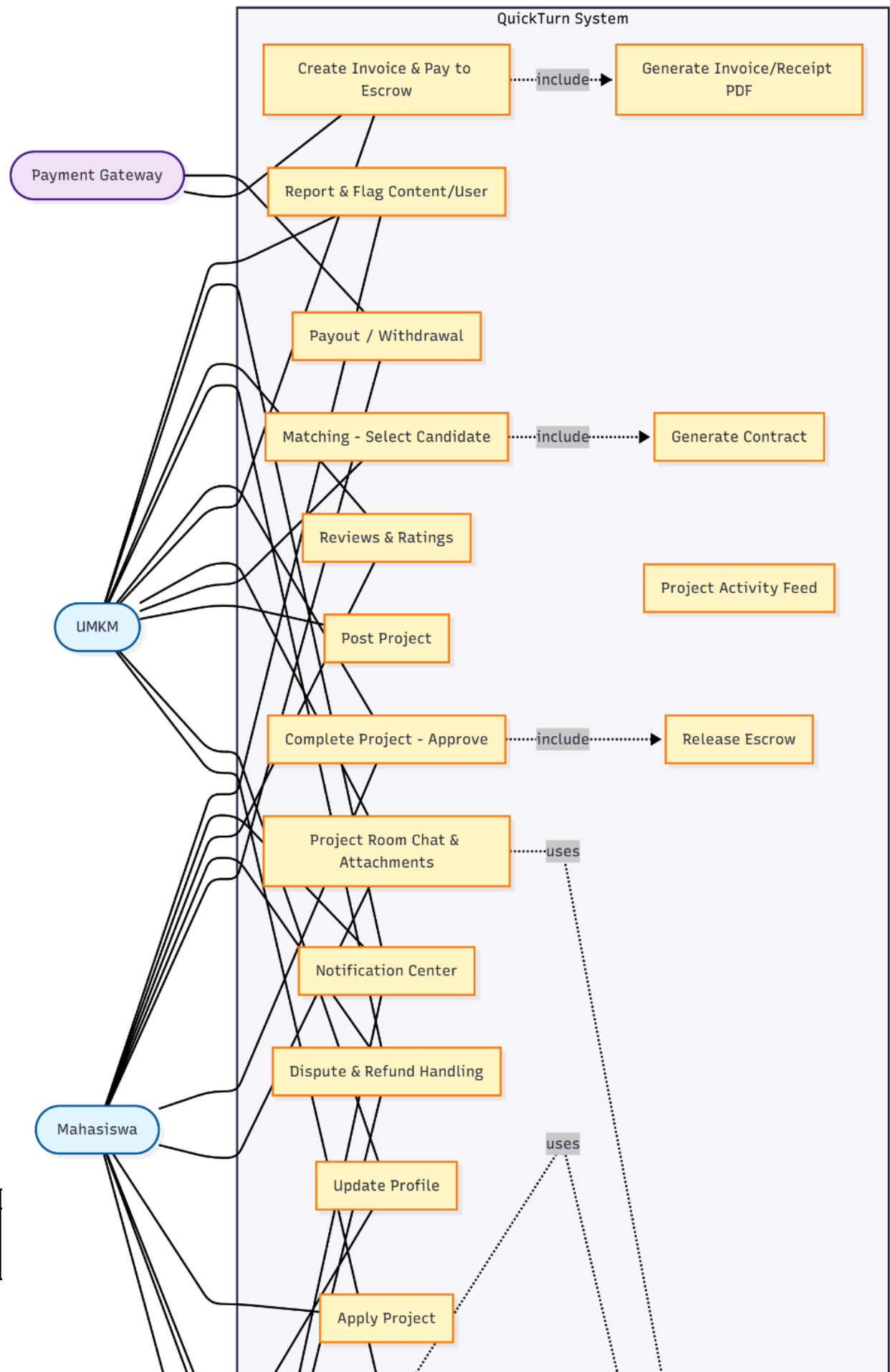
3.1.2 Kebutuhan Non-Fungsional

No.	Quality Criteria	Kode Kebutuhan	Deskripsi
1.	Usability	NFR-01	Aplikasi memiliki antarmuka yang mudah dipahami pengguna baru. Diukur melalui uji coba 10 mahasiswa/UMKM dengan target minimal 70% merasa aplikasi mudah digunakan.
2.	Performance	NFR-02	Waktu respon halaman utama, login, dan pencarian project tidak lebih dari 5 detik pada 90% kasus
3.	Availability	NFR-03	Sistem tersedia minimal 99% uptime dalam sebulan.
4.	Security	NFR-04	Sistem login menggunakan JWT dan password disimpan dengan hashing.
5.	Compatibility	NFR-05	Aplikasi dapat dijalankan di browser utama (Chrome, Firefox, Edge) dan responsive di layar laptop serta smartphone.
6.	Maintainability	NFR-06	Kode program terdokumentasi (README, komentar) agar bisa dipahami oleh anggota tim lain

--	--	--	--

3.2 Pemodelan Analisis

3.2.1 *Usecase Diagram*



3.2.1.1 Usecase Scenario #1 “REGISTER & Login”

Nama Use Case	Register & Login	
Deskripsi	User (Mahasiswa/UMKM/Admin) dapat melakukan registrasi dan login menggunakan sistem berbasis JWT.	
Pre-Kondisi	User belum login	
Post-Kondisi	User berhasil login dan diarahkan ke halaman dashboard sesuai role masing-masing (Mahasiswa/UMKM/Admin).	
Skenario Utama		
	Aktor	Sistem
	1. Membuka menu Registrasi/Login	
		2. Menampilkan form registrasi/login
	3. Mengisi data akun	
		4. Memvalidasi input data
	5. Menekan tombol "Register" atau "Login"	
		6. -Jika register: menyimpan data user baru ke database - Jika login: mengecek kredensial user
		7. Jika berhasil, sistem menghasilkan JWT Token
		8. Menampilkan dashboard sesuai role user
Skenario Eksepsional (Alternative flow)		
	Aktor	Sistem
	1. Mengisi data registrasi tidak valid (email sudah	

	terdaftar / format salah)	
		2. Menampilkan pesan error “Registrasi gagal”
	3. Mengisi data login salah (email/password salah)	
		4. Menampilkan pesan error “Login gagal, periksa kembali email dan password”
	5. Tidak ada koneksi internet atau server down	
		6. Menampilkan pesan “Koneksi gagal, coba lagi nanti.”

3.2.1.2 Usecase Scenario #2 <Role Management>

Nama Usecase	Role Management	
Deskripsi	Sistem membedakan hak akses berdasarkan role (Mahasiswa, UMKM, Admin).	
Pre-Kondisi	User telah melakukan login ke dalam sistem	
Post-Kondisi	User diarahkan ke dashboard atau menu sesuai hak akses role masing-masing	
Skenario Utama		
	Aktor	Sistem
	1. User login ke sistem	
		2. Mengecek role user
		3. Menentukan hak akses berdasarkan role (Mahasiswa,

		UMKM, Admin)
		4. Menampilkan menu dan fitur yang sesuai dengan role user
		5. Membatasi akses ke fitur yang tidak diperbolehkan
Skenario Eksepsional (Alternative flow)		
	Aktor	Sistem
	1. User mencoba mengakses fitur yang tidak sesuai role	
		2. Menampilkan pesan “Akses ditolak”

3.2.1.3 Usecase Scenario #3 <Profile Management>

Nama Use Case	Profile Management	
Deskripsi	User dapat mengubah profil (bio, skill, portofolio mahasiswa / profil bisnis UMKM).	
Pre-Kondisi	User telah login ke dalam sistem.	
Post-Kondisi	Data profil user berhasil diperbarui dan tersimpan di database.	
Skenario Utama		
	Aktor	Sistem
	1. User membuka menu “Profil”	
		2. Menampilkan halaman profil beserta tombol “Edit”
	3. User menekan tombol “Edit”	

		4. Menampilkan form untuk mengubah data profil
	5. User mengubah data (bio, skill, portofolio/profil bisnis)	
		6. Memvalidasi input data
	7. User menekan tombol “Simpan”	
		8. Menyimpan perubahan ke database
		9. Menampilkan pesan “Profil berhasil diperbarui”
Skenario Eksepsional (Alternative flow)		
	Aktor	Sistem
	1. User mengisi data tidak valid	
		2. Menampilkan pesan error “Data tidak valid.”

3.2.1.4 Usecase Scenario #4 <POSTING PROJECT>

Nama Usecase	Posting Project	
Deskripsi	UMKM dapat membuat dan mempublikasikan project baru dengan mengisi data seperti judul, deskripsi, kategori, budget, dan deadline.	
Pre-Kondisi	UMKM sudah login ke sistem dan memiliki profil bisnis yang lengkap. User telah melakukan login ke dalam sistem	
Post-Kondisi	Project baru tersimpan di database dengan status OPEN dan muncul di halaman daftar project yang dapat dilihat mahasiswa.	
Skenario Utama	Aktor	Sistem

	1. Membuka menu “Post Project” pada dashboard UMKM.	
		2. Menampilkan form input project (judul, deskripsi, kategori, budget, deadline).
	3. Mengisi seluruh data project dengan lengkap.	
		4. Memvalidasi semua input (panjang teks, format angka, tanggal deadline > tanggal hari ini).
	5. Menekan tombol “Submit” .	
		6. Menyimpan data project ke database dengan status awal OPEN .
		7. Menampilkan notifikasi “Project berhasil diposting.”
		8. Menampilkan halaman detail project yang baru dibuat.
Skenario Eksepsional (Alternative flow)	Aktor	Sistem
	1. Mengisi data project tidak lengkap atau format salah (contoh: budget kosong, tanggal melewati hari ini).	
		2. Menampilkan pesan error “Data tidak valid, periksa kembali isian Anda.” dan menandai field yang salah.
	3. Mengirim form saat koneksi terputus.	
		4. Menampilkan pesan “Gagal menyimpan project, periksa koneksi Anda.”
	5. Menutup halaman tanpa menekan tombol “Submit.”	

		6. Sistem tidak menyimpan data project apa pun.
--	--	---

3.2.1.5 Usecase Scenario #5 <BROWSE & SEARCH PROJECT>

Nama Usecase	Browse & Search Project	
Deskripsi	Mahasiswa dapat menelusuri dan memfilter daftar project berdasarkan kategori, skill, budget, dan keyword untuk menemukan project yang sesuai.	
Pre-Kondisi	Mahasiswa sudah login ke sistem.	
Post-Kondisi	Sistem menampilkan daftar project sesuai filter pencarian yang dipilih oleh mahasiswa.	
Skenario Utama	Aktor	Sistem
	1. Membuka menu “Explore Projects”.	
		2. Menampilkan daftar project yang sedang berstatus OPEN ..
	3. Memasukkan keyword atau memilih filter (kategori, skill, budget range, deadline).	
		4. Memproses filter pencarian berdasarkan input pengguna.
	5. Menekan tombol “Search” atau “Apply Filter”.	
		6. Mengambil data project dari database yang sesuai dengan kriteria pencarian.
		7. Menampilkan daftar hasil pencarian kepada pengguna.
		8. Menyediakan opsi untuk mengurutkan hasil (misalnya: terbaru, budget tertinggi, deadline terdekat).
Skenario Eksepsional (Alternative flow)	Aktor	Sistem
	1. Mengisi filter dengan kombinasi yang tidak	

	menghasilkan project apa pun.	
		2. Menampilkan pesan “Tidak ada project yang cocok dengan pencarian Anda.”
	3. Menekan tombol “Search” tanpa koneksi internet.	
		4. Menampilkan pesan “Gagal memuat data, periksa koneksi Anda.”
	5. Menginput nilai filter tidak valid (contoh: budget minimum lebih besar dari maksimum).	
		6. Menampilkan pesan error “Filter tidak valid, harap periksa kembali.”

3.2.1.6 Usecase Scenario #6 <APPLY PROJECT>

Nama Usecase	Apply Project	
Deskripsi	Mahasiswa dapat mengajukan proposal untuk mengikuti project yang berstatus OPEN , dengan menuliskan penjelasan singkat mengenai kemampuan dan estimasi pengerjaan.	
Pre-Kondisi	Mahasiswa sudah login dan belum pernah mengajukan proposal untuk project yang sama.	
Post-Kondisi	Proposal tersimpan di database dengan status awal SUBMITTED/PENDING , dan UMKM penerima project mendapatkan notifikasi pengajuan baru.	
Skenario Utama	Aktor	Sistem
	1. Membuka halaman detail project yang berstatus OPEN .	
		2. Menampilkan informasi lengkap project (judul,

		deskripsi, kategori, budget, deadline).
	3. Menekan tombol “Apply Project” .	
		4. Menampilkan form pengajuan yang berisi kolom proposal, estimasi durasi, dan expected fee.
	5. Mengisi data proposal secara lengkap lalu menekan tombol “Submit Application” .	
		6. Memvalidasi input (tidak boleh kosong, panjang maksimal teks, dll).
		7. Menyimpan data aplikasi ke database dengan status awal SUBMITTED .
		8. Mengirim notifikasi ke UMKM bahwa ada mahasiswa yang melamar project tersebut.
		9. Menampilkan pesan konfirmasi “Lamaran Anda telah berhasil dikirim.” kepada mahasiswa.
Skenario Eksepsional (Alternative flow)	Aktor	Sistem
	1. Mengajukan proposal pada project yang sama lebih dari sekali.	
		2. Menampilkan pesan error “Anda sudah mengajukan proposal untuk project ini.”
	3. Mengirim form tanpa mengisi kolom proposal.	
		4. Menampilkan pesan error “Kolom proposal wajib diisi.”

	5. Mengirim data saat koneksi ke server terputus.	
		6. Menampilkan pesan “Gagal mengirim, periksa koneksi Anda dan coba lagi.”
	7. Project telah berubah status menjadi ONGOING/CLOSED saat proses pengiriman.	
		8. Menampilkan pesan “Project tidak lagi menerima aplikasi.”

3.2.1.7 Usecase Scenario #7 <MATCHING & CONTRACT CREATION>

Nama Usecase	Matching & Contract Creation	
Deskripsi	UMKM dapat memilih salah satu mahasiswa dari daftar pelamar untuk mengerjakan project. Setelah mahasiswa dipilih, sistem otomatis membuat kontrak digital sederhana sebagai perjanjian kerja sama antara kedua pihak.	
Pre-Kondisi	Project memiliki minimal satu proposal dari mahasiswa, dan status project masih OPEN .	
Post-Kondisi	Mahasiswa terpilih menjadi talent project; status aplikasi berubah menjadi APPROVED , status project menjadi ONGOING , dan kontrak digital DRAFT tercipta di sistem.	
Skenario Utama	Aktor	Sistem
	1. UMKM membuka halaman daftar pelamar (application list) pada project.	
		2. Menampilkan seluruh daftar mahasiswa yang melamar dengan data profil & proposal mereka.
	3. UMKM meninjau detail profil mahasiswa dan proposal yang dikirim.	

		4. Menyediakan tombol “Pilih Mahasiswa” pada setiap pelamar.
	5. UMKM menekan tombol “Pilih Mahasiswa” pada kandidat yang diinginkan.	
		6. Memvalidasi apakah project masih berstatus OPEN dan belum memiliki kandidat terpilih.
		7. Mengubah status pelamar terpilih menjadi APPROVED , dan yang lain menjadi REJECTED .
		8. Membuat entitas Kontrak Digital (status awal: DRAFT) berisi detail project, durasi, dan pihak-pihak yang terlibat.
		9. Mengirim notifikasi ke mahasiswa yang terpilih dan ke UMKM bahwa kontrak telah dibuat.
		10. Menampilkan pesan konfirmasi “Mahasiswa berhasil dipilih dan kontrak telah dibuat.”
Skenario Eksepsional (Alternative flow)	Aktor	Sistem
	1. UMKM mencoba memilih lebih dari satu mahasiswa untuk project yang sama.	
		2. Menampilkan pesan error “Hanya satu mahasiswa yang dapat dipilih untuk project ini.”
	3. UMKM menekan “Pilih Mahasiswa” saat status project sudah ONGOING atau CLOSED .	
		4. Menampilkan pesan “Project ini sudah tidak

		dapat menerima kandidat baru.”
	5. Koneksi internet terputus setelah UMKM menekan tombol “Pilih Mahasiswa.”	
		6. Menampilkan pesan “Gagal memproses permintaan, periksa koneksi Anda dan coba lagi.”
	7. Sistem gagal membuat kontrak karena error server/database.	
		8. Menampilkan pesan “Terjadi kesalahan saat membuat kontrak, silakan coba ulang.”

3.2.1.8

Nama Use Case

Realtime Chat & Attachment

Deskripsi Mahasiswa dan UMKM dapat berkomunikasi melalui fitur chat realtime dalam project room, serta mengunggah file seperti gambar atau dokumen pendukung selama kolaborasi berlangsung.

Pre-Kondisi Kontrak antara Mahasiswa dan UMKM sudah dibuat (minimal status DRAFT); keduanya telah login.

Post-Kondisi Pesan dan lampiran berhasil terkirim, tersimpan di database, dan tampil pada chat room project secara realtime.

Skenario Utama

Aktor	Sistem
1. Membuka halaman “Project Room”.	2. Menampilkan daftar pesan terbaru antara Mahasiswa dan UMKM.

3. Mengetik pesan pada kolom chat dan menekan Enter atau ikon "Send".
4. Memvalidasi input pesan (tidak kosong, panjang \leq batas maksimum).
5. Menyimpan pesan ke database dan mengirimkannya ke penerima secara realtime.
6. Menampilkan pesan baru di chat room kedua pihak serta menunjukkan indikator terkirim dan dibaca.
7. Mengunggah file melalui tombol "Attach File".
8. Memvalidasi tipe dan ukuran file sesuai batas yang ditentukan (PDF, JPG, PNG, maks. 10 MB).
9. Menyimpan file ke server dan menambahkan tautan preview pada pesan.
10. Menampilkan notifikasi "File berhasil dikirim."

Skenario Eksepsional (Alternative Flow)

- | Aktor | Sistem |
|---|---|
| 1. Mengirim pesan kosong atau berisi karakter tidak valid. | 2. Menampilkan pesan error "Pesan tidak boleh kosong." |
| 3. Mengunggah file dengan ukuran melebihi batas atau format tidak didukung. | 4. Menampilkan pesan error "Format atau ukuran file tidak valid." |
| 5. Koneksi internet terputus saat mengirim pesan. | 6. Menampilkan status "Pesan gagal terkirim" dan memberikan opsi "Kirim ulang." |
| 7. Sesi login berakhir saat chat terbuka. | 8. Menampilkan pesan "Sesi berakhir, silakan login ulang." dan memblokir pengiriman pesan baru. |

3.2.1.9

Nama Use Case

Create Invoice & Escrow Payment

Deskripsi	UMKM dapat membuat invoice untuk project aktif dan melakukan pembayaran melalui payment gateway (Midtrans, Xendit, GoPay, DANA). Sistem akan menahan dana di escrow sampai project selesai.
Pre-Kondisi	Kontrak project sudah berstatus ONGOING, dan UMKM telah login ke sistem.
Post-Kondisi	Pembayaran berhasil diproses, invoice tersimpan di database, status escrow menjadi HELD, dan UMKM menerima bukti pembayaran (kwitansi PDF).

Skenario Utama

Aktor	Sistem
1. Membuka halaman project aktif.	2. Menampilkan rincian project dan status pembayaran.
2. Menekan tombol "Buat Invoice / Bayar Sekarang."	3. Menampilkan form pembuatan invoice berisi nominal, deskripsi, dan metode pembayaran.
3. Mengisi data invoice dan memilih metode pembayaran.	4. Memvalidasi input (nominal > 0, metode valid).
4. Menekan tombol "Bayar."	5. Mengirim permintaan transaksi ke payment gateway (PG) yang dipilih.
	6. Menerima callback dari PG saat pembayaran sukses.
	7. Menyimpan data pembayaran dan invoice di database dengan status PAID.

8. Mengubah status escrow menjadi HELD (dana ditahan).

9. Menghasilkan file kwitansi PDF dan mengirimkannya ke UMKM via email & dashboard.

10. Menampilkan notifikasi “Pembayaran berhasil, dana telah diamankan di escrow.”

Skenario Eksepsional (Alternative Flow)

Aktor	Sistem
1. Mengirim form dengan nominal kosong atau metode tidak dipilih.	2. Menampilkan pesan error “Data invoice tidak valid.”
3. Pembayaran gagal di payment gateway (saldo tidak cukup / transaksi dibatalkan).	4. Mengatur status invoice menjadi FAILED, menampilkan pesan “Pembayaran gagal, silakan coba lagi.”
5. Callback dari PG tidak diterima (network issue).	6. Menandai transaksi sebagai PENDING dan menunggu retry/callback manual.
7. Kesalahan teknis saat generate PDF kwitansi.	8. Menampilkan pesan “Pembayaran berhasil, namun kwitansi gagal dibuat.” dan memberi opsi unduh ulang nanti.

3.2.1.10

Nama Use Case	Complete Project & Release Escrow
Deskripsi	Setelah mahasiswa menyelesaikan project dan UMKM menyetujui hasilnya, sistem mengubah status project menjadi DONE, kemudian melepaskan dana yang ditahan di escrow ke saldo mahasiswa.

Pre-Kondisi Project berstatus ONGOING, pekerjaan telah diserahkan mahasiswa, dan UMKM sudah login.

Post-Kondisi Project berstatus DONE, dana escrow berpindah ke saldo mahasiswa, dan transaksi tercatat sebagai RELEASED.

Skenario Utama

Aktor	Sistem
1. Mahasiswa mengunggah hasil akhir (deliverable) di ruang project.	2. Menyimpan file hasil ke server dan memberi notifikasi ke UMKM.
3. UMKM meninjau hasil kerja mahasiswa.	4. Menampilkan tombol “Approve Completion” dan “Request Revision.”
4. UMKM menekan tombol “Approve Completion.”	5. Memvalidasi bahwa project berstatus ONGOING dan escrow tersedia.
	6. Mengubah status project menjadi DONE.
	7. Melepaskan dana escrow ke saldo mahasiswa (status transaksi: RELEASED).
	8. Mengirim notifikasi ke kedua pihak bahwa project telah selesai dan dana telah dicairkan.
	9. Menyimpan catatan transaksi dan memperbarui riwayat saldo mahasiswa.

Skenario Eksepsional (Alternative Flow)

Aktor	Sistem
-------	--------

1. UMKM menolak hasil dengan alasan belum sesuai.
2. Mengubah status project menjadi REVISION_REQUESTED, dan mengirim notifikasi ke mahasiswa.
3. UMKM tidak memberikan respons setelah 7 hari sejak hasil dikirim.
4. Sistem mengirim pengingat otomatis; jika tetap tidak ada respons hingga batas waktu (misalnya 14 hari), sistem otomatis menyetujui project dan me-release escrow.
5. Terjadi error saat memproses pencairan dana.
6. Sistem menandai status transaksi sebagai RELEASE_FAILED, menampilkan pesan “Gagal memproses pencairan, akan dicoba ulang otomatis.”
7. Koneksi ke payment service gagal saat release.
8. Menunda transaksi dan menampilkan status “Pending Release.”

3.2.1.11

Nama Use Case

Dispute & Refund Handling

- Deskripsi** Salah satu pihak (Mahasiswa atau UMKM) dapat membuka sengketa (dispute) jika terjadi ketidaksesuaian hasil project, keterlambatan, atau pelanggaran kontrak. Admin kemudian memediasi dan menentukan apakah dana akan direfund, direlease sebagian, atau ditahan.
- Pre-Kondisi** Project berstatus ONGOING atau DONE, terdapat kontrak aktif dan transaksi escrow yang masih ditahan (HELD).
- Post-Kondisi** Dispute tersimpan di sistem dengan status penyelesaian (RESOLVED / REJECTED / REFUNDED), dan tindakan yang diputuskan admin dijalankan pada transaksi terkait.

Skenario Utama

Aktor

Sistem

1. Mahasiswa atau UMKM membuka halaman project dan menekan tombol “Laporkan Sengketa (Open Dispute)”.
2. Menampilkan form dispute berisi kolom deskripsi, bukti (file/chat), dan jenis permasalahan.
3. User mengisi form dispute secara lengkap dan menekan “Kirim Dispute.”
4. Memvalidasi input (kolom deskripsi wajib, file ≤ 10 MB).
5. Menyimpan data dispute ke database dengan status awal PENDING REVIEW.
6. Mengirim notifikasi ke admin bahwa ada sengketa baru.
4. Admin membuka daftar dispute di dashboard admin.
7. Menampilkan detail laporan, bukti, dan histori project.
5. Admin meninjau dan berkomunikasi dengan kedua pihak (jika perlu klarifikasi).
8. Mencatat komentar dan status review (IN PROGRESS).
6. Admin menentukan hasil akhir: refund penuh, refund sebagian, atau tolak laporan.
9. Menjalankan aksi keuangan sesuai keputusan:
 - Refund → mengembalikan dana ke UMKM.
 - Release → mencairkan sebagian/penuh ke Mahasiswa.
10. Mengubah status dispute menjadi RESOLVED dan mencatat log audit.
11. Mengirim notifikasi hasil mediasi ke kedua pihak.

Skenario Eksepsional (Alternative Flow)

Aktor

Sistem

1. User mengirim dispute tanpa deskripsi atau bukti.
2. Menampilkan pesan error “Data laporan belum lengkap.”
3. Admin tidak merespons dispute dalam waktu yang ditentukan (mis. 3 hari).
4. Sistem mengirim reminder otomatis ke admin dan mencatat eskalasi.
5. Refund gagal diproses oleh payment gateway.
6. Sistem menandai transaksi sebagai REFUND_FAILED dan menjadwalkan retry otomatis.
7. Salah satu pihak mengirim data palsu/spam.
8. Admin menandai laporan INVALID, mengubah status menjadi REJECTED, dan memberi peringatan ke pelapor.
9. Koneksi ke server terputus saat pengiriman dispute.
10. Menampilkan pesan “Gagal mengirim laporan, periksa koneksi Anda.”

3.2.1.12

Nama Use Case

Payout / Withdrawal

Deskripsi	Mahasiswa dapat menarik saldo yang telah diterima dari proyek (setelah dana escrow dirilis) ke rekening bank atau e-wallet yang terdaftar.
Pre-Kondisi	Mahasiswa telah login, memiliki saldo yang tersedia (\geq minimum withdrawal), dan akun payout (bank/e-wallet) telah terverifikasi.
Post-Kondisi	Permintaan penarikan tersimpan di sistem dengan status PENDING, kemudian berubah menjadi SUCCESS atau FAILED setelah proses selesai. Riwayat transaksi diperbarui dan notifikasi dikirim ke mahasiswa.

Skenario Utama

Aktor

Sistem

1. Mahasiswa membuka halaman “Saldo / Wallet.”
2. Menampilkan saldo tersedia dan daftar penarikan sebelumnya.
2. Menekan tombol “Tarik Dana.”
3. Menampilkan form penarikan yang berisi nominal, metode pembayaran, dan nomor rekening / e-wallet.
3. Mengisi data penarikan dan menekan tombol “Konfirmasi.”
4. Memvalidasi data (nominal \geq minimum, rekening valid, saldo cukup).
5. Menyimpan permintaan penarikan ke database dengan status PENDING.
6. Mengirim permintaan ke payment gateway / bank API untuk proses transfer.
7. Setelah berhasil, mengubah status menjadi SUCCESS, memperbarui saldo user, dan mencatat transaksi di log keuangan.
8. Mengirim notifikasi ke mahasiswa bahwa penarikan berhasil.

Skenario Eksepsional (Alternative Flow)

- | Aktor | Sistem |
|---|--|
| 1. Mengisi nominal di bawah batas minimum (misal < Rp10.000). | 2. Menampilkan pesan error “Nominal penarikan tidak memenuhi batas minimum.” |
| 3. Mengisi data rekening tidak valid atau kosong. | 4. Menampilkan pesan error “Data rekening tidak valid, periksa kembali.” |
| 5. Menekan “Konfirmasi” saat saldo tidak mencukupi. | 6. Menampilkan pesan “Saldo tidak mencukupi untuk penarikan.” |

7. Proses payout gagal di payment gateway (rekening tujuan tidak ditemukan / error teknis).
8. Mengubah status menjadi FAILED, menampilkan pesan “Penarikan gagal, silakan coba ulang atau hubungi admin.”
9. Koneksi internet terputus saat pengiriman form.
10. Menampilkan pesan “Gagal mengirim permintaan, periksa koneksi Anda.”

3.2.1.13

Nama Use Case

Reviews & Ratings

Deskripsi	Setelah project selesai, mahasiswa dan UMKM dapat saling memberikan penilaian (rating) dan ulasan (review) untuk meningkatkan transparansi dan reputasi dalam sistem.
Pre-Kondisi	Project berstatus DONE/CLOSED dan kedua pihak (Mahasiswa & UMKM) telah login ke sistem.
Post-Kondisi	Review dan rating tersimpan di database, tampil pada profil user yang dinilai, dan dapat dilihat oleh pengguna lain.

Skenario Utama

Aktor

Sistem

1. Membuka halaman “Project Selesai” di dashboard.
2. Menampilkan daftar project dengan status DONE/CLOSED yang belum diberi review.
2. Menekan tombol “Beri Ulasan” pada project terkait.
3. Menampilkan form ulasan dengan kolom rating (1–5 bintang) dan komentar.
3. Mengisi rating dan komentar ulasan secara lengkap.
4. Memvalidasi input (rating wajib, komentar minimal 10 karakter).

4. Menekan tombol “Kirim Ulasan.”
5. Menyimpan data review ke database dengan status PUBLISHED.
6. Memperbarui profil pihak yang dinilai dengan rata-rata rating terbaru.
7. Menampilkan notifikasi “Ulasan berhasil dikirim.”
8. Menyediakan opsi untuk melihat daftar ulasan yang telah dikirim.

Skenario Eksepsional (Alternative Flow)

Aktor	Sistem
1. Mengirim ulasan tanpa mengisi rating atau komentar.	2. Menampilkan pesan error “Rating dan komentar wajib diisi.”
3. Menulis komentar kurang dari batas minimal karakter.	4. Menampilkan pesan “Komentar terlalu singkat, berikan ulasan yang lebih detail.”
5. Mengirim ulasan untuk project yang belum selesai (ONGOING).	6. Menampilkan pesan “Ulasan hanya dapat diberikan setelah project selesai.”
7. Mengirim ulasan dua kali untuk project yang sama.	8. Menampilkan pesan “Anda sudah memberikan ulasan untuk project ini.”
9. Koneksi internet terputus saat mengirim ulasan.	10. Menampilkan pesan “Gagal mengirim ulasan, periksa koneksi Anda.”

3.2.1.14

Nama Use
Case

Notification Center

Deskripsi	Sistem mengirimkan notifikasi (push, email, atau in-app) kepada pengguna (Mahasiswa, UMKM, atau Admin) untuk setiap event penting seperti lamaran diterima, kontrak dibuat, pembayaran berhasil, project selesai, atau ulasan diterima.
Pre-Kondisi	Event penting terjadi dalam sistem (misalnya apply project, pembayaran, approval, review, dll.), dan user yang bersangkutan telah login atau memiliki akun aktif.
Post-Kondisi	Notifikasi dikirim ke pengguna, tersimpan di database, dan dapat dilihat di halaman Notification Center serta ditandai sebagai terbaca.

Skenario Utama

Aktor	Sistem
1. Melakukan aksi tertentu yang memicu event (contoh: Mahasiswa mengajukan proposal, UMKM menyetujui kontrak, pembayaran berhasil, project selesai).	2. Mendeteksi event penting yang terjadi di sistem.
	3. Membuat notifikasi berdasarkan tipe event (Apply Success, Contract Approved, Payment Confirmed, Project Completed, Review Received).
	4. Menyimpan data notifikasi ke tabel notifications dengan status UNREAD.
	5. Mengirimkan notifikasi ke pengguna sesuai preferensi (in-app, email, atau push).
2. User membuka menu “Notifikasi” di dashboard.	6. Menampilkan daftar notifikasi dengan waktu dan jenis event.

- | | |
|---|---|
| 3. User menekan salah satu notifikasi. | 7. Mengarahkan user ke halaman terkait event (mis. detail project, chat room, payment). |
| 4. User menandai notifikasi sebagai sudah dibaca. | 8. Mengubah status notifikasi menjadi READ di database. |

Skenario Eksepsional (Alternative Flow)

- | Aktor | Sistem |
|--|--|
| 1. User menonaktifkan notifikasi email/push di pengaturan. | 2. Sistem hanya mengirim notifikasi in-app dan menandai preferensi pengguna. |
| 3. Notifikasi gagal dikirim (email bounce / push token tidak valid). | 4. Sistem menandai status notifikasi sebagai FAILED dan mencoba pengiriman ulang otomatis. |
| 5. Koneksi internet user terputus saat membuka halaman notifikasi. | 6. Menampilkan pesan “Gagal memuat notifikasi, periksa koneksi Anda.” |
| 7. User mencoba membuka notifikasi yang sudah dihapus. | 8. Menampilkan pesan “Notifikasi tidak ditemukan.” |

3.2.1.15

- | Nama Use Case | Report & Flag Content/User |
|---------------|--|
| Deskripsi | Pengguna (Mahasiswa atau UMKM) dapat melaporkan konten, project, pesan chat, atau user lain yang dianggap melanggar aturan. Laporan akan diteruskan ke Admin untuk ditinjau dan ditindaklanjuti. |
| Pre-Kondisi | User telah login dan menemukan konten, pesan, atau user yang melanggar kebijakan sistem. |

Post-Kondisi Laporan tersimpan di sistem dengan status **PENDING REVIEW**, Admin menerima notifikasi, dan tindak lanjut tercatat di log moderasi.

Skenario Utama

Aktor	Sistem
1. User membuka halaman project, profil user lain, atau pesan chat yang ingin dilaporkan.	2. Menampilkan tombol “Laporkan / Report” di area terkait.
2. User menekan tombol “Report.”	3. Menampilkan form laporan dengan kolom kategori pelanggaran (mis. spam, konten tidak pantas, penipuan) dan kolom deskripsi.
3. User mengisi kategori dan deskripsi laporan dengan detail.	4. Memvalidasi input (kategori wajib dipilih, deskripsi minimal 10 karakter).
4. User menekan tombol “Kirim Laporan.”	5. Menyimpan laporan ke database dengan status PENDING REVIEW dan timestamp.
	6. Mengirim notifikasi ke Admin bahwa ada laporan baru untuk ditinjau.
	7. Menampilkan pesan konfirmasi “Laporan Anda telah dikirim dan sedang ditinjau.”

Skenario Eksepsional (Alternative Flow)

Aktor	Sistem
1. User mengirim laporan tanpa memilih kategori atau deskripsi kosong.	2. Menampilkan pesan error “Harap pilih kategori dan isi deskripsi laporan.”
3. User mencoba melaporkan konten yang sudah dilaporkan sebelumnya.	4. Menampilkan pesan “Konten ini sudah dilaporkan sebelumnya.”

- | | |
|--|---|
| 5. Admin tidak menanggapi laporan dalam jangka waktu tertentu. | 6. Sistem mengirim reminder otomatis ke Admin terkait laporan yang belum ditindaklanjuti. |
| 7. Koneksi internet user terputus saat mengirim laporan. | 8. Menampilkan pesan “Gagal mengirim laporan, periksa koneksi Anda dan coba lagi.” |
| 9. Sistem gagal menyimpan laporan ke database. | 10. Menampilkan pesan “Terjadi kesalahan saat menyimpan laporan, silakan coba ulang.” |

3.2.1.16

Nama Use Case

Admin Moderation & Monitoring

Deskripsi Admin dapat melakukan verifikasi akun UMKM, meninjau laporan pelanggaran, memantau aktivitas user dan transaksi, membuat pengumuman sistem, serta mengekspor data laporan dalam format CSV atau PDF.

Pre-Kondisi Admin telah login menggunakan akun dengan hak akses penuh.

Post-Kondisi Tindakan admin tersimpan di sistem (audit log), laporan atau akun diverifikasi atau ditindaklanjuti, dan data ekspor berhasil diunduh atau dikirim.

Skenario Utama

Aktor	Sistem
1. Admin membuka halaman Dashboard Admin.	2. Menampilkan ringkasan metrik sistem (jumlah user, project, transaksi, laporan).
2. Admin memilih menu “Verifikasi UMKM.”	3. Menampilkan daftar UMKM yang belum diverifikasi beserta dokumen pendukungnya.

3. Admin meninjau data dan menekan tombol “Verifikasi” atau “Tolak.”
4. Memperbarui status akun UMKM di database (VERIFIED / REJECTED).
5. Admin membuka menu “Laporan & Moderasi.”
6. Menampilkan daftar laporan (dari UC-14) dengan status PENDING REVIEW.
6. Admin meninjau laporan dan menentukan tindakan (hapus konten, suspend user, abaikan laporan).
7. Menyimpan keputusan ke database dan mencatat aktivitas di log audit.
8. Admin membuat pengumuman baru untuk seluruh user.
9. Menampilkan form pengumuman dan menyimpan data ke tabel announcements.
10. Menampilkan banner/popup pengumuman di dashboard user.
10. Admin mengekspor data transaksi atau laporan.
11. Menghasilkan file CSV/PDF berisi data sesuai filter dan menampilkan tombol unduh.
12. Menampilkan pesan “Data berhasil diekspor.”

Skenario Eksepsional (Alternative Flow)

- | Aktor | Sistem |
|---|--|
| 1. Admin menolak verifikasi tanpa memberikan alasan. | 2. Menampilkan pesan error “Alasan penolakan wajib diisi.” |
| 3. Admin mencoba memproses laporan yang sudah ditutup. | 4. Menampilkan pesan “Laporan ini sudah diproses sebelumnya.” |
| 5. Admin gagal mengunggah banner pengumuman (file terlalu besar). | 6. Menampilkan pesan “Ukuran file pengumuman melebihi batas maksimal.” |

7. Admin mengekspor data dalam jumlah besar (>10.000 baris).
8. Menampilkan pesan “Data terlalu besar, ekspor akan dikirim via email.”
9. Koneksi ke server terputus saat menyimpan keputusan.
10. Menampilkan pesan “Gagal menyimpan, periksa koneksi Anda.” dan mencatat log gagal.

3.2.1.17

3.2.1.18

3.2.2 *Class Diagram:*

< Gambarkan kelas diagram model domain untuk sistem yang akan dikembangkan . Tuliskan lengkap keterhubungan antar class dengan atribut minimal dan multiplicitynya >

4. Kebutuhan Lain - Lain

4.1 Antarmuka Pengguna

<Jelaskan kebutuhan yang terkait dengan kebutuhan UI dan UX sesuai dengan karakteristik Class Pengguna .>

4.2 Antarmuka Perangkat Keras

<Jelaskan kebutuhan yang menjelaskan bagaimana sistem akan berkomunikasi dengan perangkat keras yang akan digunakan di sistem ini. Misal dengan perangkat Bar Code Reader, Server, mesin pembaca fingerprint dsb .>

4.3 Antarmuka Perangkat Lunak

<Jelaskan kebutuhan yang menjelaskan bagaimana sistem akan berinteraksi dengan sistem lainnya, yang merupakan satu kesatuan. Misalnya sistem tagihan akan berinteraksi dengan sistem pendaftaran pelanggan.>

4.4 Antarmuka Komunikasi

<Jelaskan requirements yang terkait dengan proses komunikasi yang dibutuhkan produk ini, termasuk e-mail, web browser, protokol komunikasi server jaringan, formulir elektronik, dan sebagainya. Definisikan format pesan yang tepat. Identifikasi standar komunikasi apapun yang akan digunakan, seperti FTP atau HTTP. Tentukan masalah keamanan komunikasi atau enkripsi, kecepatan transfer data, dan mekanisme sinkronisasi.>

4.5 Fitur sistem cerdas

Jelaskan kebutuhan yang berkaitan dengan pemanfaatan fitur – fitur yang dapat menjadikan aplikasi yang dikembangkan menjadi sistem yang cerdas

Lampiran A: Daftar Kata-Kata Asing

<Tentukan semua requirements yang diperlukan untuk menafsirkan SKPL ini dengan benar, termasuk akronim dan singkatan. Anda mungkin ingin membuat daftar yang terpisah yang mencakup beberapa proyek atau seluruh organisasi, dan hanya mencakup istilah khusus untuk satu proyek di setiap SKPL.>

Lampiran B: Analysis Models

<Opsional. Masukkan model analisis yang berhubungan, seperti, state-transition diagrams, flow-map, atau entity-relationship diagrams (ERD).

Catatan : Flow-map dan ERD + Skema Relasi BD (untuk aplikasi SI) dan Flow chart (untuk aplikasi non SI, misal game

>