

## BAB 3 METODOLOGI PENELITIAN

### 3.1 Studi Literatur

Tahapan ini merupakan fondasi dari penelitian, di mana identifikasi dan perumusan masalah yang dibahas pada Bab 1 dilakukan melalui penelusuran serta analisis referensi yang relevan. Studi literatur dilakukan dengan menelusuri dan mempelajari berbagai referensi serta teori yang digunakan dalam proses penelitian. Sumber referensi dan teori yang digunakan berasal dari artikel, jurnal, dan buku yang berkaitan dengan Manajemen Tugas, Gamifikasi, Octalysis, *Hedonic Motivation System Adoption Model* (HMSAM), dan Skala Likert.

### 3.2 Perancangan Website

Proses perancangan *website* meliputi beberapa tahapan, yaitu perancangan gamifikasi, perancangan arsitektur *website*, perancangan *flowchart* dan *sitemap*, perancangan tampilan antarmuka, dan perancangan aset.

#### 3.2.1 Perancangan Gamifikasi

*Website* gamifikasi manajemen tugas "*SpaceQuest*" dibangun menggunakan metode gamifikasi dengan *framework* Octalysis. Elemen-elemen permainan yang digunakan berdasarkan delapan pendorong inti (8 *core drives*) dari *framework* Octalysis adalah sebagai berikut:

1. *Epic Meaning & Calling* (Makna Epik & Panggilan)

Elemen permainan yang diambil dari pendorong ini adalah *Narrative* (Narasi). Narasi yang dirancang menggunakan tema ekspedisi luar angkasa. Pengguna berperan sebagai seorang *Commander* yang menjalankan berbagai *Missions* dan *Expeditions*. Istilah seperti *Starship Dashboard*, *Mission Logs*, dan *Crew Expeditions* digunakan di seluruh *website* untuk menciptakan suasana fiktif yang imersif, membuat pengguna merasa menjadi bagian dari sebuah misi produktivitas yang penting.

2. *Development & Accomplishment* (Pengembangan & Pencapaian)

Elemen permainan yang diambil dari pendorong ini adalah *Badges* (Lencana) dan *Progress Bar*. Sistem *Medal Bay* berfungsi sebagai lemari piala digital di mana pengguna dapat melihat semua lencana yang telah diperoleh. Selain itu, *Level* dan *XP Bar* yang selalu terlihat di *header website* memberikan umpan balik visual yang konstan mengenai kemajuan dan pencapaian pengguna.

3. *Empowerment of Creativity & Feedback* (Pemberdayaan Kreativitas & Umpan Balik)

Elemen permainan yang diambil dari pendorong ini adalah *Kanban Board* pada fitur *Crew Expeditions*. Pengguna (khususnya *Commander*) memiliki kebebasan untuk membuat proyek, menamai ekspedisi, dan mengatur tugas-tugas di dalam papan Kanban. Mekanisme *drag-and-drop* untuk memindahkan tugas dari *To Do* ke *In Progress* hingga *Done* memberikan umpan balik visual yang instan dan rasa kontrol penuh atas alur kerja proyek.

4. *Ownership & Possession* (Kepemilikan & Kepemilikan)

Elemen permainan yang diambil dari pendorong ini adalah *Exchangeable Points* (Poin yang Dapat Ditukar), *Customizable Avatar & Profile*, dan *Inventory*. Pengguna mengumpulkan *Cosmic Points (CP)* dari menyelesaikan tugas dan misi. Poin ini dapat digunakan di *Star Market* untuk membeli *item* kustomisasi seperti Tema visual dan Bingkai *Avatar*. Pengguna juga memiliki kepemilikan penuh atas nama dan gambar profil pengguna, yang dapat diubah di halaman *Ship Systems*.

5. *Social Influence & Relatedness* (Pengaruh Sosial & Keterhubungan)

Elemen permainan yang diambil dari pendorong ini diimplementasikan melalui dua fitur utama: *Crew Expeditions* (Proyek Tim) dan *Leaderboard* (Papan Peringkat). Fitur *Crew Expeditions* memungkinkan pengguna untuk mengundang anggota lain dan berkolaborasi dalam satu proyek, menciptakan rasa kebersamaan. Sementara itu, *Hall of Fame* atau *Leaderboard* memotivasi pengguna dengan menampilkan peringkat pengguna berdasarkan *XP* dibandingkan dengan komandan lain di galaksi.

6. *Scarcity & Impatience* (Kelangkaan & Ketidaksabaran)

Elemen permainan yang diambil dari pendorong ini adalah *Daily Gated Features* (Fitur Terbatas Harian). Fitur seperti *Daily Login Bonus* dan *Daily Discovery* hanya dapat diklaim satu kali setiap hari. Hal ini menciptakan

kelangkaan (*scarcity*) dan mendorong pengguna untuk kembali setiap hari agar tidak ketinggalan hadiah, yang memicu rasa "ketidaksabaran" yang positif untuk *login* di hari berikutnya.

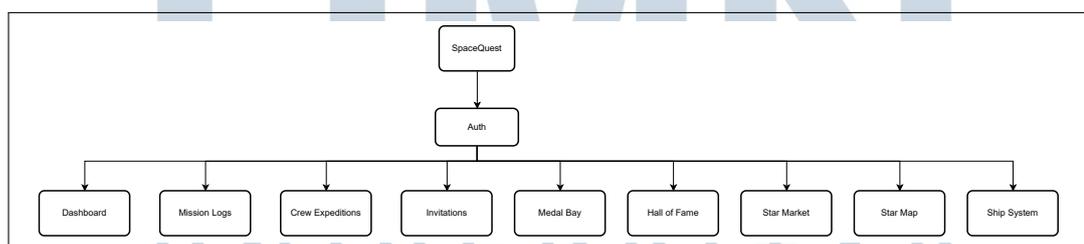
#### 7. *Unpredictability & Curiosity* (Ketidakpastian & Rasa Penasaran)

Elemen permainan yang diambil dari pendorong ini adalah *Daily Discovery* (Penemuan Harian). Setiap hari, pengguna dapat mengklaim *Supply Drop* yang berisi hadiah *XP* dan Kredit dengan jumlah yang acak. Elemen ketidakpastian mengenai jumlah hadiah yang didapat ini menciptakan kejutan kecil yang menyenangkan dan mendorong rasa penasaran untuk melakukan klaim setiap hari.

#### 8. *Loss & Avoidance* (Kehilangan & Penghindaran)

Elemen permainan yang diambil dari pendorong ini diterapkan melalui sistem *Login Streak* (Rentetan *Login*). Pengguna termotivasi untuk *login* setiap hari untuk mempertahankan dan meningkatkan rentetan *login* pengguna. Jika pengguna melewatkan satu hari, pengguna akan "kehilangan" rentetan tersebut dan *bonus* yang lebih besar yang menyertainya. Ini mendorong pengguna untuk menghindari kehilangan progres (*streak*) yang telah dibangun.

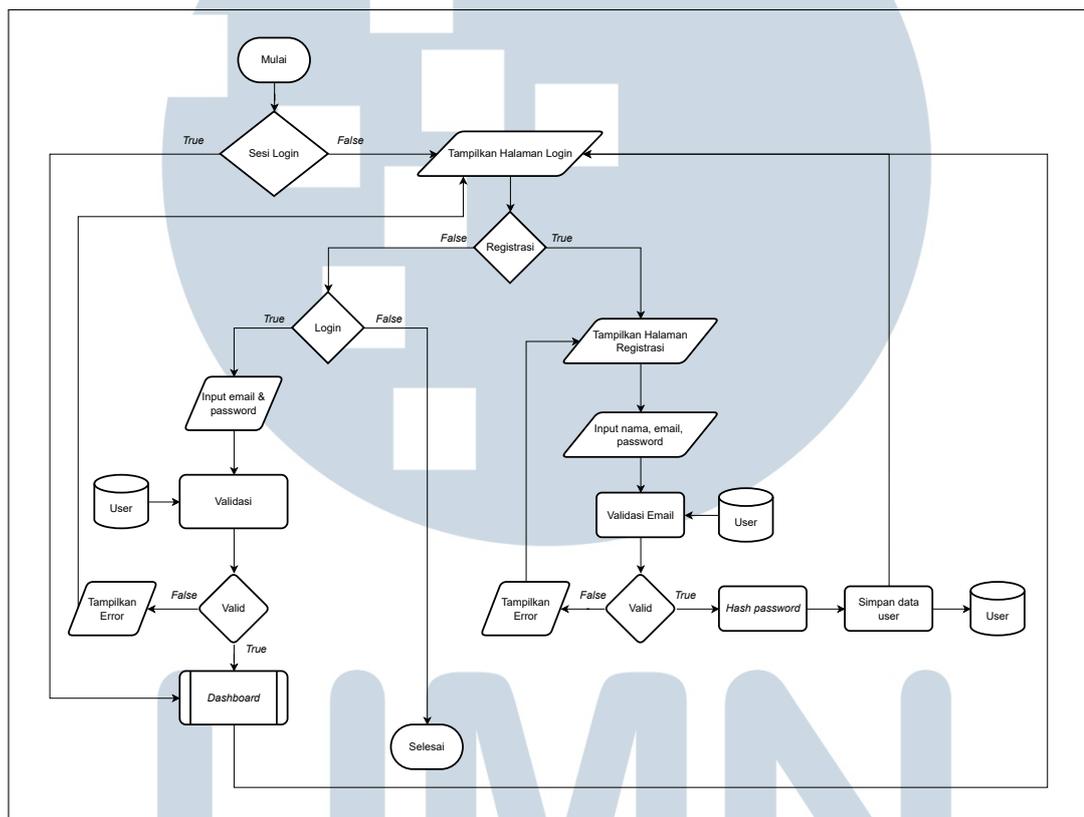
### 3.2.2 Perancangan Sitemap dan Flowchart



Gambar 3.1. *Sitemap* Website SpaceQuest

Gambar 3.1 menunjukkan *sitemap website* yang memetakan seluruh halaman serta fitur yang ada pada "SpaceQuest". *Sitemap* ini menggambarkan struktur hierarki dan alur navigasi utama dalam *website*, mulai dari proses autentikasi hingga fitur-fitur inti. Untuk mempermudah pemahaman, fitur-fitur yang saling berkaitan dikelompokkan menjadi modul-modul fungsional yang akan dijelaskan secara terperinci pada bagian selanjutnya.

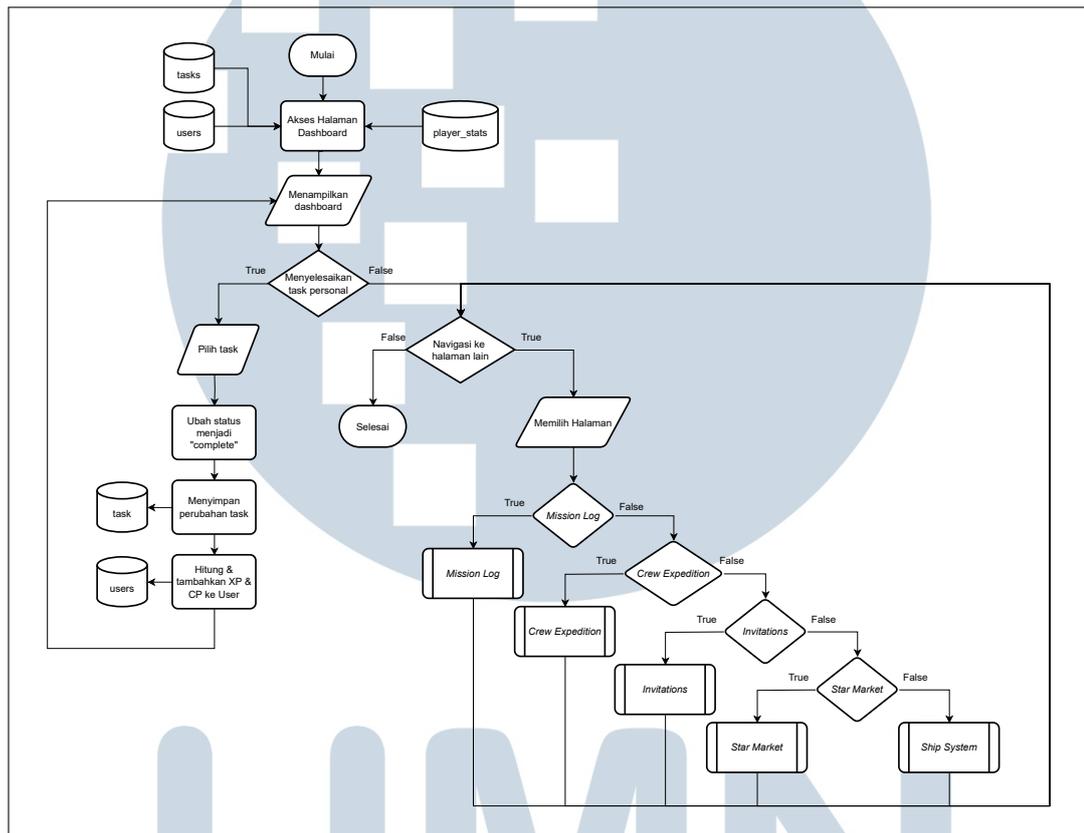
Autentikasi menangani proses pengguna pertama kali berinteraksi dengan *Website*, mulai dari pendaftaran akun baru hingga masuk ke dalam sistem. Modul ini mencakup halaman registrasi dan halaman *login*, yang merupakan gerbang utama sebelum pengguna dapat mengakses fitur-fitur "SpaceQuest". Gambar 3.2 menjelaskan alur proses yang terjadi ketika pengguna melakukan registrasi dan *login*.



Gambar 3.2. Flowchart Autentikasi

Gambar 3.2 menjelaskan alur proses yang terjadi ketika pengguna melakukan registrasi dan *login*. Pengecekan Sesi, saat pengguna membuka *website*, sistem akan memeriksa keberadaan sesi *login* yang valid (berdasarkan *token* JWT). Jika sesi ditemukan, pengguna akan langsung diarahkan ke halaman utama atau *dashboard*. Jika tidak, pengguna akan diarahkan ke halaman *Login*. Alur Registrasi, pengguna yang belum memiliki akun dapat memilih opsi untuk registrasi. Pengguna akan diminta mengisi data seperti nama, *email*, dan *password*. Sistem akan melakukan validasi terhadap data yang diinput. Jika data valid, *password* akan di-*hash* untuk keamanan dan data pengguna akan disimpan ke dalam *database*. Setelah berhasil, pengguna diarahkan kembali ke halaman *Login*. Pengguna memasukkan

*email* dan *password*. Sistem akan memvalidasi kredensial ini dengan data yang ada di *database*. Jika kredensial tidak cocok, sebuah pesan kesalahan akan ditampilkan. Jika berhasil, sistem akan membuat sebuah sesi aman menggunakan *JSON Web Token (JWT)* yang kemudian disimpan di sisi klien (*browser*) dan pengguna diarahkan ke *dashboard website*.

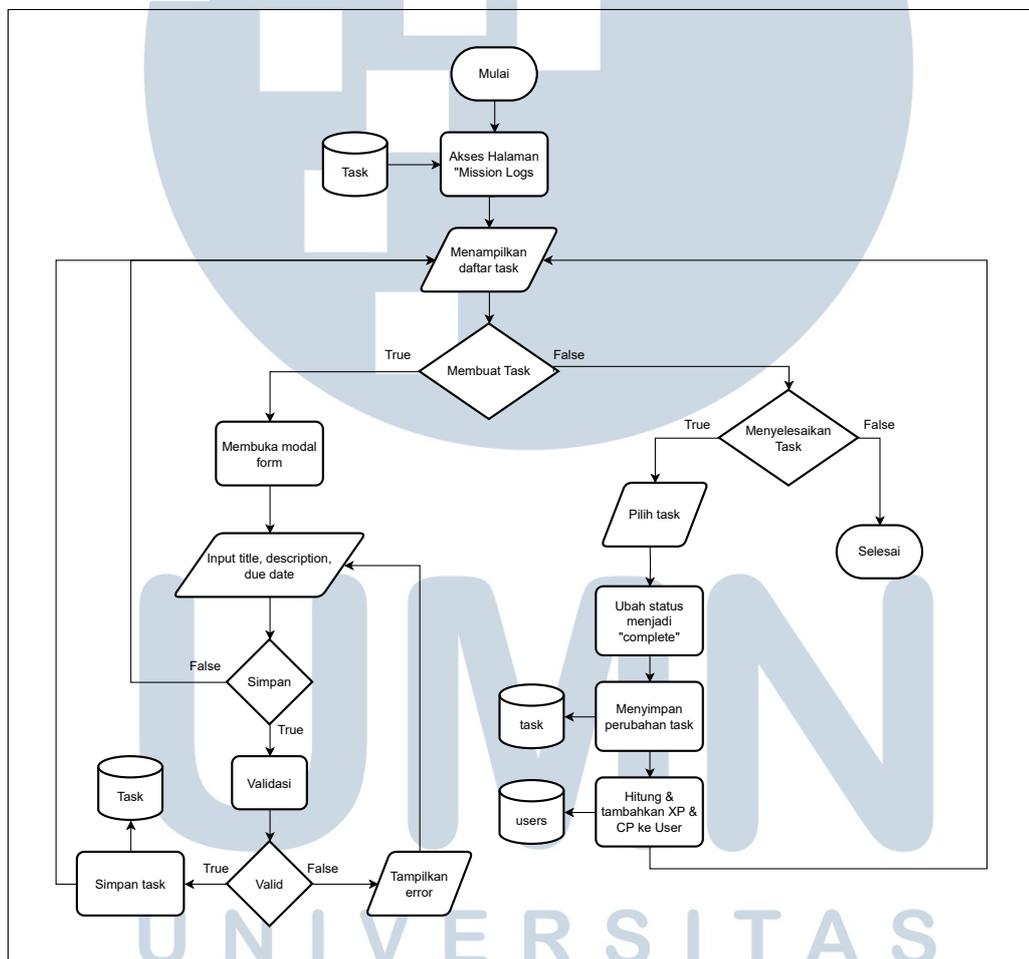


Gambar 3.3. *Flowchart Dashboard*

Gambar 3.3 mengilustrasikan alur interaksi utama yang dapat dilakukan pengguna dari Halaman Utama atau "Starship Dashboard". Proses dimulai saat pengguna mengakses halaman, di mana sistem akan mengambil data relevan dari tabel *tasks* dan *player\_stats* untuk ditampilkan. Dari sini, pengguna memiliki dua jalur interaksi utama. Pengguna dapat langsung menyelesaikan tugas pribadi dari *dashboard*, yang akan memicu serangkaian proses gamifikasi: mengubah status di tabel *task*, lalu menghitung dan menambahkan imbalan XP serta CP ke tabel *users*. Setelah itu, alur kembali ke tampilan *dashboard* yang sudah diperbarui. Jalur interaksi kedua adalah ketika pengguna memilih untuk bernavigasi ke fitur lain yang lebih spesifik. Salah satu navigasi kunci dari *dashboard* adalah menuju halaman "Mission Log", yang menyediakan fungsionalitas manajemen tugas pribadi secara

lebih mendalam. Alur proses yang lebih rinci untuk mengelola tugas di halaman tersebut akan diilustrasikan pada *flowchart* berikutnya.

Manajemen Tugas Pribadi (*Mission Logs*), merupakan inti dari fungsionalitas produktivitas individu dalam *website* "SpaceQuest". Di sini, pengguna dapat membuat dan menyelesaikan tugas-tugas pribadi pengguna. Setiap tugas yang diselesaikan akan memberikan imbalan berupa *Experience Points* (XP) dan *Cosmic Points* (CP), yang menjadi pendorong utama dalam mekanisme gamifikasi.



Gambar 3.4. *Flowchart* Manajemen Tugas Pribadi

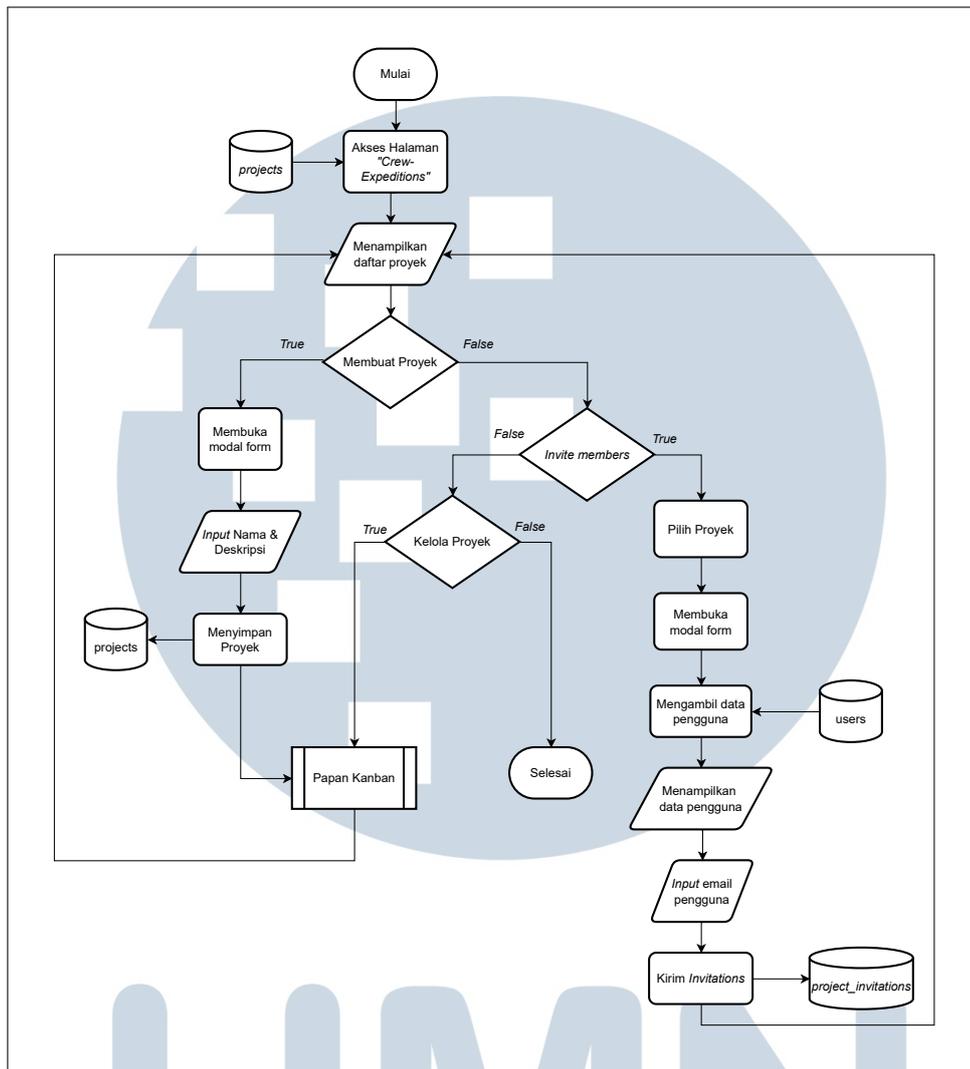
Gambar 3.4 merincikan alur proses pengguna dalam mengelola tugas-tugas pribadinya. Alur ini mencakup proses pembuatan tugas baru hingga penyelesaiannya, yang terintegrasi langsung dengan sistem gamifikasi. Ketika pengguna mengakses halaman "Mission Logs", sistem kemudian mengambil data tugas dari *database* yang dimiliki oleh pengguna tersebut dan menampilkannya

dalam dua kategori: tugas yang masih aktif dan yang sudah selesai. Pengguna dapat memilih untuk menambahkan tugas baru, sebuah *form* akan ditampilkan untuk memasukkan detail seperti nama tugas, deskripsi, dan tanggal. Setelah disimpan, data akan divalidasi dan disimpan ke *database* sebagai tugas aktif yang baru.

Dibagian penyelesaian tugas, pengguna dapat memilih salah satu tugas dari daftar aktif dan menandainya sebagai "selesai". Aksi ini akan memicu beberapa proses di *backend*, yaitu status tugas diubah menjadi "selesai" di *database*. *GameLogicService* dipanggil untuk secara otomatis menghitung dan menambahkan hadiah berupa XP dan CP ke akun pengguna. Terdapat batasan harian untuk perolehan XP dan CP dari tugas pribadi untuk menjaga keseimbangan permainan. Sistem juga akan memeriksa apakah penyelesaian tugas tersebut memenuhi kriteria untuk membuka pencapaian (*achievement*) atau menyelesaikan misi (*quest*) tertentu. Daftar tugas di antarmuka pengguna akan diperbarui, memindahkan tugas tersebut dari daftar aktif ke daftar selesai.

Selanjutnya modul "*Crew Expeditions*" adalah fitur kolaboratif utama yang memungkinkan pengguna untuk bekerja sama dalam sebuah proyek. Pengguna dapat membuat sebuah "ekspedisi", mengundang anggota lain, dan mengelola tugas-tugas secara bersama-sama menggunakan papan Kanban. Fitur ini dirancang untuk mendorong interaksi sosial dan kerja sama tim.





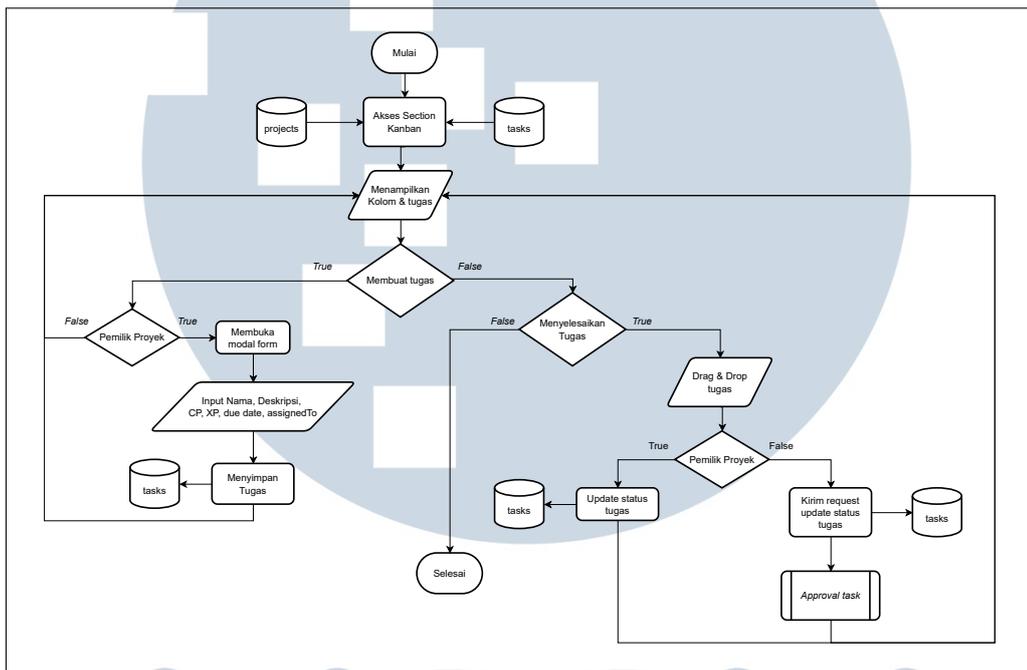
Gambar 3.5. Flowchart Modul Manajemen Proyek

Gambar 3.5 mengilustrasikan alur tingkat tinggi dari manajemen proyek. Ketika pengguna menavigasi ke halaman "Crew Expeditions". Sistem akan menampilkan daftar semua proyek di mana pengguna terdaftar sebagai anggota atau pemilik.

Dibagian pembuatan proyek, pengguna (yang bertindak sebagai *Commander*) dapat memulai proyek baru dengan memasukkan nama dan deskripsi ekspedisi. Setelah dibuat, proyek tersebut akan muncul di daftar proyek pengguna, dan pengguna secara otomatis menjadi pemilik proyek tersebut. Dibagian mengundang anggota, pemilik proyek dapat mengundang pengguna lain untuk bergabung ke dalam ekspedisinya melalui *email*. Sistem akan mengirimkan sebuah *invitation* yang akan muncul di halaman "Invitations" milik pengguna yang

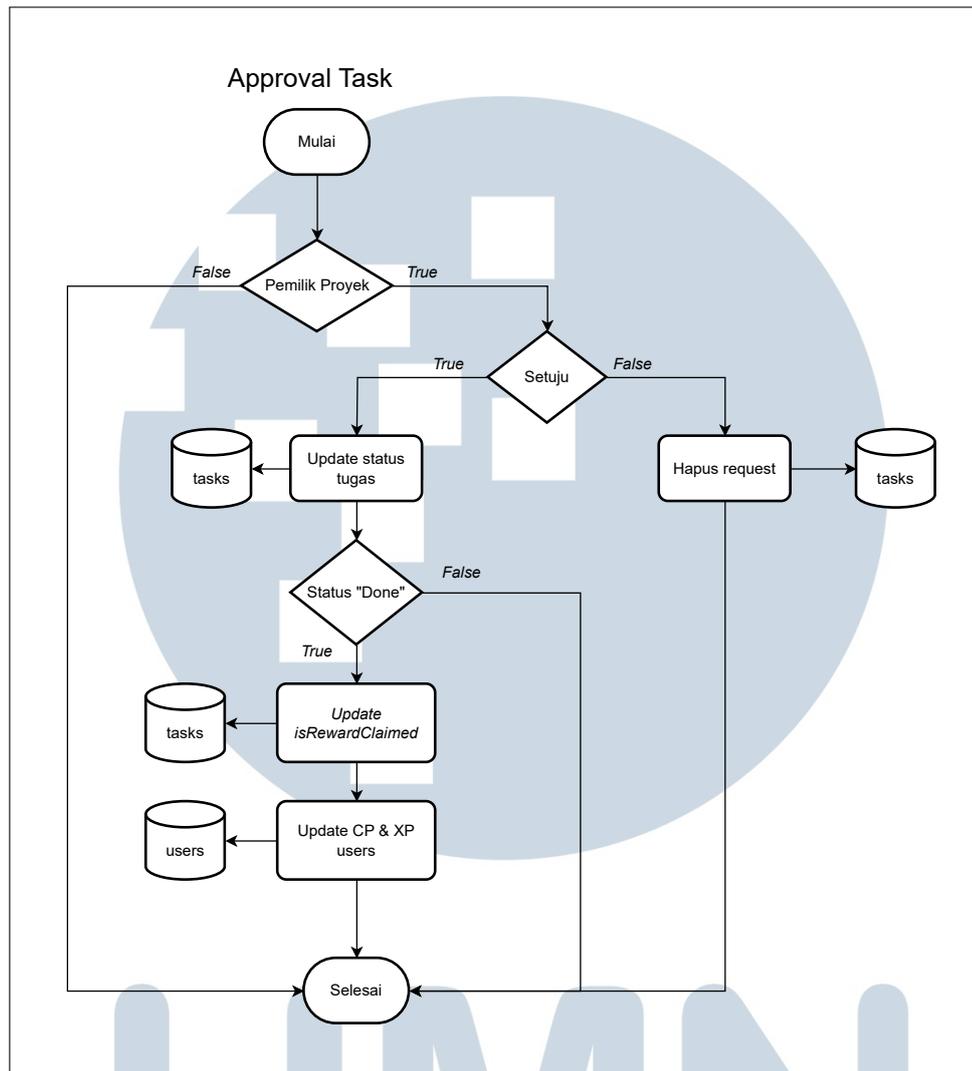
diundang. Ketika mengakses proyek, pengguna dapat memilih salah satu proyek dari daftar untuk masuk ke halaman detail proyek, yang berisi Papan Kanban. Di sini, seluruh alur kerja tugas kolaboratif dikelola.

Papan Kanban adalah pusat dari fitur kolaborasi, di mana tugas-tugas proyek divisualisasikan dalam beberapa kolom status (misalnya, 'Backlog', 'To Do', 'In Progress', 'Done') untuk memudahkan pelacakan progres.



Gambar 3.6. Flowchart Alur Kerja Utama Papan Kanban

Gambar 3.6 menjelaskan alur kerja utama pada papan Kanban. Saat pengguna membuka sebuah proyek, sistem akan mengambil dan menampilkan semua tugas yang terkait. Anggota tim dapat membuat tugas baru yang spesifik untuk proyek tersebut, yang akan muncul di kolom awal. Tugas-tugas ini kemudian dapat dipindahkan antar kolom menggunakan mekanisme *drag-and-drop*. Untuk menjaga integritas dan kontrol dalam alur kerja proyek, sistem menerapkan mekanisme persetujuan pemindahan tugas seperti yang diilustrasikan pada Gambar 3.7.

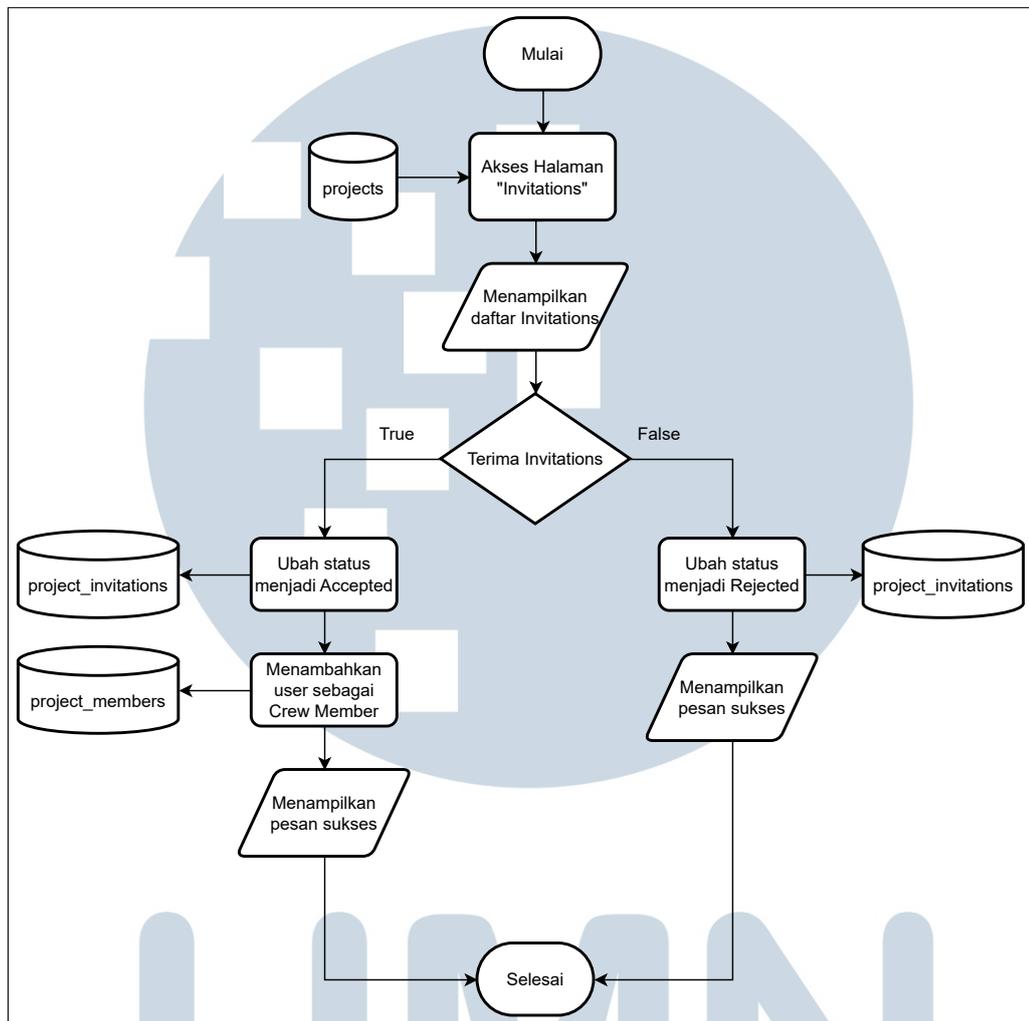


Gambar 3.7. Flowchart Proses Persetujuan Pemindahan Tugas

Jika pengguna yang memindahkan tugas adalah pemilik proyek (*Commander*), status tugas akan langsung diperbarui di *database*. Namun, jika pengguna adalah anggota biasa, pemindahan tugas (terutama ke kolom "Done") akan memicu sebuah permintaan persetujuan (*request move*). Permintaan ini harus ditinjau dan disetujui oleh pemilik proyek sebelum status tugas benar-benar berubah. Setelah sebuah tugas berhasil dipindahkan ke kolom "Done"—baik secara langsung oleh pemilik atau setelah disetujui—anggota yang mengerjakan tugas tersebut dapat mengklaim hadiah XP dan CP.

Alur Penerimaan *invitation* Proyek, fitur ini merupakan jembatan yang menghubungkan pengguna satu dengan yang lainnya dalam sebuah proyek kolaboratif. Alur ini menjelaskan proses dari sisi pengguna yang menerima

invitation untuk bergabung dalam sebuah "Crew Expedition".

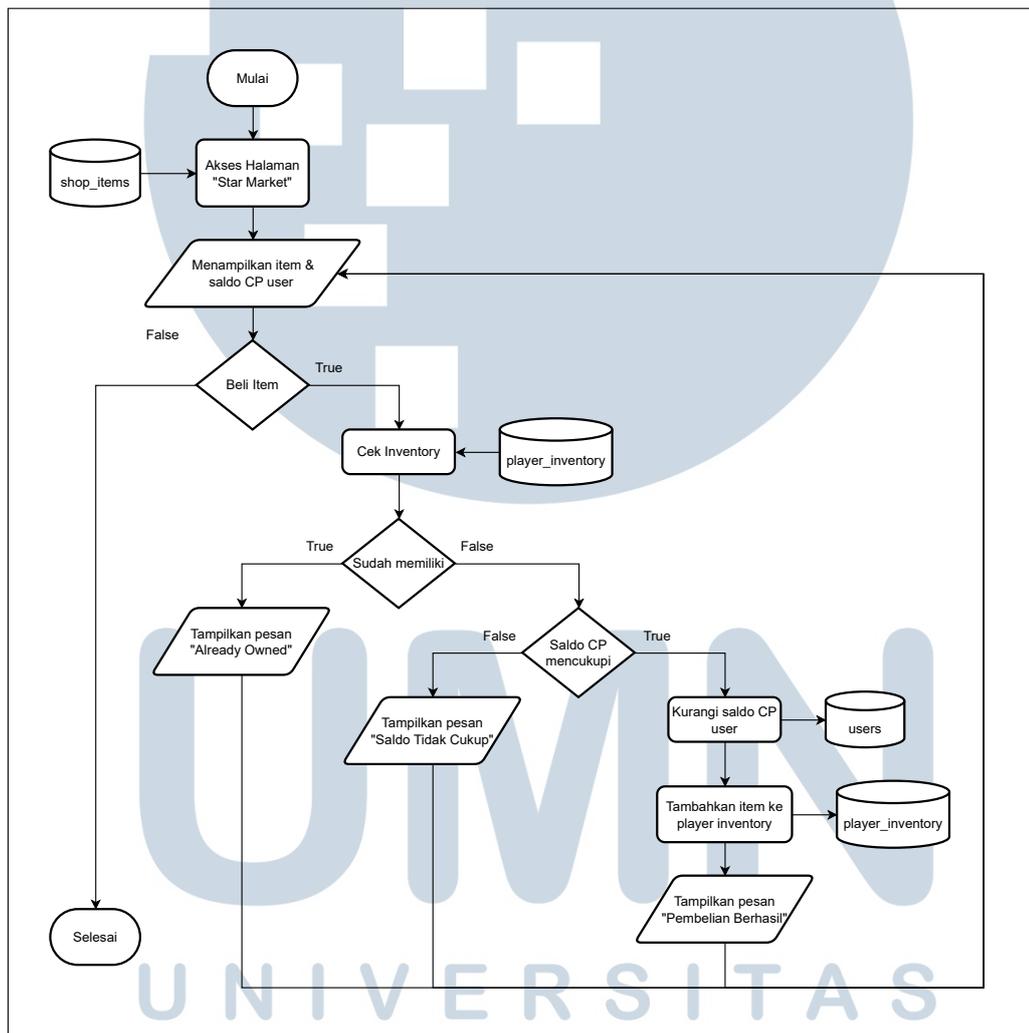


Gambar 3.8. Flowchart Alur Penerimaan invitation Proyek

Gambar 3.8 menjelaskan bagaimana seorang pengguna merespons *invitation* proyek yang dikirimkan oleh pemilik proyek. Ketika pengguna yang telah diundang ke sebuah proyek akan menerima notifikasi atau dapat langsung mengakses halaman "Invitations". Sistem akan menampilkan daftar semua *invitation* yang memiliki status "PENDING". Untuk setiap *invitation*, pengguna diberikan dua opsi: "Accept" (Terima) atau "Reject" (Tolak). Proses Penerimaan, jika pengguna memilih "Accept", sistem akan melakukan dua hal, status *invitation* di *database* akan diubah dari "PENDING" menjadi "ACCEPTED", dan pengguna tersebut akan ditambahkan sebagai anggota baru ke dalam proyek terkait di tabel *ProjectMember*, sehingga mendapatkan akses penuh ke papan Kanban proyek. Lalu Proses Penolakan, jika pengguna memilih "Reject", sistem hanya akan mengubah status

*invitation* menjadi "REJECTED" di *database*. Pengguna tersebut tidak akan ditambahkan ke dalam proyek.

"Star Market" adalah pusat perbelanjaan dalam *website* "SpaceQuest". Di sinilah pengguna dapat membelanjakan *Cosmic Points* (CP) yang telah pengguna kumpulkan dari menyelesaikan tugas dan misi. Item yang tersedia bersifat kosmetik, seperti tema visual dan bingkai avatar, yang berfungsi sebagai hadiah dan simbol status.



Gambar 3.9. Flowchart Modul Toko (Star Market)

Gambar 3.9 mendeskripsikan alur interaksi pengguna saat melakukan pembelian di dalam toko. Prosesnya adalah sebagai berikut:

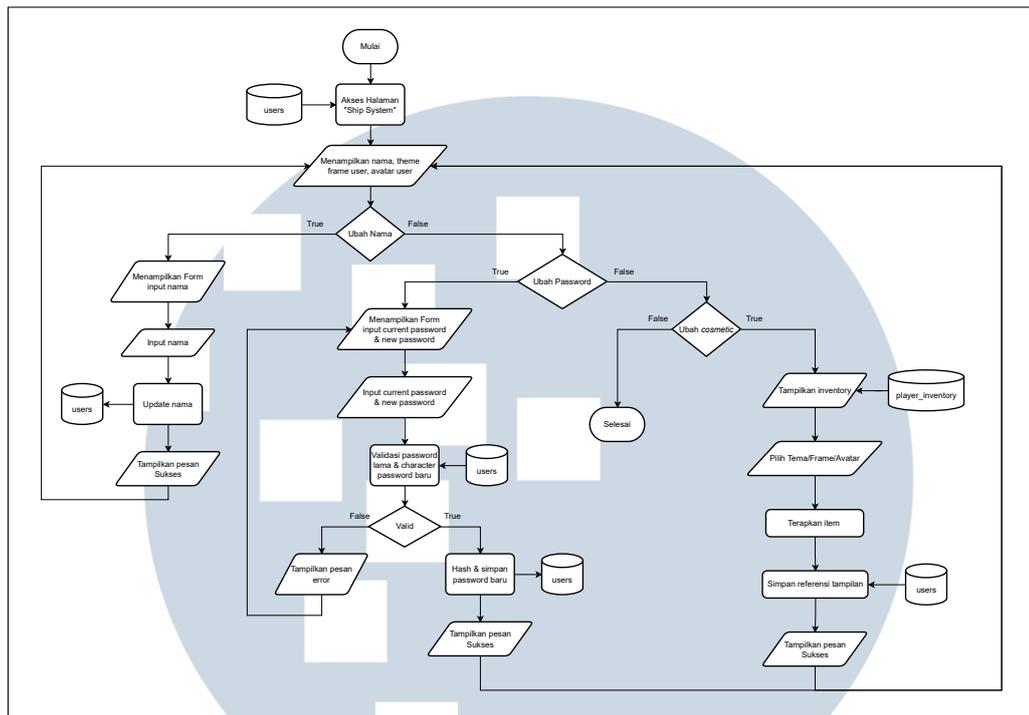
1. Akses Toko, pengguna menavigasi ke halaman "Star Market". Sistem akan mengambil dan menampilkan semua *item* yang tersedia untuk dibeli, beserta

saldo CP yang dimiliki pengguna saat ini.

2. Pemilihan *Item*, pengguna memilih salah satu *item* yang ingin dibeli. Sistem akan menampilkan detail *item* tersebut, termasuk harganya dalam CP.
3. Proses Pembelian, saat pengguna menekan tombol beli, sistem akan melakukan serangkaian validasi. Pertama sistem memeriksa apakah pengguna sudah memiliki *item* tersebut. Jika sudah, proses dibatalkan. Kemudian sistem memeriksa apakah saldo CP pengguna mencukupi untuk melakukan pembelian. Jika saldo tidak cukup, sebuah pesan kesalahan akan ditampilkan. Namun jika saldo mencukupi, transaksi akan dieksekusi, saldo CP pengguna dikurangi sesuai harga *item*, dan *item* tersebut ditambahkan ke dalam inventaris pengguna (*PlayerInventory*).
4. Konfirmasi, setelah pembelian berhasil, antarmuka akan diperbarui untuk menunjukkan bahwa *item* tersebut telah dimiliki.

Modul "*Ship Systems*" berfungsi sebagai pusat kontrol personal bagi pengguna. Di halaman ini, pengguna dapat mengelola informasi profil pengguna, mengubah kredensial keamanan seperti kata sandi, dan menerapkan *item* kustomisasi yang telah pengguna peroleh dari "*Star Market*" untuk mempersonalisasi pengalaman pengguna.





Gambar 3.10. Flowchart Modul Pengaturan

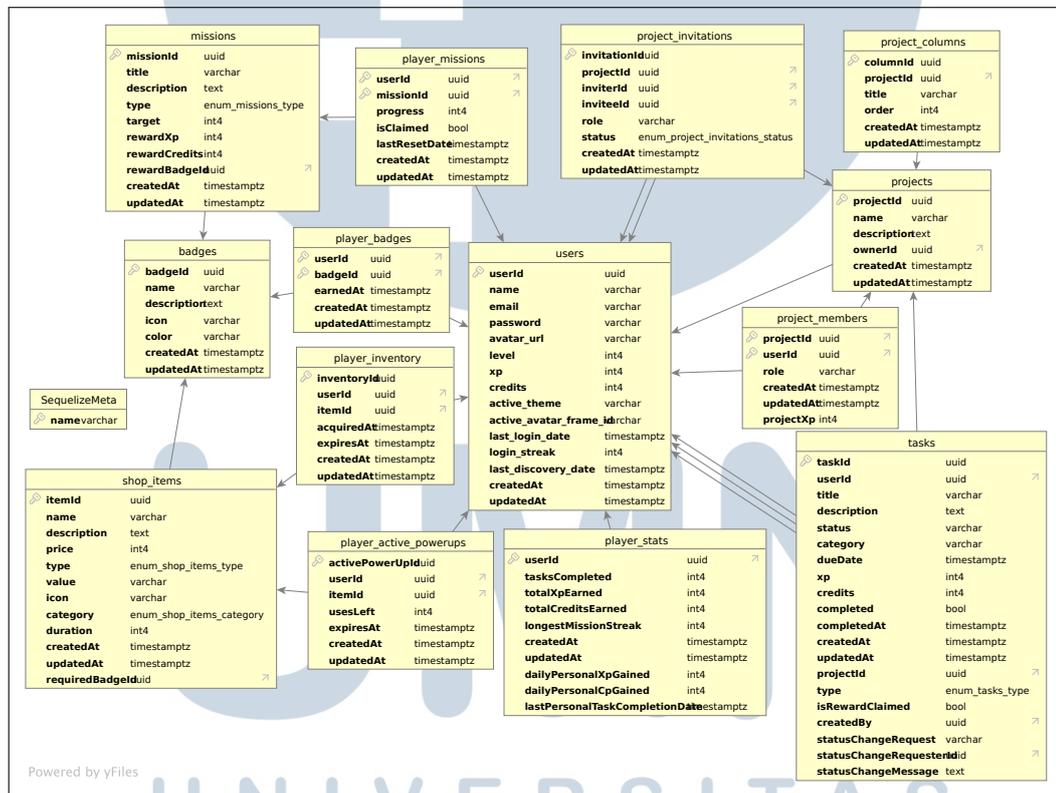
Gambar 3.10 menguraikan alur proses yang tersedia bagi pengguna di dalam halaman pengaturan. Prosesnya adalah sebagai berikut:

1. Akses Halaman Pengaturan, pengguna menavigasi ke halaman "Ship Systems". Sistem akan menampilkan beberapa opsi manajemen akun.
2. Ubah Profil, jika pengguna memilih untuk mengubah profil, sebuah *form* akan ditampilkan yang memungkinkan pengguna untuk memperbarui nama profil pengguna. Perubahan ini akan disimpan ke *database* dan akan langsung terlihat di seluruh *website*.
3. Ubah Kata Sandi, untuk alasan keamanan, pengguna dapat mengubah kata sandi pengguna. Pengguna akan diminta untuk memasukkan kata sandi lama dan kata sandi baru. Sistem akan memvalidasi kata sandi lama sebelum mengenkripsi dan menyimpan kata sandi yang baru.
4. Kustomisasi Tampilan, pengguna dapat melihat dan menerapkan *item-item* kosmetik (seperti Tema atau Avatar atau Bingkai Avatar) yang telah pengguna beli dan miliki di inventaris. Ketika sebuah *item* dipilih dan diterapkan, sistem akan menyimpan preferensi ini dan antarmuka *website* akan diperbarui secara visual sesuai dengan *item* yang aktif.

### 3.2.3 Perancangan Database

Perancangan *database* merupakan tahap krusial untuk mendefinisikan struktur penyimpanan data aplikasi "SpaceQuest". Struktur ini divisualisasikan menggunakan sebuah *Database Scheme*. *Database scheme* memetakan semua entitas (tabel), atribut-atributnya (kolom), beserta relasi yang menghubungkan antar entitas, seperti hubungan antara pengguna, tugas, dan proyek.

Gambar 3.11 menampilkan *database* yang dirancang untuk *website* ini. Desain ini memastikan integritas data, skalabilitas, dan efisiensi dalam pengambilan data yang dibutuhkan oleh setiap fitur aplikasi, mulai dari manajemen tugas hingga sistem gamifikasi.



Gambar 3.11. Database Scheme SpaceQuest

#### A Tabel Users

Tabel Users adalah inti dari sistem yang menyimpan semua informasi krusial terkait akun setiap pengguna. Tabel 3.1 menunjukkan struktur rinci dari tabel Users.

Tabel 3.1. Struktur Tabel Users

Nama Kolom	Tipe Data	Keterangan
id	UUID	<i>Primary Key</i> . Identifier unik untuk setiap pengguna.
name	VARCHAR(255)	Nama lengkap atau nama panggilan pengguna.
email	VARCHAR(255)	Alamat email unik yang digunakan untuk <i>login</i> .
password	VARCHAR(255)	Kata sandi pengguna yang sudah di- <i>hash</i> .
level	INTEGER	Level pengguna saat ini dalam sistem gamifikasi.
xp	INTEGER	Jumlah <i>Experience Points</i> yang dimiliki pengguna.
cp	INTEGER	Jumlah <i>Cosmic Points</i> (mata uang) pengguna.
avatar	VARCHAR(255)	Menyimpan ID atau URL dari avatar yang digunakan.
frame	VARCHAR(255)	Menyimpan ID atau URL dari bingkai avatar.
theme	VARCHAR(255)	Menyimpan ID atau nama tema visual yang aktif.
createdAt	TIMESTAMP	Waktu kapan akun pengguna pertama kali dibuat.
updatedAt	TIMESTAMP	Waktu kapan data pengguna terakhir kali diperbarui.

Setiap pengguna dalam sistem diidentifikasi secara unik oleh kolom id yang berfungsi sebagai *Primary Key*. Untuk identitas dasar, tabel Users menyimpan name yang akan ditampilkan di seluruh aplikasi dan email yang digunakan sebagai kredensial unik untuk *login*. Keamanan akun dijamin oleh kolom password yang menyimpan kata sandi dalam format *hash* untuk melindunginya dari akses tidak sah. Inti dari sistem progresi gamifikasi terletak pada kolom level dan xp (*Experience Points*), di mana xp terus terakumulasi untuk menaikkan level pengguna. Selain itu, terdapat kolom cp (*Cosmic Points*) yang berfungsi sebagai mata uang virtual untuk dibelanjakan. Fitur personalisasi didukung oleh kolom avatar, frame, dan

theme, yang menyimpan penanda *item* kustomisasi yang sedang aktif digunakan oleh pengguna. Terakhir, untuk keperluan audit dan pelacakan, tabel ini dilengkapi dengan *createdAt* yang mencatat waktu pembuatan akun dan *updatedAt* yang secara otomatis menandai waktu setiap kali data pengguna diperbarui.

## B Tabel Tasks

Tabel Tasks bertanggung jawab untuk menyimpan setiap tugas atau "misi" yang dibuat oleh pengguna. Tugas ini bisa bersifat pribadi maupun menjadi bagian dari sebuah proyek tim. Struktur tabel ini dirancang untuk menangkap semua detail yang diperlukan untuk manajemen tugas yang efektif, seperti yang ditunjukkan pada Tabel 3.2.

Tabel 3.2. Struktur Tabel Tasks

Nama Kolom	Tipe Data	Keterangan
id	UUID	<i>Primary Key</i> . Identifier unik untuk setiap tugas.
title	VARCHAR(255)	Judul atau nama dari tugas yang harus dikerjakan.
description	TEXT	Deskripsi opsional untuk memberikan detail lebih lanjut.
status	VARCHAR(50)	Status tugas saat ini, misalnya 'ACTIVE' atau 'COMPLETED'.
userId	UUID	<i>Foreign Key</i> yang terhubung ke tabel Users.
projectId	UUID	<i>Foreign Key</i> yang terhubung ke tabel Projects.
createdAt	TIMESTAMP	Waktu kapan tugas pertama kali dibuat.
updatedAt	TIMESTAMP	Waktu kapan tugas terakhir kali diperbarui.

Setiap baris dalam tabel ini merepresentasikan satu tugas unik yang diidentifikasi oleh kolom *id* sebagai *Primary Key*. Kolom *title* menyimpan nama singkat dari tugas, sementara *description* menyediakan ruang untuk detail yang lebih panjang jika diperlukan. Untuk melacak kemajuan, kolom *status* digunakan untuk menandai apakah sebuah tugas masih aktif atau sudah selesai. Keterhubungan data dijaga melalui dua *Foreign Key* penting: kolom *userId* menghubungkan setiap tugas ke pengguna yang membuatnya, memastikan kepemilikan yang jelas. Sementara itu, kolom *projectId* mengaitkan tugas ke sebuah proyek tim. Jika tugas

tersebut bersifat pribadi dan tidak terikat pada proyek manapun, nilai pada kolom `projectId` akan dibiarkan kosong (*NULL*). Terakhir, kolom `createdAt` dan `updatedAt` secara otomatis mencatat jejak waktu pembuatan dan pembaruan terakhir dari setiap tugas.

### C Tabel Projects

Tabel `Projects` berfungsi sebagai wadah untuk setiap proyek kolaboratif atau "ekspedisi" yang dibuat dalam aplikasi. Tabel ini menyimpan informasi umum mengenai setiap proyek. Detail struktur tabel ini disajikan pada Tabel 3.3.

Tabel 3.3. Struktur Tabel `Projects`

Nama Kolom	Tipe Data	Keterangan
<code>id</code>	UUID	<i>Primary Key</i> . Identifier unik untuk setiap proyek.
<code>name</code>	VARCHAR(255)	Nama dari proyek tim.
<code>description</code>	TEXT	Deskripsi opsional mengenai tujuan atau detail proyek.
<code>ownerId</code>	UUID	<i>Foreign Key</i> yang terhubung ke tabel <code>Users</code> .
<code>createdAt</code>	TIMESTAMP	Waktu kapan proyek pertama kali dibuat.
<code>updatedAt</code>	TIMESTAMP	Waktu kapan proyek terakhir kali diperbarui.

Setiap proyek yang dibuat diidentifikasi secara unik oleh kolom `id` sebagai *Primary Key*. Informasi dasar proyek disimpan dalam kolom `name` sebagai judul dan `description` untuk penjelasan yang lebih rinci. Aspek kepemilikan didefinisikan oleh kolom `ownerId`, sebuah *Foreign Key* yang secara langsung menunjuk ke `id` pengguna di tabel `Users` yang membuat dan bertanggung jawab atas proyek tersebut. Seperti tabel lainnya, tabel ini juga dilengkapi dengan kolom `createdAt` dan `updatedAt` untuk melacak siklus hidup data proyek, dari saat pertama kali dibuat hingga pembaruan terakhir.

### D Tabel `Project_Members`

Tabel `Project_Members` adalah sebuah tabel penghubung (*join table*) yang memiliki peran krusial dalam mengelola hubungan *many-to-many* antara pengguna dan proyek. Tabel 3.4 secara efektif mencatat siapa saja anggota dari setiap proyek yang ada.

Tabel 3.4. Struktur Tabel Project\_Members

Nama Kolom	Tipe Data	Keterangan
userId	UUID	<i>Foreign Key</i> yang terhubung ke tabel Users.
projectId	UUID	<i>Foreign Key</i> yang terhubung ke tabel Projects.
role	VARCHAR(50)	Peran pengguna dalam proyek, misalnya 'OWNER' atau 'MEMBER'.
xp_in_project	INTEGER	Jumlah XP yang disumbangkan pengguna untuk proyek ini.
createdAt	TIMESTAMP	Waktu kapan pengguna bergabung dengan proyek.
updatedAt	TIMESTAMP	Waktu kapan data keanggotaan terakhir kali diperbarui.

Tabel 3.4 tidak memiliki *Primary Key* tunggal, melainkan menggunakan kombinasi dari kolom `userId` dan `projectId` sebagai kunci komposit untuk memastikan bahwa seorang pengguna hanya dapat bergabung dalam satu proyek yang sama sebanyak satu kali. Kolom `userId` menunjuk ke pengguna yang menjadi anggota, sementara `projectId` menunjuk ke proyek tempat ia bergabung. Peran spesifik pengguna dalam proyek tersebut, seperti menjadi pemilik atau anggota biasa, didefinisikan dalam kolom `role`. Untuk mendukung gamifikasi dalam konteks tim, kolom `xp_in_project` secara khusus melacak total poin yang telah disumbangkan oleh anggota untuk proyek tersebut, yang datanya digunakan untuk papan peringkat internal proyek. Seperti tabel lainnya, `createdAt` dan `updatedAt` juga disertakan untuk mencatat waktu bergabungnya anggota dan pembaruan data terakhir.

### E Tabel Shop\_Items

Tabel `Shop_Items` berfungsi sebagai katalog untuk semua *item* virtual yang dapat dibeli oleh pengguna di "Star Market". Tabel ini menyimpan informasi detail mengenai setiap *item*, mulai dari nama hingga harganya. Strukturnya disajikan pada Tabel 3.5.

Setiap *item* yang tersedia di toko diidentifikasi secara unik oleh kolom `id` sebagai *Primary Key*. Informasi yang ditampilkan kepada pengguna di antarmuka toko disimpan dalam kolom `name` dan `description`. Harga *item*, yang harus dibayar menggunakan mata uang dalam *game*, ditentukan oleh nilai pada

Tabel 3.5. Struktur Tabel Shop\_Items

Nama Kolom	Type Data	Keterangan
id	UUID	<i>Primary Key</i> . Identifier unik untuk setiap <i>item</i> .
name	VARCHAR(255)	Nama dari <i>item</i> yang ditampilkan di toko.
description	TEXT	Deskripsi singkat mengenai <i>item</i> tersebut.
price	INTEGER	Harga <i>item</i> dalam satuan <i>Cosmic Points</i> (CP).
type	VARCHAR(50)	Jenis <i>item</i> , misalnya 'AVATAR', 'FRAME', atau 'THEME'.
asset_key	VARCHAR(255)	Kunci atau penanda aset visual untuk <i>item</i> di sisi <i>frontend</i> .
createdAt	TIMESTAMP	Waktu kapan <i>item</i> pertama kali ditambahkan ke toko.
updatedAt	TIMESTAMP	Waktu kapan detail <i>item</i> terakhir kali diperbarui.

kolom price. Untuk membedakan fungsionalitas *item*, kolom type digunakan untuk mengkategorikannya, seperti avatar, bingkai, atau tema. Kolom asset\_key menyimpan sebuah penanda unik yang digunakan oleh aplikasi *frontend* untuk menampilkan aset visual yang sesuai dengan *item* tersebut. Terakhir, kolom createdAt dan updatedAt mencatat kapan setiap *item* ditambahkan ke dalam sistem dan kapan terakhir kali informasinya diubah.

## F Tabel Player\_Inventory

Tabel Player\_Inventory adalah tabel penghubung (*join table*) yang berfungsi untuk melacak *item* apa saja yang telah dimiliki oleh setiap pengguna. Tabel ini secara efektif menciptakan hubungan *many-to-many* antara tabel Users dan tabel Shop\_Items. Strukturnya disajikan pada Tabel 3.6.

Tabel ini secara sederhana menjawab pertanyaan "pengguna mana memiliki *item* apa". Setiap baris menandakan bahwa seorang pengguna tertentu telah berhasil memperoleh sebuah *item* dari toko. Kolom userId menunjuk ke pengguna yang memiliki *item*, sementara kolom itemId menunjuk ke *item* yang dimiliki dari tabel Shop\_Items. Kombinasi kedua kolom ini berfungsi sebagai kunci komposit untuk memastikan bahwa setiap entri kepemilikan adalah unik. Kolom createdAt dan updatedAt digunakan untuk mencatat kapan *item* tersebut diperoleh oleh pengguna.

Tabel 3.6. Struktur Tabel Player\_Inventory

Nama Kolom	Tipe Data	Keterangan
userId	UUID	<i>Foreign Key</i> yang terhubung ke tabel Users.
itemId	UUID	<i>Foreign Key</i> yang terhubung ke tabel Shop_Items.
createdAt	TIMESTAMP	Waktu kapan pengguna memperoleh <i>item</i> .
updatedAt	TIMESTAMP	Waktu kapan data kepemilikan terakhir kali diperbarui.

## G Tabel Badges

Tabel Badges berfungsi sebagai sebuah kamus atau definisi untuk semua lencana pencapaian yang bisa diperoleh oleh pengguna. Setiap baris dalam tabel ini merepresentasikan satu jenis pencapaian unik yang dapat diraih dalam aplikasi. Strukturnya disajikan pada Tabel 3.7.

Tabel 3.7. Struktur Tabel Badges

Nama Kolom	Tipe Data	Keterangan
id	UUID	<i>Primary Key</i> . Identifier unik untuk setiap lencana.
name	VARCHAR(255)	Nama dari lencana pencapaian.
description	TEXT	Deskripsi mengenai kriteria untuk mendapatkan lencana.
icon_key	VARCHAR(255)	Kunci atau penanda aset visual untuk ikon lencana.
createdAt	TIMESTAMP	Waktu kapan lencana pertama kali dibuat dalam sistem.
updatedAt	TIMESTAMP	Waktu kapan detail lencana terakhir kali diperbarui.

Setiap jenis lencana yang ada diidentifikasi secara unik oleh kolom id sebagai *Primary Key*. Kolom name menyimpan nama pencapaian yang akan ditampilkan kepada pengguna, misalnya "Misionis Pertama", sementara kolom description menjelaskan bagaimana cara atau syarat untuk memperoleh lencana tersebut. Untuk aspek visual, kolom icon\_key menyimpan sebuah penanda unik yang digunakan oleh aplikasi untuk menampilkan gambar atau ikon lencana yang sesuai. Terakhir, kolom createdAt dan updatedAt mencatat kapan setiap definisi lencana ini ditambahkan ke dalam sistem dan kapan terakhir kali informasinya diubah.

## H Tabel Player\_Badges

Tabel Player\_Badges adalah sebuah tabel penghubung (*join table*) yang secara spesifik mencatat lencana mana saja yang telah berhasil diperoleh oleh setiap pengguna. Tabel ini menciptakan hubungan *many-to-many* antara tabel Users dan tabel Badges. Strukturnya disajikan pada Tabel 3.8.

Tabel 3.8. Struktur Tabel Player\_Badges

Nama Kolom	Tipe Data	Keterangan
userId	UUID	<i>Foreign Key</i> yang terhubung ke tabel Users.
badgeId	UUID	<i>Foreign Key</i> yang terhubung ke tabel Badges.
createdAt	TIMESTAMP	Waktu kapan pengguna memperoleh lencana.
updatedAt	TIMESTAMP	Waktu kapan data perolehan terakhir kali diperbarui.

Tabel ini memiliki fungsi yang lugas, yaitu untuk menandai pencapaian. Setiap baris dalam tabel ini menunjukkan bahwa seorang pengguna telah memenuhi kriteria untuk mendapatkan sebuah lencana. Kolom *userId* menunjuk ke pengguna yang menerima lencana, sementara kolom *badgeId* menunjuk ke jenis lencana yang diperoleh dari tabel Badges. Kombinasi kedua kolom ini bersifat unik untuk memastikan seorang pengguna hanya tercatat satu kali untuk setiap jenis lencana yang diraih. Kolom *createdAt* menjadi sangat penting di sini karena menandai momen pasti kapan sebuah pencapaian berhasil diraih oleh pengguna.

## I Tabel Missions

Tabel Missions berfungsi untuk menyimpan definisi dari berbagai tantangan atau misi berbasis sistem yang dapat diselesaikan oleh pengguna untuk mendapatkan hadiah tambahan. Berbeda dengan tugas biasa, misi ini umumnya bersifat *metagame*, seperti "Selesaikan 10 tugas" atau "Bergabunglah dalam sebuah proyek". Strukturnya disajikan pada Tabel 3.9.

Setiap misi sistem diidentifikasi secara unik oleh kolom *id* sebagai *Primary Key*. Kolom *name* dan *description* digunakan untuk menampilkan informasi mengenai tantangan kepada pengguna. Logika inti dari misi ini didefinisikan oleh kolom *type*, yang memberitahu sistem jenis aksi apa yang harus dilacak (misalnya,

Tabel 3.9. Struktur Tabel Missions

Nama Kolom	Tipe Data	Keterangan
id	UUID	<i>Primary Key</i> . Identifier unik untuk setiap misi.
name	VARCHAR(255)	Nama dari misi atau tantangan.
description	TEXT	Deskripsi mengenai objektif yang harus dicapai.
type	VARCHAR(50)	Tipe misi yang menentukan logika penyelesaiannya.
goal	INTEGER	Target angka yang harus dicapai (misalnya, 10 untuk 10 tugas).
reward_cp	INTEGER	Jumlah <i>Cosmic Points</i> yang diberikan sebagai hadiah.
reward_xp	INTEGER	Jumlah <i>Experience Points</i> yang diberikan sebagai hadiah.
createdAt	TIMESTAMP	Waktu kapan misi pertama kali dibuat dalam sistem.
updatedAt	TIMESTAMP	Waktu kapan detail misi terakhir kali diperbarui.

‘COMPLETE\_TASK‘), dan kolom goal, yang menentukan target kuantitatif yang harus dicapai pengguna untuk menyelesaikan misi. Sebagai insentif, kolom reward\_cp dan reward\_xp menentukan jumlah mata uang dan poin pengalaman yang akan diterima pengguna sebagai hadiah setelah berhasil menyelesaikan misi. Terakhir, kolom createdAt dan updatedAt mencatat kapan setiap definisi misi ini ditambahkan ke dalam sistem dan kapan terakhir kali informasinya diubah.

## J Tabel Player\_Missions

Tabel Player\_Missions adalah tabel penghubung yang secara dinamis melacak progres setiap pengguna terhadap misi-misi yang tersedia. Tabel ini menyimpan data kemajuan individual dan status penyelesaian untuk setiap misi yang sedang atau sudah dikerjakan oleh pengguna. Strukturnya disajikan pada Tabel 3.10.

Tabel ini menghubungkan pengguna dan misi melalui kolom userId dan missionId, yang kombinasi keduanya berfungsi sebagai kunci komposit. Kolom yang paling dinamis adalah progress, yang nilainya akan terus diperbarui oleh sistem setiap kali pengguna melakukan aksi yang relevan dengan tipe misi tersebut. Ketika nilai pada kolom progress telah mencapai atau melampaui nilai ‘goal‘ yang

Tabel 3.10. Struktur Tabel Player\_Missions

Nama Kolom	Tipe Data	Keterangan
userId	UUID	<i>Foreign Key</i> yang terhubung ke tabel Users.
missionId	UUID	<i>Foreign Key</i> yang terhubung ke tabel Missions.
progress	INTEGER	Angka kemajuan pengguna saat ini terhadap goal misi.
is_completed	BOOLEAN	Status yang menandakan apakah misi sudah selesai.
createdAt	TIMESTAMP	Waktu kapan pengguna memulai progres pada misi.
updatedAt	TIMESTAMP	Waktu kapan progres misi terakhir kali diperbarui.

didefinisikan di tabel Missions, kolom is\_completed akan diubah menjadi 'true' untuk menandakan bahwa misi tersebut telah berhasil diselesaikan oleh pengguna. Kolom createdAt dan updatedAt melacak kapan pengguna memulai sebuah misi dan kapan kemajuan terakhir dicatat.

## K Tabel Project\_Invitations

Tabel 3.11. Struktur Tabel Project\_Invitations

Nama Kolom	Tipe Data	Keterangan
id	UUID	<i>Primary Key</i> . Identifier unik untuk setiap undangan.
projectId	UUID	<i>Foreign Key</i> ke proyek yang ditawarkan.
inviterId	UUID	<i>Foreign Key</i> ke pengguna yang mengirim undangan.
inviteeId	UUID	<i>Foreign Key</i> ke pengguna yang diundang.
status	VARCHAR(50)	Status undangan, misalnya 'PENDING', 'ACCEPTED', 'DECLINED'.
Lanjut ke halaman berikutnya		

Tabel 3.11 (lanjutan)

Nama Kolom	Tipe Data	Keterangan
createdAt	TIMESTAMP	Waktu kapan undangan pertama kali dikirim.
updatedAt	TIMESTAMP	Waktu kapan status undangan terakhir kali diperbarui.

Tabel `Project_Invitations` mengelola sistem undangan kolaborasi dalam aplikasi. Setiap baris dalam tabel ini merepresentasikan sebuah undangan yang dikirim dari satu pengguna ke pengguna lain untuk bergabung dalam sebuah proyek. Strukturnya disajikan pada Tabel 3.11.

Setiap undangan diidentifikasi secara unik oleh kolom `id` sebagai *Primary Key*. Tabel ini menghubungkan tiga entitas utama melalui *Foreign Key*: kolom `projectId` menunjuk ke proyek yang relevan, kolom `inviterId` menunjuk ke pengguna yang mengirimkan undangan, dan kolom `inviteeId` menunjuk ke pengguna yang menerima undangan. Alur kerja undangan dikelola oleh kolom `status`, yang melacak apakah sebuah undangan masih menunggu tanggapan, sudah diterima, atau sudah ditolak. Kolom `createdAt` dan `updatedAt` digunakan untuk mencatat waktu pengiriman undangan dan waktu respons terakhir dari pengguna yang diundang.

## L Tabel `Project.Columns`

Tabel `Project.Columns` mendefinisikan struktur kolom pada papan Kanban untuk setiap proyek. Secara *default*, setiap proyek memiliki kolom standar seperti "To Do", "In Progress", dan "Done", namun tabel ini memungkinkan adanya kustomisasi di masa depan. Strukturnya disajikan pada Tabel 3.12.

Tabel 3.12. Struktur Tabel `Project.Columns`

Nama Kolom	Tipe Data	Keterangan
<code>id</code>	UUID	<i>Primary Key</i> . Identifier unik untuk setiap kolom.
<code>name</code>	VARCHAR(255)	Nama dari kolom yang ditampilkan di papan Kanban.
Lanjut ke halaman berikutnya		

Tabel 3.12 (lanjutan)

Nama Kolom	Tipe Data	Keterangan
projectId	UUID	<i>Foreign Key</i> ke proyek tempat kolom ini berada.
order	INTEGER	Urutan posisi kolom dari kiri ke kanan.
createdAt	TIMESTAMP	Waktu kapan kolom pertama kali dibuat.
updatedAt	TIMESTAMP	Waktu kapan kolom terakhir kali diperbarui.

Setiap kolom pada papan Kanban diidentifikasi secara unik oleh kolom id sebagai *Primary Key*. Kolom name menyimpan judul yang akan ditampilkan di kepala kolom, misalnya "In Progress". Keterhubungan dengan proyek spesifik diatur oleh kolom projectId, sebuah *Foreign Key* yang memastikan setiap kolom hanya dimiliki oleh satu proyek. Untuk menjaga urutan visual pada antarmuka, kolom order digunakan untuk menentukan posisi relatif setiap kolom, misalnya kolom dengan order 1 akan tampil paling kiri. Kolom createdAt dan updatedAt mencatat jejak waktu pembuatan dan pembaruan terakhir dari setiap kolom Kanban.

## M Tabel Player.Stats

Tabel Player.Stats adalah tabel serbaguna yang dirancang untuk menyimpan berbagai macam statistik atau data metrik pengguna yang bersifat dinamis. Daripada menambahkan banyak kolom baru ke tabel Users, tabel ini menggunakan pendekatan kunci-nilai (*key-value*) untuk melacak data seperti *login streak* atau jumlah tugas yang diselesaikan. Strukturnya disajikan pada Tabel 3.13.

Tabel 3.13. Struktur Tabel Player.Stats

Nama Kolom	Tipe Data	Keterangan
id	UUID	<i>Primary Key</i> . Identifier unik untuk setiap entri statistik.
userId	UUID	<i>Foreign Key</i> yang terhubung ke tabel Users.
value	VARCHAR(255)	Nilai dari statistik yang disimpan.
Lanjut ke halaman berikutnya		

Tabel 3.13 (lanjutan)

Nama Kolom	Tipe Data	Keterangan
createdAt	TIMESTAMP	Waktu kapan statistik pertama kali dicatat.
updatedAt	TIMESTAMP	Waktu kapan nilai statistik terakhir kali diperbarui.

Setiap entri statistik unik diidentifikasi oleh kolom id sebagai *Primary Key*. Kolom *userId* menghubungkan setiap statistik ke pengguna yang bersangkutan. Fleksibilitas tabel ini berasal dari pasangan kolom *stat\_key* dan *value*. Kolom *stat\_key* berfungsi sebagai nama atau label dari metrik yang dilacak, seperti 'TOTAL\_TASKS\_COMPLETED', sementara kolom *value* menyimpan nilai aktual dari metrik tersebut. Pendekatan ini memungkinkan sistem untuk dengan mudah menambahkan pelacakan statistik baru di masa depan tanpa perlu mengubah struktur database. Kolom *createdAt* dan *updatedAt* digunakan untuk mencatat kapan sebuah statistik mulai dilacak dan kapan nilai terakhirnya diperbarui.

## N Tabel Active\_Powerups

Tabel *Active\_Powerups* secara spesifik melacak *power-up* atau efek sementara yang sedang aktif untuk setiap pengguna. Tabel ini memastikan bahwa efek seperti bonus XP atau CP diterapkan dengan benar dan hanya berlaku selama durasi yang telah ditentukan. Strukturnya disajikan pada Tabel 3.14.

Tabel 3.14. Struktur Tabel *Active\_Powerups*

Nama Kolom	Tipe Data	Keterangan
id	UUID	<i>Primary Key</i> . Identifier unik untuk setiap efek <i>power-up</i> yang aktif.
userId	UUID	<i>Foreign Key</i> yang terhubung ke tabel <i>Users</i> .
powerup_type	VARCHAR(255)	Jenis <i>power-up</i> yang sedang aktif, misalnya 'XP_BOOST'.
expires_at	TIMESTAMP	Waktu pasti kapan efek <i>power-up</i> akan berakhir.
createdAt	TIMESTAMP	Waktu kapan <i>power-up</i> pertama kali diaktifkan.
updatedAt	TIMESTAMP	Waktu kapan data <i>power-up</i> terakhir kali diperbarui.

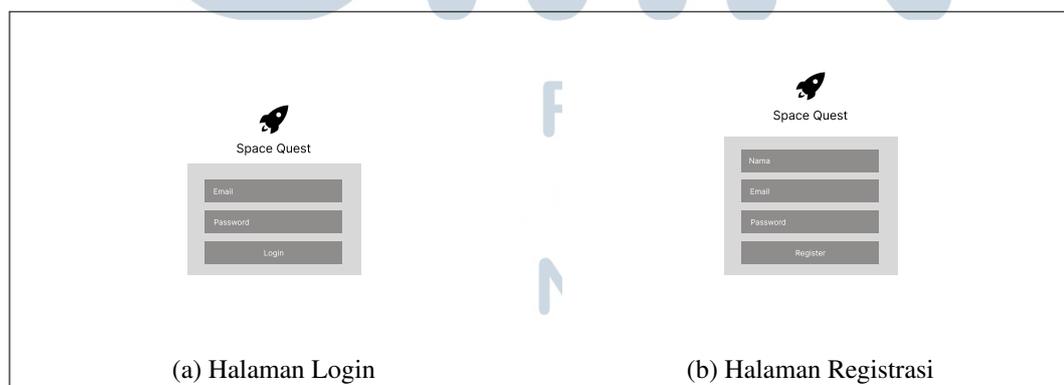
Setiap efek *power-up* yang sedang berjalan diidentifikasi secara unik oleh kolom id sebagai *Primary Key*. Kolom *userId* menghubungkan setiap efek ke pengguna yang bersangkutan. Jenis efek yang aktif ditentukan oleh kolom *powerup\_type*, yang nilainya akan digunakan oleh sistem untuk menerapkan logika bonus yang sesuai, misalnya menggandakan perolehan XP. Kolom yang paling krusial dalam tabel ini adalah *expires\_at*, yang menyimpan *timestamp* masa kedaluwarsa efek. Sistem akan secara berkala memeriksa kolom ini untuk menonaktifkan bonus yang waktunya telah habis. Terakhir, kolom *createdAt* dan *updatedAt* digunakan untuk mencatat kapan sebuah *power-up* diaktifkan dan kapan terakhir kali informasinya diperbarui.

### 3.2.4 Perancangan Antarmuka Pengguna

Perancangan tampilan antarmuka (UI) bertujuan untuk memvisualisasikan tata letak fungsional dari setiap halaman pada website "SpaceQuest". Desain ini difokuskan pada pengalaman pengguna (UX) dengan menyajikan alur interaksi yang logis dan intuitif. Berikut adalah perancangan antarmuka untuk setiap modul utama website.

#### A Halaman Onboarding dan Autentikasi

Halaman Onboarding dan Autentikasi merupakan gerbang pertama yang akan dilalui oleh pengguna. Desain antarmuka pada bagian ini dirancang untuk memberikan proses masuk dan pendaftaran yang jelas, cepat, dan tidak membingungkan, sehingga pengguna dapat dengan mudah memulai perjalanannya di dalam "SpaceQuest".



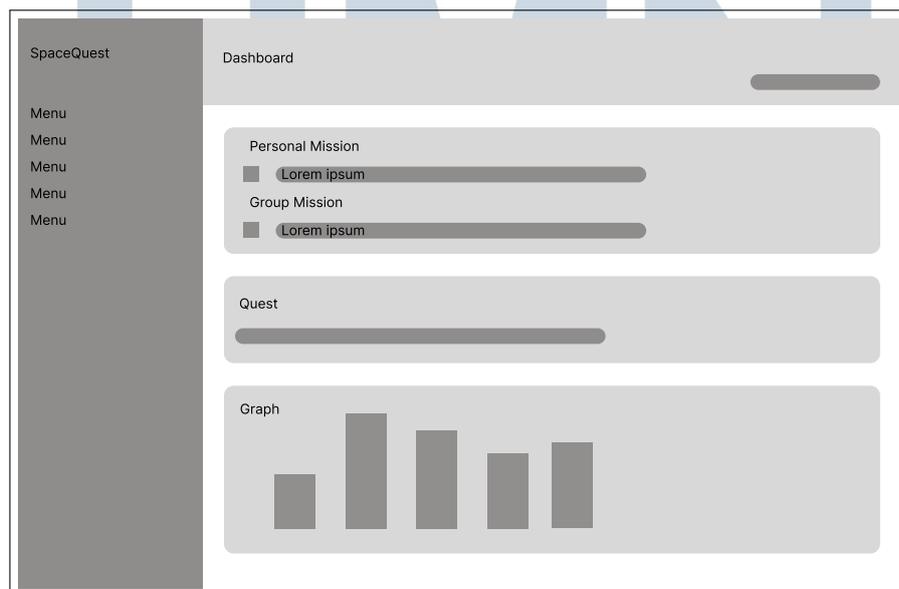
Gambar 3.12. Desain Antarmuka Halaman Onboarding dan Autentikasi

Gambar 3.12 menampilkan desain antarmuka untuk proses autentikasi pengguna. Pada Gambar 3.12a, Halaman Login dirancang dengan minimalis untuk meminimalisir distraksi. Halaman ini menyediakan kolom input untuk email dan kata sandi, yang merupakan kredensial utama pengguna. Terdapat sebuah tombol "Login" yang jelas sebagai aksi utama dan sebuah tautan navigasi ke Halaman Registrasi bagi pengguna baru.

Sementara itu, pada Gambar 3.12b, Halaman Registrasi dirancang untuk proses pembuatan akun yang terstruktur. Pengguna diminta untuk mengisi informasi esensial seperti nama, email, dan kata sandi. Untuk memastikan tidak ada kesalahan pengetikan kata sandi, disediakan pula kolom konfirmasi. Alur ini dirancang untuk menjadi seintuitif mungkin, memandu pengguna baru untuk dapat segera memiliki akun dan masuk ke dalam ekosistem website "SpaceQuest".

## B Halaman Utama (Starship Dashboard)

Halaman Utama atau "Starship Dashboard" adalah pusat komando bagi pengguna. Halaman ini dirancang untuk memberikan gambaran umum yang cepat dan informatif mengenai status dan progres pengguna sesaat setelah pengguna login. Tampilan ini berfungsi sebagai jangkar, memberikan akses mudah ke semua fitur utama website sambil tetap menonjolkan elemen-elemen gamifikasi yang esensial untuk menjaga motivasi pengguna.



Gambar 3.13. Desain Antarmuka Halaman Utama (Starship Dashboard)

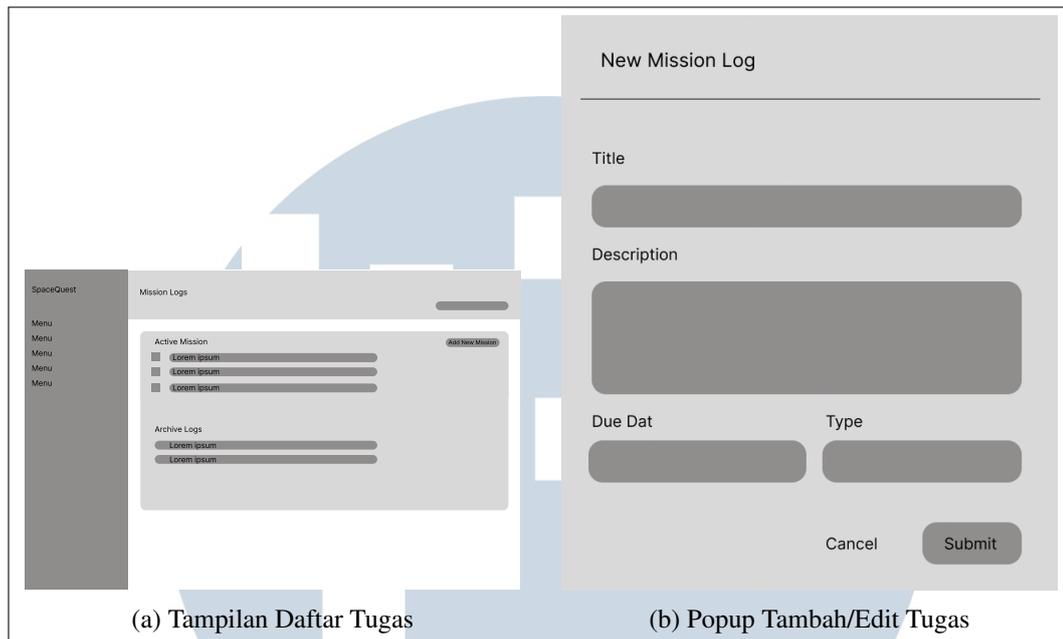
Gambar 3.13 menampilkan perancangan tata letak untuk Halaman Utama. Desain ini terbagi menjadi tiga area utama:

1. Header Website, bagian atas antarmuka secara konsisten menampilkan informasi vital terkait progresi gamifikasi pengguna. Ini mencakup nama dan avatar pengguna, progres level saat ini yang direpresentasikan oleh *XP Bar*, serta saldo mata uang virtual *Cosmic Points (CP)*. Keberadaan elemen-elemen ini di setiap halaman utama memastikan pengguna selalu sadar akan pencapaian dan sumber daya yang pengguna miliki.
2. Area Konten Utama, bagian tengah halaman didedikasikan untuk menampilkan informasi yang paling relevan bagi pengguna pada sesi tersebut. Ini dapat mencakup ringkasan tugas pribadi yang harus diselesaikan hari ini, notifikasi penting dari proyek tim, tombol aksi untuk mengklaim hadiah harian seperti *Daily Discovery*, dan juga chart penyelesaian tugas.
3. Menu Navigasi Utama, di sisi kiri terdapat *sidebar* yang berfungsi sebagai sistem navigasi utama website. Menu ini menyediakan akses langsung ke semua modul penting seperti "Mission Logs", "Crew Expeditions", "Star Market", dan "Ship Systems", memungkinkan pengguna untuk berpindah antar fitur dengan mudah dan efisien.

### **C Halaman Manajemen Tugas Pribadi (Mission Logs)**

Halaman "Mission Logs" adalah ruang kerja utama bagi pengguna untuk mengelola tugas-tugas individual pengguna. Desain antarmuka di sini difokuskan untuk memberikan kemudahan dalam melihat, menambah, dan menyelesaikan tugas. Tampilan yang bersih dan fungsional sangat penting untuk mendukung produktivitas, sementara integrasi umpan balik gamifikasi yang subtil tetap dipertahankan.

Gambar 3.14 menampilkan desain antarmuka untuk modul manajemen tugas pribadi. Pada Gambar 3.14a, antarmuka utama "Mission Logs" menyajikan daftar tugas yang terorganisir dengan baik. Untuk membedakan progres, tugas-tugas dikelompokkan ke dalam dua tab atau bagian: "Aktif" dan "Selesai". Setiap item tugas menampilkan informasi esensial seperti nama tugas dan tombol untuk menandainya sebagai selesai. Terdapat juga tombol aksi utama "Tambah Tugas Baru" yang mudah diakses untuk mendorong pengguna menambahkan tugas baru.



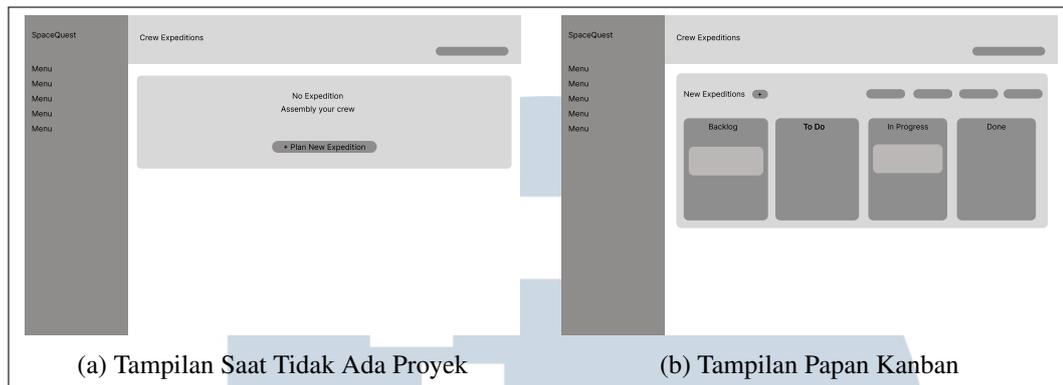
Gambar 3.14. Desain Antarmuka Modul Manajemen Tugas Pribadi

Ketika pengguna memilih untuk menambah atau mengedit tugas, sebuah *popup* atau *modal* akan muncul seperti yang ditunjukkan pada Gambar 3.14b. Desain ini bertujuan agar pengguna tetap berada dalam konteks halaman utama tanpa harus berpindah ke halaman lain. *Modal* ini berisi formulir sederhana untuk memasukkan detail tugas, seperti nama, deskripsi, dan tanggal jatuh tempo, beserta tombol untuk menyimpan atau membatalkan aksi. Alur ini memastikan proses manajemen tugas berjalan dengan lancar dan efisien.

#### D Halaman Proyek Tim (Crew Expeditions)

Modul "Crew Expeditions" adalah inti dari fitur kolaborasi "SpaceQuest", dirancang untuk memfasilitasi kerja sama tim melalui manajemen proyek yang visual dan interaktif. Perancangan antarmuka pada bagian ini menekankan pada panduan bagi pengguna baru dan kejelasan dalam menampilkan alur kerja pada papan Kanban.

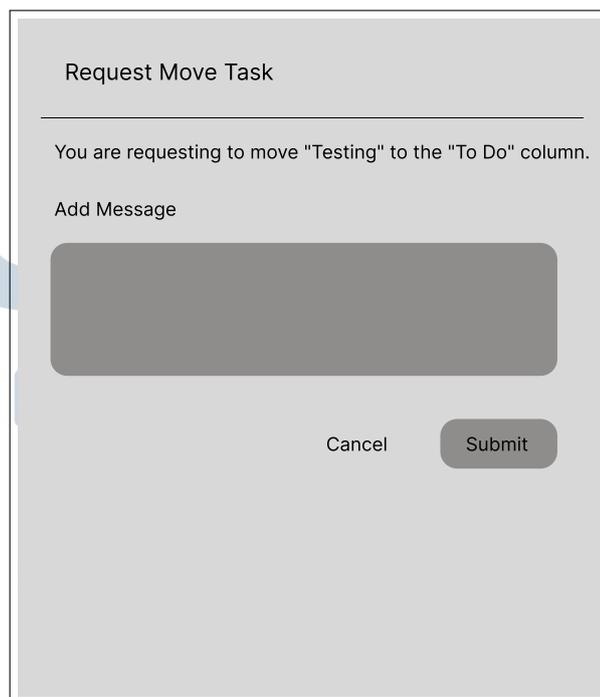
Gambar 3.15 menampilkan desain antarmuka untuk modul proyek tim. Pada Gambar 3.15a, ditampilkan desain untuk kondisi *empty state*, yaitu ketika pengguna belum bergabung atau membuat proyek sama sekali. Desain ini sangat penting untuk memberikan pengalaman pertama yang baik, dengan menyediakan teks panduan dan tombol "Buat Proyek Baru" yang menonjol. Tujuannya adalah untuk



Gambar 3.15. Desain Antarmuka Modul Proyek Tim

secara proaktif mengarahkan pengguna untuk melakukan aksi pertama pengguna dalam menggunakan fitur kolaboratif.

Setelah pengguna membuat atau bergabung dengan sebuah proyek, pengguna akan disajikan Tampilan Papan Kanban, seperti yang diilustrasikan pada Gambar 3.15b. Halaman ini adalah pusat visualisasi alur kerja tim. Desainnya menampilkan beberapa kolom vertikal yang merepresentasikan tahapan tugas (misalnya, "Backlog", "To Do", "In Progress", "Done"). Setiap tugas direpresentasikan sebagai kartu yang dapat dipindahkan antar kolom, memberikan transparansi penuh atas progres setiap tugas dalam proyek.

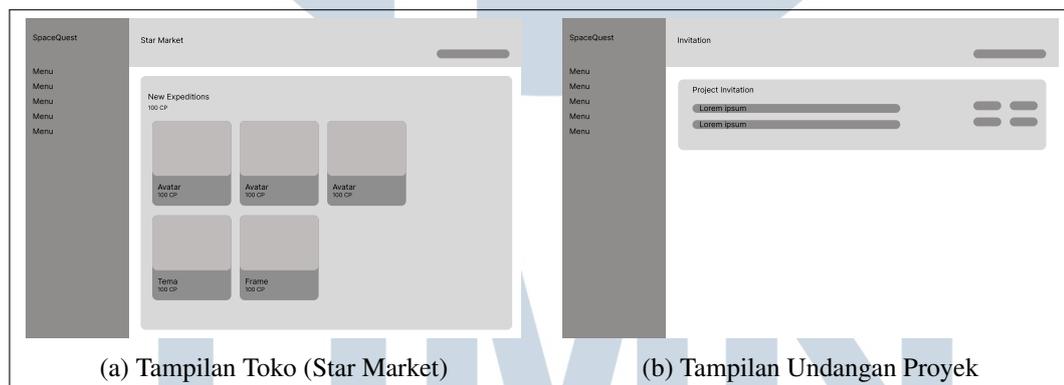


Gambar 3.16. Desain Antarmuka Popup Permintaan Pindah Tugas

Gambar 3.16 menunjukkan desain untuk interaksi kunci di dalam papan Kanban, yaitu alur persetujuan. Ketika seorang anggota tim (bukan pemilik proyek) mencoba memindahkan sebuah tugas ke kolom lain, sebuah *modal* konfirmasi akan muncul. Ini adalah bagian dari mekanisme kontrol, di mana anggota diminta untuk mengirimkan permintaan pindah tugas yang kemudian harus disetujui oleh pemilik proyek. Desainnya dirancang agar jelas, memberikan instruksi kepada pengguna untuk mengirim permintaan tersebut, yang memastikan alur kerja tetap terkelola dengan baik.

## E Halaman Toko dan invitation

Selain fitur produktivitas utama, "SpaceQuest" juga dilengkapi dengan berbagai halaman pendukung yang bertujuan untuk memperkaya pengalaman gamifikasi dan interaksi sosial. Halaman-halaman ini mencakup "Star Market" sebagai pusat ekonomi virtual dan halaman "Invitations" untuk mengelola koneksi sosial dalam proyek.



Gambar 3.17. Desain Antarmuka Halaman Toko dan Undangan

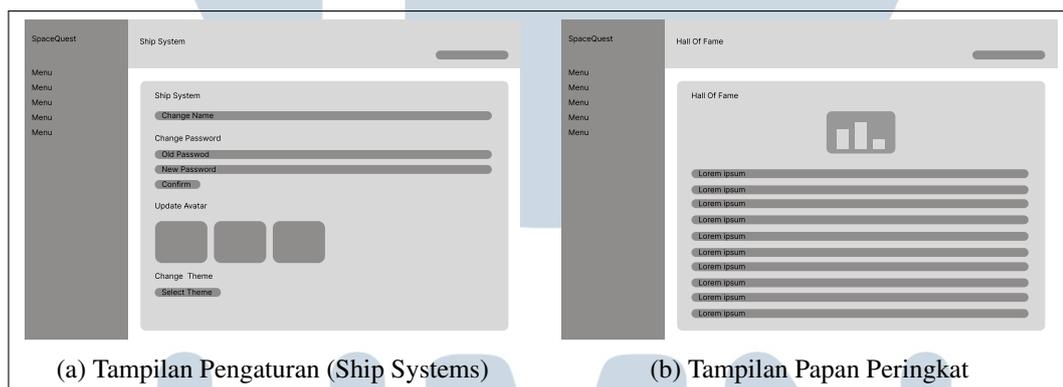
Gambar 3.17 menampilkan desain antarmuka untuk Halaman Toko dan Halaman Invitation. Pada Gambar 3.17a, Tampilan Toko atau "Star Market" dirancang sebagai etalase virtual yang menarik. Antarmukanya menampilkan galeri item kosmetik yang dapat dibeli, seperti Tema dan Bingkai Avatar, lengkap dengan gambar dan harganya dalam *Cosmic Points* (CP). Saldo CP yang dimiliki pengguna juga ditampilkan secara jelas untuk memberikan informasi daya beli pengguna. Ketika sebuah item dipilih, detail lebih lanjut dan tombol "Beli" akan muncul, memfasilitasi proses transaksi yang sederhana dan memuaskan.

Pada Gambar 3.17b, ditampilkan Tampilan Invitation Proyek. Halaman ini berfungsi sebagai pusat notifikasi sosial, di mana semua invitation untuk

bergabung ke dalam "Crew Expeditions" dikumpulkan. Desainnya sederhana dan fungsional, menyajikan daftar invitation dalam format kartu atau baris. Setiap entri menampilkan informasi penting seperti nama proyek dan siapa yang mengundang, disertai dengan tombol "Terima" dan "Tolak" yang jelas, memungkinkan pengguna untuk merespons invitation dengan cepat dan efisien.

## F Halaman Pengaturan dan Gamifikasi

Untuk melengkapi pengalaman pengguna, website "SpaceQuest" menyediakan halaman-halaman fungsional untuk personalisasi akun dan visualisasi pencapaian. Halaman Pengaturan memberikan kontrol penuh kepada pengguna atas data pribadi pengguna, sementara Halaman Papan Peringkat dan Pencapaian berfungsi untuk memvalidasi dan memamerkan progres pengguna.

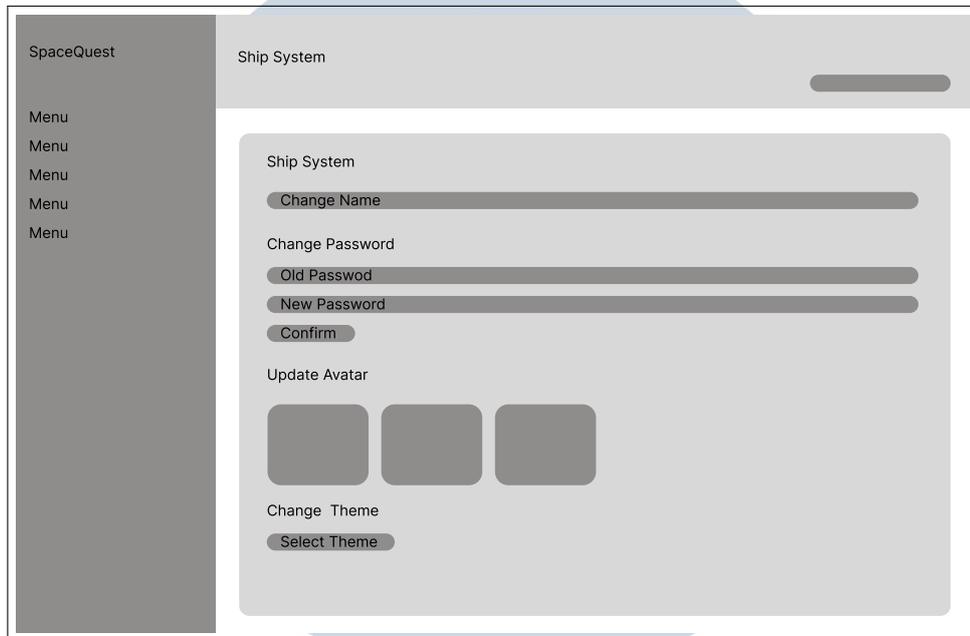


Gambar 3.18. Desain Antarmuka Halaman Pengaturan dan Papan Peringkat

Gambar 3.18 menampilkan desain antarmuka untuk Halaman Pengaturan dan Papan Peringkat. Pada Gambar 3.18a, Tampilan Pengaturan atau "Ship Systems" dirancang dengan tata letak yang terstruktur, biasanya menggunakan tab atau bagian terpisah untuk setiap kategori. Desain ini menyediakan form yang jelas untuk mengubah profil pengguna (seperti nama dan avatar), form untuk memperbarui kata sandi demi keamanan, serta galeri untuk memilih dan menerapkan item kustomisasi yang telah dimiliki. Alur yang terorganisir ini memastikan pengguna dapat dengan mudah menemukan dan mengelola pengaturan akun pengguna.

Pada Gambar 3.18b, ditampilkan desain untuk Halaman Papan Peringkat atau "Hall of Fame". Antarmuka ini dirancang untuk mendorong kompetisi yang sehat dengan menampilkan daftar peringkat pengguna berdasarkan total XP

yang pengguna kumpulkan. Desainnya menyoroti peringkat pengguna saat ini di antara pengguna lainnya, memberikan umpan balik langsung mengenai posisi sosial pengguna dalam ekosistem *website*.



Gambar 3.19. Desain Antarmuka Halaman Pencapaian (Medal Bay)

Terakhir, Gambar 3.19 menampilkan Halaman Pencapaian atau "Medal Bay". Halaman ini berfungsi sebagai lemari piala virtual. Desainnya menggunakan format galeri visual untuk menampilkan semua lencana (*badges*) yang telah berhasil dibuka oleh pengguna. Setiap lencana merepresentasikan sebuah *milestone* atau prestasi tertentu. Tampilan visual yang menarik ini memberikan rasa kepuasan dan berfungsi sebagai catatan abadi atas pencapaian pengguna selama menggunakan "SpaceQuest".

### 3.2.5 Perancangan Aset

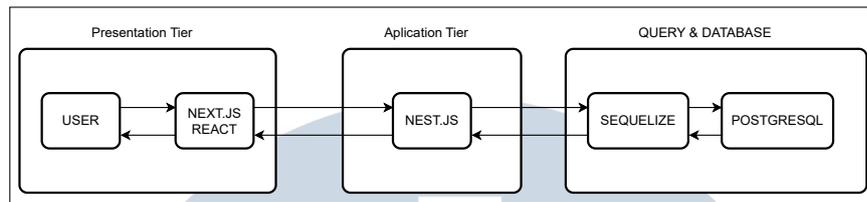
Daftar aset yang digunakan dalam proses perancangan website Space Quest terdapat pada tabel 3.15.

Tabel 3.15. Daftar aset Space Quest

Gambar	Keterangan	Sumber Gambar
	Logo website	www.fontawesome.com
	Latar belakang website	www.motionbgs.com
	Button navigasi untuk berpindah halaman	www.fontawesome.com
	List avatar yang terdapat pada halaman shop	www.elements.envato.com
	List badge dan thropy untuk achievement dan reward pada halaman Medal Bay	www.fontawesome.com

### 3.3 Pembangunan Website

Pembangunan website "SpaceQuest" menggunakan pendekatan arsitektur 3-Tier (3-Tier Architecture), yang memisahkan website menjadi tiga lapisan komputasi logis dan fisik yang berbeda: Presentation Tier, Application Tier, dan Data Tier. Pendekatan ini dipilih karena memberikan skalabilitas, fleksibilitas, dan kemudahan dalam pemeliharaan.



Gambar 3.20. Arsitektur Website SpaceQuest

Gambar 3.20 menampilkan arsitektur yang digunakan dalam pengembangan website "SpaceQuest". Penjelasan untuk setiap lapisan adalah sebagai berikut:

1. Presentation Tier (Frontend), merupakan lapisan antarmuka pengguna (*User Interface*) yang berinteraksi langsung dengan pengguna. Lapisan ini dibangun menggunakan Next.js, sebuah *framework* React yang mendukung *server-side rendering* dan pembuatan situs statis. Bahasa yang digunakan adalah TypeScript untuk memastikan *type safety* dan skalabilitas kode. Untuk *styling*, digunakan Tailwind CSS yang memungkinkan pembuatan desain kustom dengan cepat. Lapisan ini bertanggung jawab untuk menampilkan data yang diterima dari *Application Tier* dan mengirimkan input dari pengguna kembali ke *server*.
2. Application Tier (Backend), adalah lapisan logika bisnis yang berfungsi sebagai jembatan antara *Presentation Tier* dan *Data Tier*. Lapisan ini dibangun menggunakan *framework* NestJS dengan bahasa pemrograman TypeScript. NestJS dipilih karena arsitekturnya yang modular dan terstruktur, yang sangat cocok untuk membangun website sisi *server* yang efisien dan andal. Lapisan ini menangani semua logika bisnis, termasuk autentikasi pengguna, validasi data, pemrosesan gamifikasi, dan operasi CRUD (*Create, Read, Update, Delete*) melalui API RESTful.
3. Data Tier (Database), adalah lapisan tempat semua data website disimpan dan dikelola. "SpaceQuest" menggunakan sistem manajemen database relasional PostgreSQL. Interaksi antara *Application Tier* dan *Data Tier* dikelola oleh Sequelize, sebuah *Object-Relational Mapper* (ORM) berbasis Promise untuk Node.js. Sequelize mempermudah proses interaksi dengan database dengan memetakan model data dalam kode ke tabel dalam database PostgreSQL, sehingga abstraksi dari kueri SQL mentah dapat tercapai.

### 3.4 Uji Coba dan Evaluasi

Evaluasi dilakukan dengan pengujian kepada responden dari kalangan pekerja (magang atau karyawan atau wiraswasta ataupun freelancer), menggunakan model HMSAM, di mana setiap pertanyaan dinilai menggunakan skala Likert. Pertanyaan-pertanyaan dari model HMSAM telah diterjemahkan dan disesuaikan dengan konteks penggunaan website Space Quest. Rincian pertanyaan yang diajukan kepada responden dapat dilihat pada Tabel 3.16. Tanda \* pada item menandakan skala dengan nilai terbalik.

Tabel 3.16. Daftar pertanyaan metode evaluasi HMSAM

Construct	Items
Perceived ease of use	<p><b>PEOU1.</b> Interaksi saya pada website Space Quest terasa jelas dan dapat dimengerti.</p> <p><b>PEOU2.</b> Saya tidak butuh banyak usaha untuk memahami cara menggunakan website Space Quest.</p> <p><b>PEOU3.</b> Saya merasa website Space Quest bebas dari masalah.</p> <p><b>PEOU4.</b> Saya dapat dengan mudah untuk melakukan apa yang saya inginkan saat menggunakan website Space Quest.</p> <p><b>PEOU5.</b> Saya merasa mudah untuk menjadi terampil dalam menggunakan website Space Quest.</p> <p><b>PEOU6.</b> Saya merasa sederhana untuk melakukan apa yang saya inginkan dengan website Space Quest.</p> <p><b>PEOU7.</b> Saya merasa mudah untuk menjadi ahli dalam menggunakan website Space Quest.</p> <p><b>PEOU8.</b> Saya merasa website Space Quest mudah untuk digunakan.</p>

Lanjut ke halaman berikutnya

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA

Tabel 3.16 Daftar pertanyaan metode evaluasi HMSAM (lanjutan)

Construct	Items
Perceived usefulness	<p><b>PU1.</b> Website Space Quest mengurangi stres saya dalam mengelola tugas.</p> <p><b>PU2.</b> Website Space Quest membantu saya mengorganisir aktivitas harian dengan lebih baik.</p> <p><b>PU3.</b> Website Space Quest memberikan cara yang berguna untuk melacak kemajuan saya.</p> <p><b>PU4.</b> Website Space Quest membantu saya mengelola tugas dengan lebih efisien.</p> <p><b>PU5.</b> Website Space Quest membantu saya merasa lebih produktif dan terorganisir.</p>
Curiosity	<p><b>CUR1.</b> Menggunakan website Space Quest membangkitkan rasa penasaran saya tentang manajemen tugas.</p> <p><b>CUR2.</b> Website Space Quest membuat saya penasaran untuk mengeksplorasi lebih banyak fitur.</p> <p><b>CUR3.</b> Website Space Quest membangkitkan imajinasi saya tentang mengorganisir tugas secara kreatif.</p>
Joy	<p><b>JOY1.</b> Saya merasa menggunakan website Space Quest menyenangkan.</p> <p><b>JOY2.</b> Saya merasa senang mengelola tugas dengan website Space Quest.</p> <p><b>*JOY3.</b> Menggunakan website Space Quest membosankan.</p> <p><b>*JOY4.</b> Website Space Quest benar-benar mengganggu saya.</p> <p><b>JOY5.</b> Pengalaman menggunakan website Space Quest menyenangkan.</p> <p><b>*JOY6.</b> Website Space Quest membuat saya tidak puas.</p>
Lanjut ke halaman berikutnya	

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA

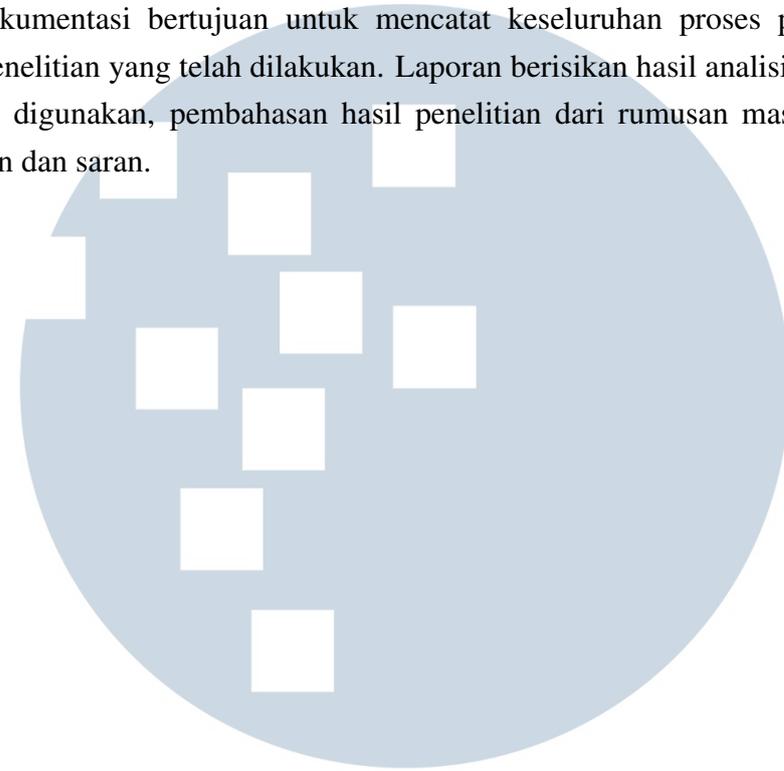
Tabel 3.16 Daftar pertanyaan metode evaluasi HMSAM (lanjutan)

Construct	Items
Control	<p><b>CTL1.</b> Saya memiliki banyak kontrol saat menggunakan website Space Quest.</p> <p><b>CTL2.</b> Saya dapat memilih dengan bebas apa yang ingin saya lihat atau lakukan di website Space Quest.</p> <p><b>*CTL3.</b> Saya memiliki sedikit kontrol atas apa yang dapat saya lakukan di website Space Quest.</p> <p><b>CTL4.</b> Saya merasa memiliki kendali saat menggunakan website Space Quest.</p> <p><b>*CTL5.</b> Saya tidak memiliki kontrol atas interaksi saya dengan website Space Quest.</p> <p><b>CTL6.</b> Saya diizinkan untuk mengontrol interaksi saya dengan website Space Quest.</p>
Behavioral intention to use	<p><b>BIU1.</b> Saya berencana untuk menggunakan website Space Quest di masa depan.</p> <p><b>BIU2.</b> Saya berniat untuk terus menggunakan website Space Quest di masa depan.</p> <p><b>BIU3.</b> Saya mengharapkan penggunaan website Space Quest saya akan berlanjut di masa depan.</p>
Focused immersion	<p><b>FI1.</b> Saya dapat mengabaikan sebagian besar gangguan lain saat menggunakan website Space Quest.</p> <p><b>FI2.</b> Saya larut dalam mengelola tugas dengan website Space Quest.</p> <p><b>FI3.</b> Saya tenggelam dalam sistem manajemen tugas Space Quest.</p> <p><b>*FI4.</b> Saya sangat mudah terganggu oleh hal lain saat menggunakan website Space Quest.</p> <p><b>FI5.</b> Perhatian saya tidak mudah teralihkan saat menggunakan website Space Quest.</p>

Keterangan: PEOU = *Perceived Ease-of-Use*, PU = *Perceived Usefulness*, CUR = *Curiosity*, JOY = *Joy*, CTL = *Control*, BIU = *Behavioral Intention to Use*, FI = *Focused Immersion*.

### 3.5 Dokumentasi

Dokumentasi bertujuan untuk mencatat keseluruhan proses pada setiap tahapan penelitian yang telah dilakukan. Laporan berisikan hasil analisis, landasan teori yang digunakan, pembahasan hasil penelitian dari rumusan masalah, serta kesimpulan dan saran.



# UMMN

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA