

BAB 3

PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi

Selama melaksanakan program magang di PT Braincode Digital Teknologi, posisi yang diemban adalah sebagai *Software Engineer*, dengan fokus pada pengembangan antarmuka pengguna di bawah divisi *Frontend Development*. Dalam struktur organisasi proyek, peserta magang berada di bawah supervisi langsung **Pak Ramdani**, yang menjabat sebagai *Technical Lead* sekaligus menjadi pembimbing teknis selama program berlangsung.

Posisi sebagai *Software Engineer* ini diisi oleh empat orang peserta magang dari latar belakang institusi pendidikan yang beragam, antara lain:

- **Reinhard Javera Maheswara**
Universitas Multimedia Nusantara – Informatika (Angkatan 2022)
- **Muhammad Tristan Ajibrilyan Nandipinto**
Universitas Multimedia Nusantara – Informatika (Angkatan 2022)
- **Muhammad Ukasah Hayata, S.Si.**
Universitas Pendidikan Indonesia – Sistem Informasi (Angkatan 2020)
- **Farhan Syarif Hidayatulloh**
SMK Wikrama Bogor – Rekayasa Perangkat Lunak (Angkatan 2023)

Dalam kegiatan harian, seluruh peserta magang berkoordinasi secara langsung dengan *Technical Lead* melalui diskusi rutin, pelaporan progres harian, serta sesi peninjauan kode (*code review*). Koordinasi juga dilakukan dengan tim pengembang lainnya guna memastikan keselarasan tugas dan pencapaian target proyek yang telah ditetapkan.

3.2 Tugas yang Dilakukan

Tabel 3.1. Pekerjaan yang dilakukan tiap minggu selama pelaksanaan magang

Minggu Ke -	Pekerjaan yang dilakukan
1	Telah dilakukan serangkaian pengembangan dan pengujian fitur pada sistem, meliputi implementasi dan demonstrasi fitur <i>zoom</i> pada tampilan <i>Gantt Chart</i> , serta pengujian dan penyelesaian fungsi <i>api.on()</i> untuk memastikan fungsionalitasnya berjalan dengan baik. Selain itu, dilakukan juga pengembangan antarmuka pengguna, termasuk pembuatan <i>user interface</i> untuk fitur <i>filter</i> berdasarkan wilayah, penambahan <i>search bar</i> , serta penyempurnaan tampilannya agar lebih optimal.
2	Penyempurnaan fitur <i>filter</i> wilayah, termasuk pembatasan pemilihan level wilayah, penyesuaian struktur data, dan penambahan tombol cepat untuk wilayah tertentu. <i>UI</i> ditingkatkan dengan efek <i>hover</i> , perbaikan tampilan <i>input</i> , serta reposisi tombol agar lebih mudah diakses. Selain itu, sistem berhasil diintegrasikan dengan sumber data eksternal setelah perbaikan pada <i>API</i> .
3	Menambahkan fitur integrasi <i>preset</i> wilayah (<i>save/load</i>) yang terhubung ke <i>backend</i> , tombol <i>Expand/Collapse All</i> untuk kemudahan navigasi, serta alternatif <i>API</i> untuk <i>Source B</i> . Selain itu, halaman <i>login</i> telah dirancang dan dihubungkan ke <i>backend</i> sesuai desain <i>Figma</i> untuk versi <i>desktop</i> dan <i>mobile</i> .

UNIVERSITAS
MULTIMEDIA
NUSANTARA

Tabel 3.1. Pekerjaan yang dilakukan tiap minggu selama pelaksanaan magang (lanjutan)

Minggu Ke -	Pekerjaan yang dilakukan
4	<p>Mengembangkan tampilan <i>Employee List</i> untuk <i>admin</i> dalam bentuk tabel, lengkap dengan fitur <i>dropdown filter</i>, pencarian, dan <i>checkbox</i> seleksi ganda. Data karyawan terhubung ke <i>API</i> eksternal sehingga dapat dimuat otomatis setelah <i>login</i>. Navigasi antar halaman diatur agar <i>admin</i> dapat mengakses <i>Employee Detail</i> berdasarkan <i>NIK</i> karyawan. Perbaikan juga dilakukan pada komponen <i>UserProfileHeader</i> dan penyesuaian terhadap perubahan versi <i>AG Grid</i>. Selain itu, telah dibuat tampilan awal untuk halaman <i>Course</i>, termasuk <i>Course Field</i>, <i>Course Grade</i>, dan <i>Course Selection</i>, beserta komponen <i>Course Field Card</i> yang interaktif.</p>
5	<p>Membuat fungsi <i>login</i> dengan verifikasi akun, penyimpanan <i>token</i>, dan pengalihan pengguna ke halaman sesuai peran. Sistem juga telah terhubung ke <i>server</i> untuk menampilkan daftar karyawan. Fitur <i>Course Enrollment</i> dikembangkan dengan tampilan detail kursus dan tombol <i>enroll</i> yang aktif hanya satu kali. Selain itu, fitur <i>Module</i> dan <i>Submodule</i> telah diimplementasikan, termasuk navigasi antar modul serta pemutaran konten video dari <i>YouTube</i> melalui komponen khusus seperti <i>CourseSubModule</i> dan <i>ModuleSidebar</i>.</p>
6	<p>Pengembangan halaman <i>Course SubModule</i> mencakup penambahan komponen <i>FreeText</i> untuk konten <i>HTML</i> dan <i>PDFViewer</i> untuk dokumen <i>PDF</i> dengan <i>dynamic import</i>. Tampilan <i>CourseCard</i> diperbarui sesuai desain <i>Figma</i>. Halaman <i>Add Employee</i> mulai dibangun, lengkap dengan <i>step indicator</i>, <i>form</i> data karyawan, unggah foto interaktif, serta isian <i>contract</i>, <i>department</i>, dan <i>position</i> yang saling menyesuaikan.</p>

Tabel 3.1. Pekerjaan yang dilakukan tiap minggu selama pelaksanaan magang (lanjutan)

Minggu Ke -	Pekerjaan yang dilakukan
7	<p>Mengembangkan fitur <i>input</i> data untuk penambahan pengguna baru, mencakup informasi pribadi, posisi, alamat, dan detail pekerjaan. Fitur <i>simpan data</i> telah terhubung ke <i>server</i> sehingga data langsung masuk ke <i>database</i>. Proses <i>kompresi otomatis</i> ditambahkan untuk foto karyawan setelah diedit. <i>Dropdown</i> seperti <i>Department</i>, <i>Position</i>, <i>Role</i>, dan <i>Employee Type</i> kini terhubung ke <i>server</i> dan mampu menyimpan data lengkap berupa <i>ID</i> dan <i>label</i> untuk memastikan integritas data saat disimpan.</p>
8	<p><i>Dropdown Work Type</i> dan <i>Job Position</i> telah terintegrasi dengan <i>API</i> dan dilengkapi opsi “<i>All</i>”. Halaman <i>Employee List</i> ditingkatkan dengan tombol <i>Reset Filter</i>, aksi <i>Edit/Delete</i>, serta penyesuaian visual dan responsivitas. Fungsi <i>Edit/Delete</i> kini terhubung ke <i>API</i> dan disertai konfirmasi. Halaman <i>Add Employee</i> dan <i>Resume</i> ditingkatkan dengan format tanggal baru, validasi <i>form</i>, dan pengecekan ukuran foto. Proyek di-<i>build</i> dan di-<i>deploy</i> ke <i>server</i>.</p>
9	<p>Mengimplementasikan validasi keunikan data pada halaman <i>Add Employee</i>. Menambahkan juga <i>error handling</i> dan <i>loading state</i> pada pemanggilan <i>API</i>. Format tanggal di halaman <i>Employee Resume</i> disesuaikan menggunakan nama bulan berbahasa Indonesia. Fitur tambahan meliputi konfirmasi saat <i>form</i> belum disimpan, <i>Edit Avatar</i> dengan <i>modal</i> dan <i>editor gambar</i>, serta pembaruan desain sesuai <i>Figma</i>.</p>

UNIVERSITAS
MULTIMEDIA
NUSANTARA

Tabel 3.1. Pekerjaan yang dilakukan tiap minggu selama pelaksanaan magang (lanjutan)

Minggu Ke -	Pekerjaan yang dilakukan
10	Menerapkan validasi keunikan data dan validasi <i>real-time</i> pada <i>form input</i> , lengkap dengan <i>error handling</i> . Halaman <i>Employee Resume</i> diperbarui sesuai desain <i>Figma</i> , termasuk format tanggal Indonesia dan fitur <i>Edit Avatar</i> . Beberapa logika otomatis ditambahkan, seperti pengecualian <i>Contract Data</i> untuk <i>PKWTT</i> dan pengisian <i>default grade/level</i> untuk posisi tertentu. Validasi ukuran foto di halaman <i>Add User</i> juga ditingkatkan. Selain itu, halaman <i>Forgot Password</i> dan <i>Reset Password</i> dikembangkan lengkap dengan validasi kode dan masa berlaku.
11	Integrasi <i>API Forgot/Reset Password</i> dengan validasi kekuatan sandi, <i>Autocomplete</i> untuk <i>Tempat Lahir</i> via <i>GraphQL API</i> kota, reposisi <i>popup</i> sukses di <i>Employee Resume</i> , unggah <i>avatar</i> via metode <i>PATCH</i> terpisah, dan pengembangan halaman <i>Edit Course</i> (termasuk detail <i>course</i> dan <i>modul dummy</i> dari <i>API</i>).
12	Melakukan penyeragaman tampilan <i>Modal Upload PDF</i> agar konsisten dengan <i>Modal Upload Video</i> . Struktur data <i>submodul</i> dipisahkan per <i>modul</i> untuk mencegah pencampuran data. Beberapa perbaikan dilakukan, termasuk <i>error</i> saat <i>Add Question</i> di <i>Quiz</i> , <i>dropdown Grade</i> yang kini mengambil data dari <i>API</i> , dan <i>field Title</i> pada <i>AG Grid</i> yang telah disesuaikan. Fitur <i>Pagination</i> ditambahkan pada halaman <i>Course List</i> menggunakan <i>HeroUI</i> . Selain itu, halaman <i>Course Enrollment</i> telah disinkronkan dengan berbagai <i>API</i> , termasuk daftar <i>modul</i> dan tombol <i>Enroll</i> , yang kini berubah menjadi <i>Open Course</i> jika pengguna sudah terdaftar, serta diarahkan otomatis ke halaman <i>modul</i> .

Tabel 3.1. Pekerjaan yang dilakukan tiap minggu selama pelaksanaan magang (lanjutan)

Minggu Ke -	Pekerjaan yang dilakukan
13	Halaman <i>SubModule</i> kini terintegrasi penuh dengan <i>API</i> menggunakan <i>Axios</i> untuk konten dan daftar, serta dapat menampilkan video atau <i>PDF</i> sebagai <i>URL Blob</i> melalui <i>Attachment Service</i> . Komponen <i>PDFViewer</i> telah diperbaiki dengan penambahan fitur input halaman, dan <i>Quill JS</i> digunakan untuk menggantikan <i>textarea</i> pada bagian <i>Deskripsi Kursus</i> .
14	Tampilan <i>Course Description</i> pada halaman <i>Course Enrollment</i> telah diperbarui menggunakan <i>Quill JS</i> dalam mode <i>read-only</i> . Logika pengisian konten juga diperbaiki dengan menghapus awalan " <i>RichText</i> " sebelum isi konten. Di sisi lain, tampilan <i>SubModule List</i> dan <i>Quiz</i> pada halaman <i>Course SubModule</i> diperbaiki agar menampilkan label yang sesuai berdasarkan data <i>API</i> . Sementara itu, halaman <i>Course Edit</i> dikembangkan lebih lanjut dengan menambahkan <i>field Course Duration</i> dan <i>Course Image</i> , serta menggunakan komponen baru <i>RichTextEditor</i> untuk mengisi <i>Course Description</i> . Komponen ini memastikan data tersimpan dan diperbarui saat <i>form</i> di-submit.
15	Tampilan <i>Quiz Instruction</i> dan halaman <i>Quiz View</i> berhasil dikembangkan sesuai desain <i>Figma</i> , dan diintegrasikan dengan <i>API</i> untuk menampilkan konten kuis saat pengguna mengakses dari <i>Module Sidebar</i> . Selain itu, <i>RichTextEditor</i> untuk <i>Course Description</i> diperbarui dengan fitur <i>color palette</i> agar menyerupai <i>Microsoft Word</i> . Di halaman <i>Course Edit</i> , <i>UI</i> dan layanan unggah file (gambar, video, dan <i>PDF</i>) juga disempurnakan, termasuk sinkronisasi dengan <i>API</i> dan penggunaan <i>AG Grid</i> untuk menampilkan daftar file yang tersedia.

Tabel 3.1. Pekerjaan yang dilakukan tiap minggu selama pelaksanaan magang (lanjutan)

Minggu Ke -	Pekerjaan yang dilakukan
16	<i>Quiz List</i> kini ditampilkan dalam satu halaman dengan <i>SubModule</i> lain. Navigasi ke halaman kuis disesuaikan melalui tombol ‘ <i>Start Quiz</i> ’, dan tombol ‘ <i>Next</i> ’ pada halaman <i>SubModule</i> telah diperbaiki agar dapat mengarahkan ke <i>submodul</i> maupun kuis dengan lancar.
17	Membuat halaman <i>Dashboard</i> untuk <i>admin</i> dan <i>employee</i> dengan <i>layout</i> responsif untuk tampilan <i>mobile</i> dan <i>desktop</i> . Kedua halaman menggunakan <i>path</i> yang sama, namun menampilkan konten yang berbeda sesuai peran pengguna. Komponen grafik dibangun menggunakan <i>AmCharts</i> , dan pengerjaan halaman dilanjutkan hingga seluruh tampilan dan fungsionalitas <i>dashboard</i> selesai.
18	Pengembangan halaman <i>Dashboard - Employee</i> dilanjutkan dan telah terintegrasi dengan berbagai API. Selain itu, halaman <i>Quiz</i> kini terhubung dengan API saat tombol <i>submit</i> diklik. Komponen <i>Course Enrolled</i> juga diintegrasikan ke <i>Dashboard - Admin</i> dan <i>Employee</i> menggunakan endpoint yang sama.
19	Detailing halaman <i>Dashboard - Employee</i> dan <i>Dashboard - Admin</i> untuk tampilan <i>mobile</i> dan <i>desktop</i> telah dilakukan agar semakin sesuai dengan desain di <i>Figma</i> . Pekerjaan dilanjutkan secara bertahap hingga seluruh elemen visual pada kedua halaman menyerupai desain yang direncanakan.

3.3 Uraian Pelaksanaan Magang

Pelaksanaan magang dilakukan dengan mengerjakan tugas-tugas yang diberikan oleh supervisor selama kurang lebih 5 bulan 1 minggu. Tugas utama yang dikerjakan adalah membangun website *HIVES (Holistic Intelligent Virtual Evaluation System)*, yaitu sebuah *Learning Management System (LMS)* internal milik PT Braincode Digital Teknologi.

Proyek *HIVES* dirancang sebagai platform evaluasi holistik untuk memantau perkembangan kompetensi karyawan secara berkelanjutan. Sistem ini memungkinkan perusahaan untuk mengukur tingkat kelayakan, eligibilitas, serta potensi promosi karyawan berdasarkan capaian keterampilan. Dalam platform ini,

setiap karyawan akan diberikan penilaian dalam bentuk tingkatan atau *grade* yang diklasifikasikan menjadi tiga level, yaitu A, B, dan C.

Penentuan *grade* didasarkan pada penyelesaian kursus-kursus pembelajaran yang telah disesuaikan dengan posisi atau jabatan masing-masing. Semakin tinggi *grade* yang dicapai, semakin besar peluang untuk memperoleh kenaikan gaji maupun promosi jabatan. Dengan adanya sistem ini, perusahaan dapat melakukan penilaian berbasis data terhadap performa dan perkembangan sumber daya manusia secara lebih objektif.

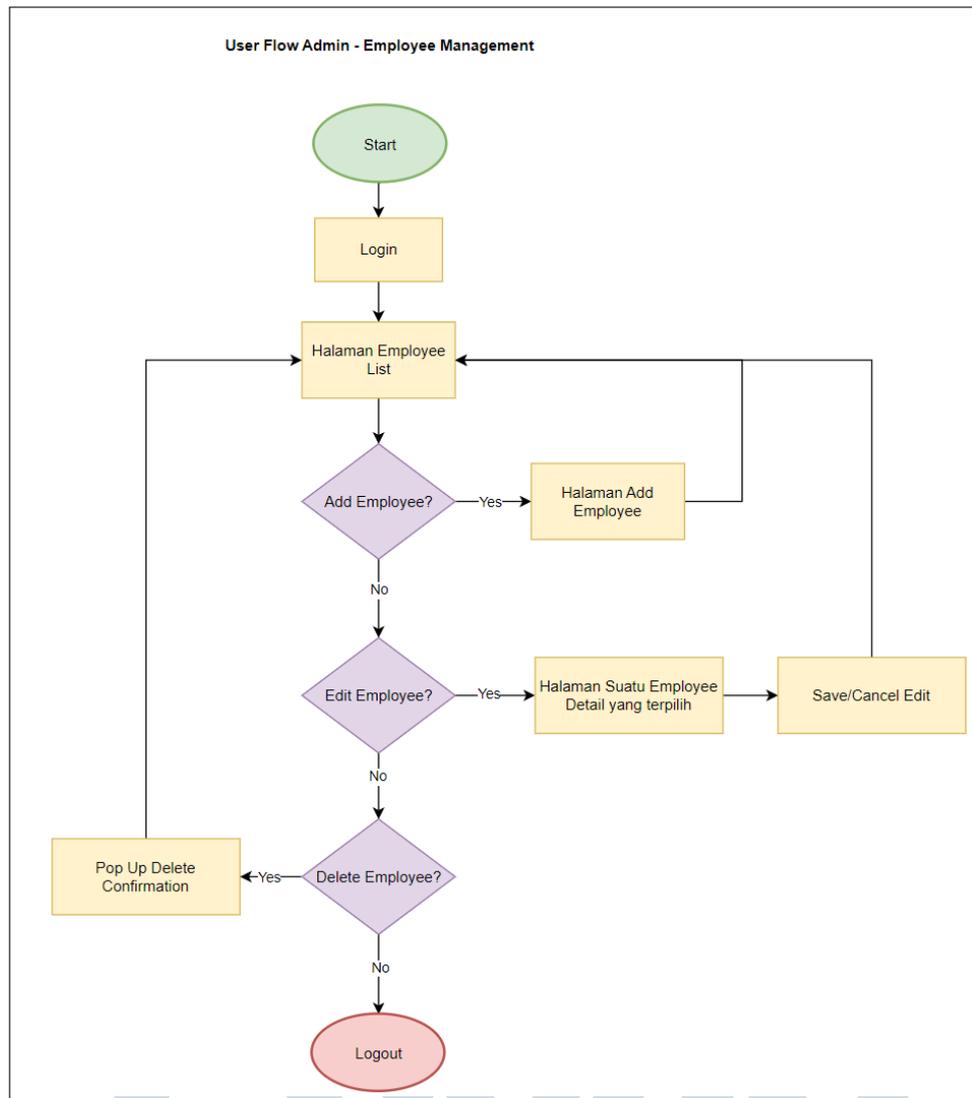
3.3.1 Perancangan Sistem

Sebelum proses pengembangan dilakukan, langkah awal yang dilakukan adalah merancang alur sistem agar pengembangan antarmuka berjalan terstruktur dan sesuai kebutuhan pengguna. Perancangan ini mencakup pemetaan proses bisnis, alur navigasi pengguna, serta struktur halaman yang akan diterapkan pada *Learning Management System (LMS) HIVES*. Diagram-diagram berikut digunakan sebagai acuan utama dalam proses implementasi antarmuka dan memastikan setiap fitur memiliki keterkaitan logis dalam ekosistem aplikasi [4, 5].

A User Flow Admin - User Management

User flow admin - user management ini menggambarkan alur aktivitas *Admin* dalam mengelola data karyawan pada sistem *Learning Management System (LMS)*. Proses dimulai dari tahap login, di mana sistem akan melakukan verifikasi kredensial dan menampilkan halaman utama berupa daftar karyawan. Dari halaman ini, *Admin* memiliki tiga opsi utama: menambah, mengedit, atau menghapus data karyawan.

Jika *Admin* memilih untuk menambah karyawan, maka akan diarahkan ke halaman form input data karyawan baru. Untuk pengeditan, *Admin* dapat membuka detail karyawan dan melakukan perubahan sesuai kebutuhan. Sementara itu, ketika ingin menghapus data karyawan, sistem akan menampilkan pop-up konfirmasi terlebih dahulu. Semua alur akan kembali ke halaman daftar karyawan setelah aksi dilakukan, dan proses diakhiri dengan opsi logout untuk keluar dari sistem.



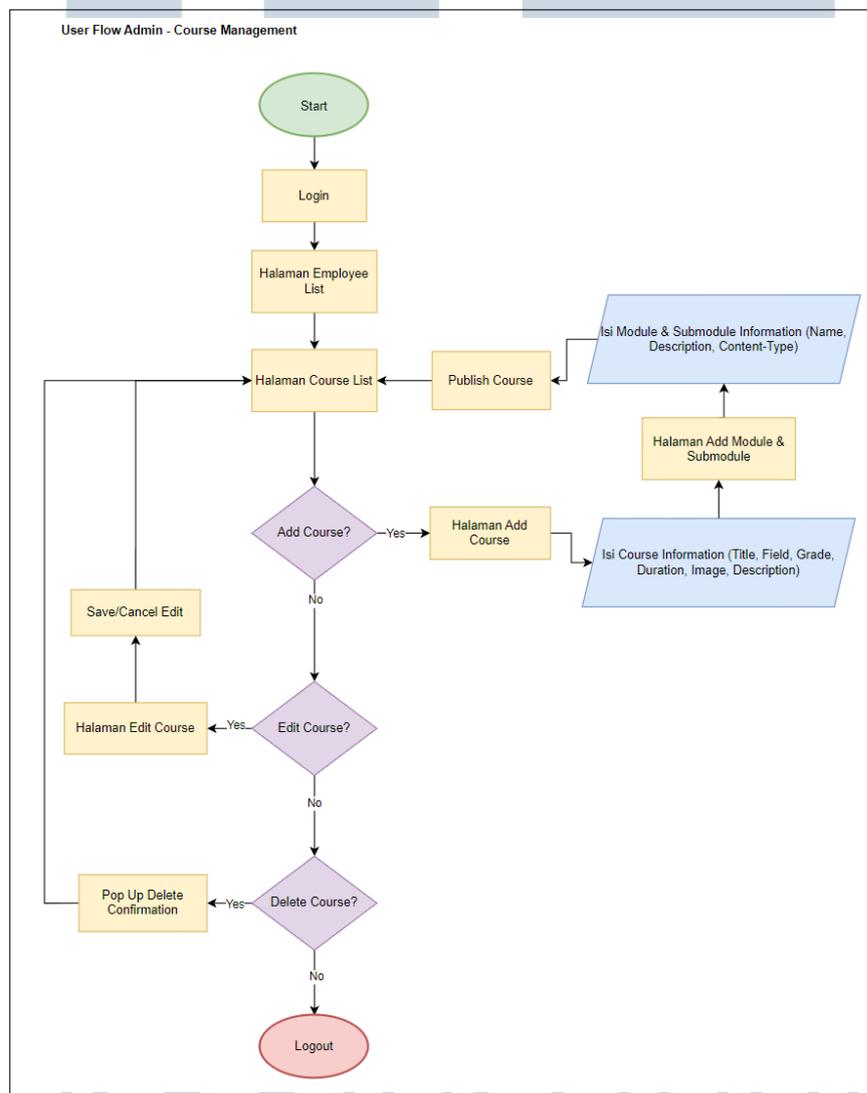
Gambar 3.1. User flow admin - user management

B User Flow Admin - Course Management

User flow admin - course management ini menunjukkan alur kerja admin dalam fitur *Course Management* pada sistem LMS. Proses diawali dengan autentikasi login, dilanjutkan dengan navigasi menuju halaman *Course List*. Di halaman ini, admin dapat melihat daftar kursus yang tersedia dan melakukan beberapa aksi, seperti menambahkan, mengedit, atau menghapus kursus. Untuk menambahkan kursus baru, admin akan diarahkan ke halaman *Add Course* dan mengisi informasi seperti judul, bidang, tingkat, durasi, gambar, serta deskripsi kursus. Setelah itu, admin dapat menambahkan struktur pembelajaran berupa modul

dan submodul dengan informasi seperti nama, deskripsi, dan tipe konten.

Setelah seluruh informasi kursus dan kontennya diisi, admin dapat mempublikasikan kursus agar tersedia bagi karyawan. Fitur pengeditan memungkinkan admin memperbarui data kursus yang sudah ada, sedangkan fitur hapus dilengkapi dengan *pop-up confirmation* untuk menghindari penghapusan data secara tidak sengaja. Seluruh alur kembali ke halaman *Course List* setelah setiap aksi diselesaikan, dan ditutup dengan proses *logout*. Flowchart ini mencerminkan bagaimana admin mengelola konten pembelajaran yang terstruktur dalam bentuk hierarki *course*, *module*, dan *SubModule* di dalam sistem LMS.



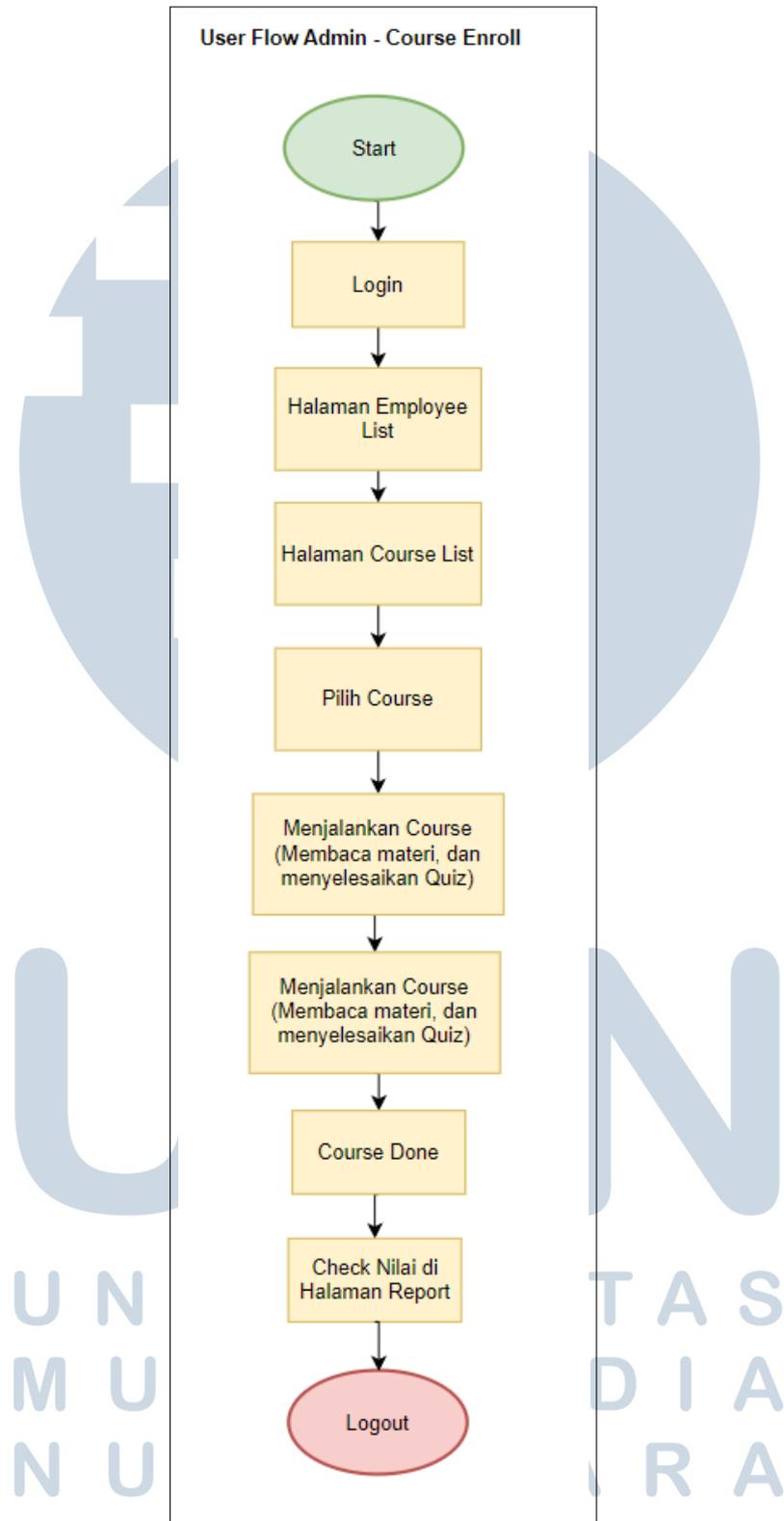
Gambar 3.2. User flow admin - course management

C User flow admin - course enroll

User flow admin - course enroll ini menunjukkan alur aktivitas admin saat menjalankan peran sebagai pembelajar dalam sistem Learning Management System (LMS). Alur dimulai dari proses login, kemudian diarahkan ke halaman daftar karyawan (*Employee List*) dan kursus (*Course List*). Setelah memilih kursus yang ingin diikuti, admin akan diarahkan ke halaman pembelajaran yang terdiri dari berbagai jenis konten seperti teks, video, PDF, dan kuis. Alur pembelajaran bisa terjadi secara bertahap hingga seluruh modul selesai.

Setelah seluruh materi dan evaluasi berhasil diselesaikan, sistem akan menandai kursus sebagai selesai (*Course Done*), dan admin dapat mengakses halaman *Report* untuk melihat nilai dan performa pembelajaran. Tahapan ini menunjukkan bahwa admin tidak hanya memiliki akses manajemen, tetapi juga dapat mengakses kursus seperti pengguna umum. Hal ini memungkinkan admin untuk mengevaluasi langsung kualitas konten dari sudut pandang pengguna akhir, sebelum akhirnya mengakhiri sesi dengan *logout*.





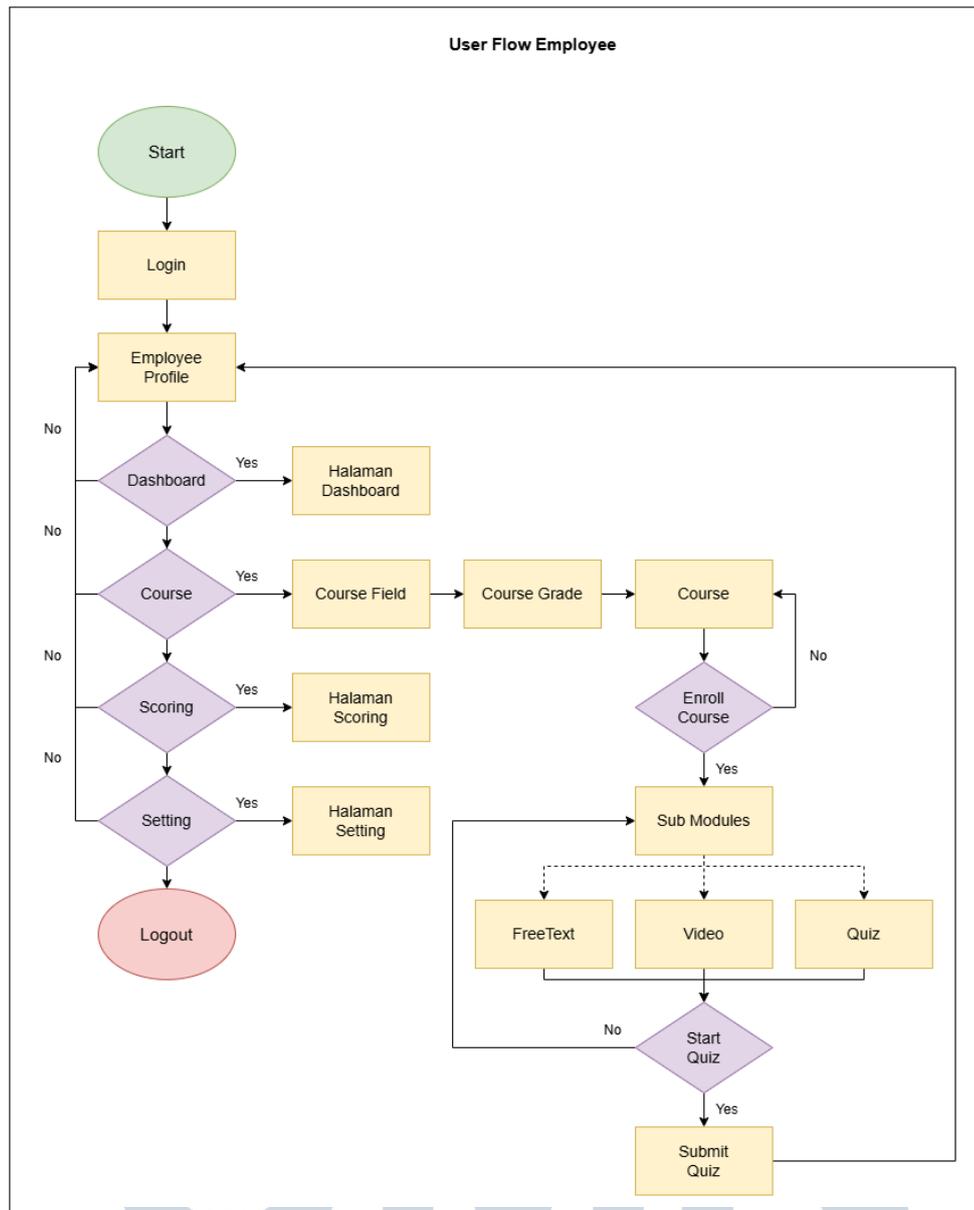
Gambar 3.3. User flow admin - course management

D User Flow Employee

User flow employee ini menunjukkan alur interaksi pengguna dengan peran *Employee* saat menggunakan sistem Learning Management System (LMS). Proses dimulai dari autentikasi login, di mana pengguna dengan role *Employee* akan diarahkan ke halaman *Employee Profile*. Dari halaman ini, karyawan dapat memilih beberapa jalur navigasi seperti *Dashboard*, *Course*, *Scoring*, dan *Setting*. Jika memilih *Dashboard*, pengguna akan melihat ringkasan aktivitas pembelajaran. Jika memilih *Course*, pengguna diarahkan ke halaman pemilihan kursus berdasarkan kategori dan tingkat kesulitan, lalu dapat melakukan *enroll* dan memulai pembelajaran melalui *SubModules* seperti materi teks, video, atau kuis.

Setelah menyelesaikan materi, pengguna dapat mengikuti *quiz*, mengirimkan hasilnya, dan kembali ke halaman profil. Selain itu, melalui halaman *Scoring*, karyawan dapat melihat nilai dari kursus yang telah mereka ikuti, serta memonitor progres pembelajaran. Menu *Setting* memungkinkan pengaturan akun secara mandiri. Akhir dari alur ini adalah proses *logout*, yang akan menghapus token autentikasi dan mengarahkan kembali ke halaman login. Flowchart ini menggambarkan peran *Employee* sebagai pembelajar dalam sistem LMS, dengan fokus pada pengembangan kompetensi melalui pembelajaran mandiri dan terstruktur.





Gambar 3.4. User flow employee

E Use Case Diagram HIVES

Use Case Diagram untuk sistem *Learning Management System (LMS)* yang dinamakan **HIVES** ini menggambarkan hubungan dan interaksi antara dua aktor utama, yaitu **Admin** dan **Employee**, dengan berbagai fungsionalitas yang tersedia di dalam sistem. Masing-masing aktor memiliki hak akses dan peran yang berbeda sesuai tanggung jawabnya dalam operasional sistem.

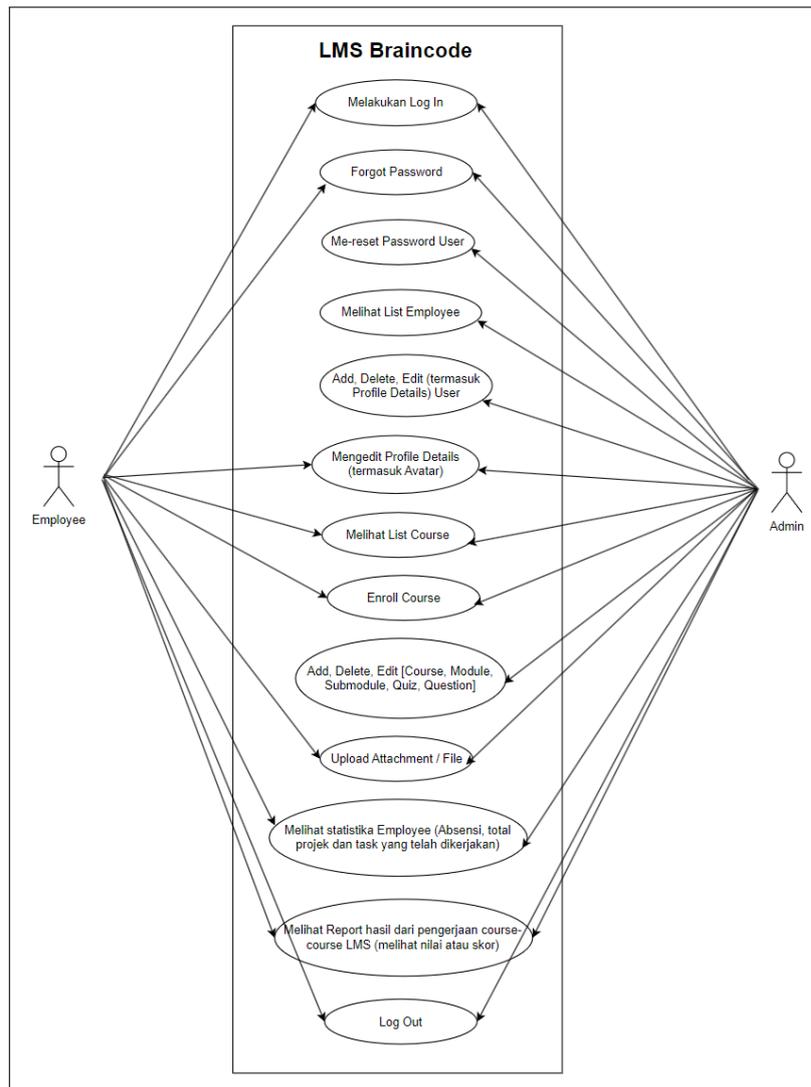
Baik *Employee* maupun *Admin* memiliki akses terhadap fitur umum

seperti melakukan *log in*, mengakses *forgot password*, mengedit *profile details* termasuk avatar, melihat daftar kursus, melakukan pendaftaran atau *enroll course*, mengakses hasil pembelajaran, serta melakukan *log out* dari sistem. Fungsi-fungsi ini merupakan inti dari kegiatan pembelajaran yang dilakukan oleh pengguna dalam sistem.

Sementara itu, *Admin* memiliki hak akses tambahan yang mencakup pengelolaan sistem, seperti *me-reset password user lain*, melihat daftar karyawan (*list employee*), serta melakukan proses *add, delete, edit* pada data pengguna termasuk detail profil. Selain itu, Admin juga bertanggung jawab terhadap manajemen konten pembelajaran, yang mencakup pembuatan, penghapusan, dan pengeditan *course, module, SubModule, quiz*, serta *question*. Admin juga memiliki kemampuan untuk mengunggah *attachment/file* sebagai materi pembelajaran dan melihat statistik karyawan seperti absensi, total proyek, dan tugas yang telah diselesaikan.

Secara keseluruhan, diagram ini memberikan gambaran menyeluruh mengenai kapabilitas sistem *Learning Management System (LMS)* ini. Diagram ini memperjelas struktur hak akses di mana Admin bertugas sebagai pengelola sistem, sementara Employee merupakan pengguna utama sistem pembelajaran. Fitur-fitur yang ditampilkan pada diagram ini juga telah dijelaskan dan diintegrasikan dalam *flowchart* sebelumnya, sehingga memberikan konteks yang lebih utuh terhadap keseluruhan desain sistem.





Gambar 3.5. Use case diagram LMS Braincode

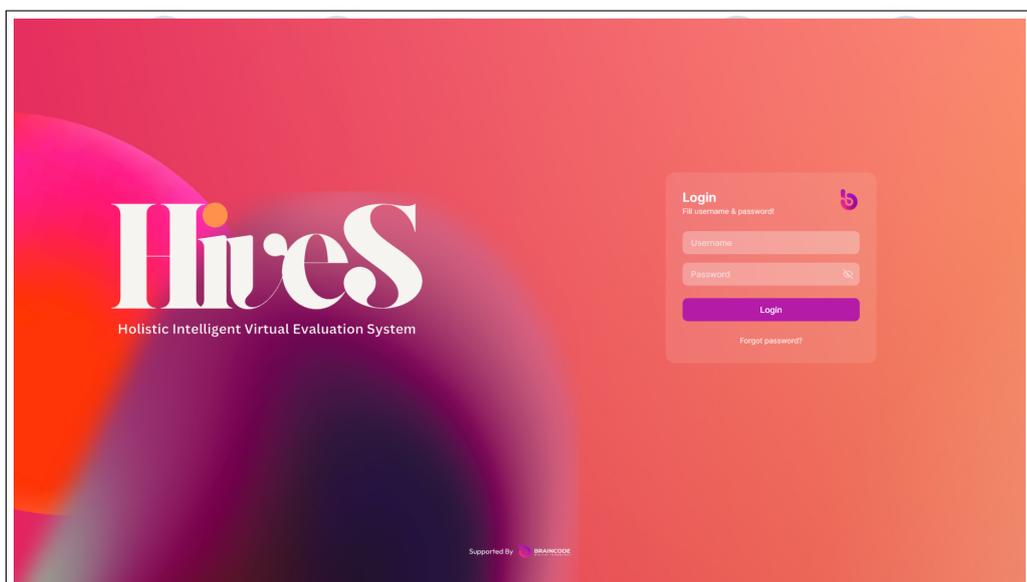
3.3.2 Antarmuka Sistem dan Hasil Pengembangan

Bagian ini menjelaskan antarmuka sistem yang telah berhasil dikembangkan selama masa proyek magang. Setiap antarmuka dirancang untuk mendukung kebutuhan operasional *admin* dan *employee*, mulai dari autentikasi pengguna, manajemen kursus, hingga pengelolaan data karyawan. Hasil pengembangan ditampilkan dalam bentuk halaman-halaman utama yang menunjukkan fitur, alur kerja, serta integrasi dengan sistem lain seperti layanan *upload*, *API* kursus, dan pengaturan submodul. Dokumentasi visual berupa tangkapan layar juga disertakan untuk menggambarkan tampilan dan fungsionalitas dari masing-masing bagian sistem.

A Halaman Login

Halaman *Login* pada sistem *LMS HIVES* berfungsi sebagai gerbang utama bagi pengguna untuk mengakses sistem, baik sebagai *Admin* maupun *Employee*. Tampilan halaman ini dirancang dengan kombinasi warna gradasi merah-oranye-ungu dan elemen visual berbentuk lingkaran untuk menciptakan kesan dinamis dan menarik secara visual, sesuai dengan prinsip desain antarmuka pengguna yang dapat meningkatkan perhatian dan kepuasan pengguna [6]. Komponen utama dalam halaman ini meliputi input field untuk *username* dan *password*, tombol *Login*, serta tautan *Forgot password?*. Selain itu, elemen identitas visual seperti logo “HiveS” dan logo *Braincode* ditampilkan secara mencolok untuk memperkuat branding sistem.

Secara fungsional, halaman ini terintegrasi dengan *service API* untuk proses autentikasi pengguna. Setelah data login diisi dan dikirim, sistem akan melakukan validasi serta memanggil `loginService()` ke backend. Jika berhasil, token autentikasi disimpan di `cookies` dan `localStorage`, lalu pengguna diarahkan sesuai dengan peran: *Admin* menuju halaman `/employee`, dan *Employee* menuju `/employee/profile`. Sistem juga memiliki penanganan kesalahan yang menampilkan pesan error ketika terjadi kegagalan autentikasi atau input tidak valid, serta dukungan pengelolaan sesi berbasis token untuk menjaga keamanan akses.

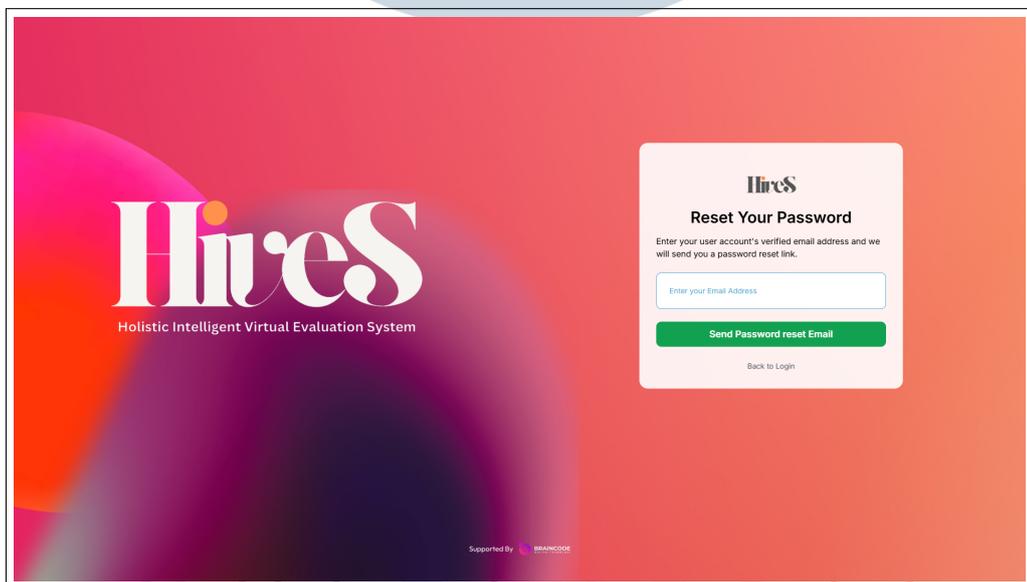


Gambar 3.6. Halaman login

B Halaman Forgot Password

Halaman *Forgot Password* pada LMS *HIVES* dirancang untuk membantu pengguna memulihkan akses akun secara mandiri. Tampilan visualnya konsisten dengan halaman login, menggunakan latar belakang gradasi merah-oranye-ungu dan elemen gelembung sebagai aksent untuk menjaga keseragaman antarmuka dan meningkatkan pengalaman pengguna [7]. Logo “HiveS” serta slogan “Holistic Intelligent Virtual Evaluation System” tampil di sisi kiri, sedangkan logo *Braincode* dengan teks “Supported By” berada di kanan bawah. Form utama terdiri dari input email, tombol *Send Password Reset Email* berwarna hijau, serta tautan *Back to Login* untuk kembali ke halaman login.

Secara fungsional, pengguna mengisi alamat email yang terverifikasi, lalu sistem akan memproses permintaan melalui `forgotPasswordService`. Jika berhasil, pengguna menerima notifikasi bahwa link reset telah dikirim ke email. Jika gagal, sistem akan menampilkan pesan kesalahan. Proses ini dilakukan secara aman melalui token khusus yang dikirim via email, dan menjadi bagian penting dari sistem manajemen akun tanpa perlu intervensi admin.

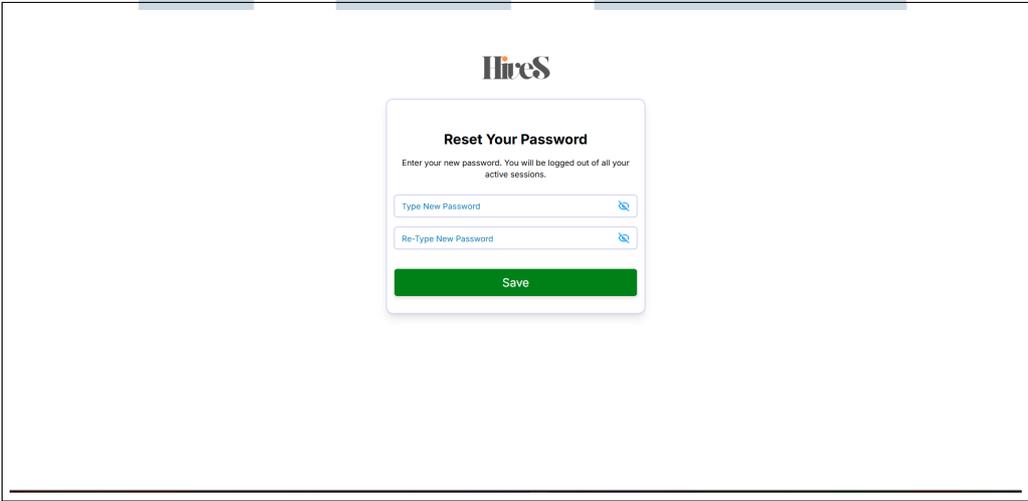


Gambar 3.7. Halaman Forgot Password

C Halaman Reset Password

Halaman *Reset Password* pada LMS *HIVES* merupakan tahap akhir dari proses pemulihan akun yang diakses melalui tautan yang dikirim ke email

pengguna. Desain halaman ini bersifat minimalis dengan latar belakang putih, menampilkan logo “HiveS” dan sebuah form dengan dua field input password beserta ikon toggle visibilitas. Pengguna diminta untuk memasukkan password baru dan konfirmasinya, lalu sistem akan memverifikasi kecocokan, kekuatan password, serta validitas token yang diambil dari URL. Jika berhasil, password baru dikirim melalui API ke server untuk disimpan, seluruh sesi pengguna sebelumnya diakhiri, dan pengguna diarahkan kembali ke halaman login. Fitur ini menjamin keamanan proses reset password dengan mekanisme token yang hanya berlaku satu kali dan memiliki batas waktu.



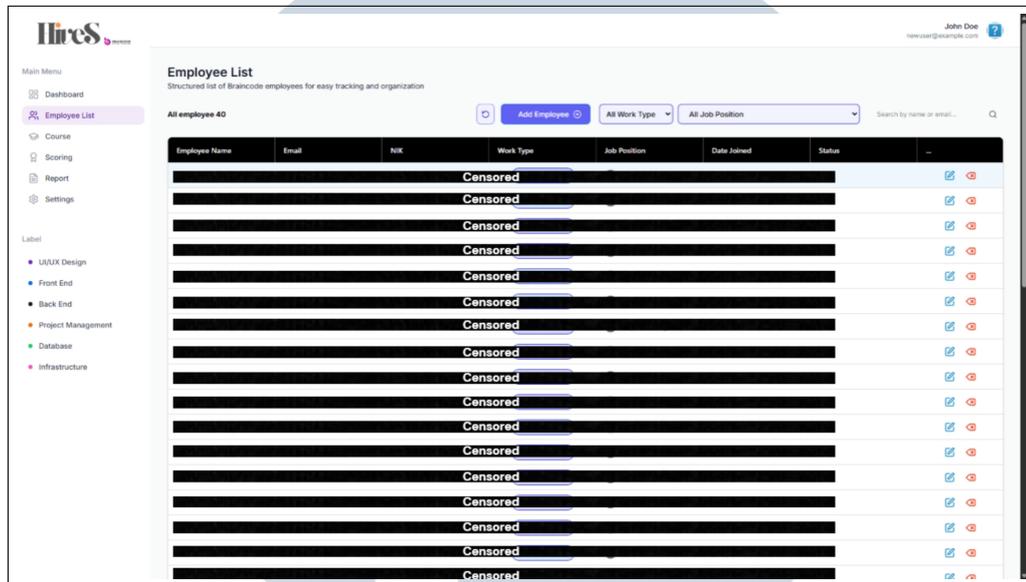
Gambar 3.8. Halaman reset password

D Halaman Employee List - (Admin)

Halaman *Employee List* menampilkan daftar lengkap karyawan dalam format tabel yang hanya dapat diakses oleh *admin*, dengan fitur manajemen data yang mencakup penambahan, pengeditan, dan penghapusan karyawan. Sistem ini dilengkapi dengan fungsi *filtering* berdasarkan *work type* dan *job position*, serta *search bar* untuk pencarian cepat berdasarkan nama atau email. Setiap data karyawan memuat informasi penting seperti *employee name*, *email*, *NIK*, *work type*, *job position*, *date joined*, dan *status* (aktif/tidak aktif).

Integrasi dengan sistem *LMS* memungkinkan pengelolaan akses pembelajaran berdasarkan kategori pekerjaan. Ketika halaman dimuat, data ditarik dari *API*, dan ditampilkan secara dinamis dengan dukungan *pagination* jika jumlah data besar. Tombol *Add Employee* serta ikon *edit* dan *delete* memungkinkan operasi *CRUD* secara efisien, dan setiap perubahan status atau atribut karyawan

disimpan melalui integrasi dengan API. Seluruh data dalam gambar disensor sebagai langkah standar keamanan untuk menjaga kerahasiaan informasi internal perusahaan [8].



Gambar 3.9. Halaman employee list

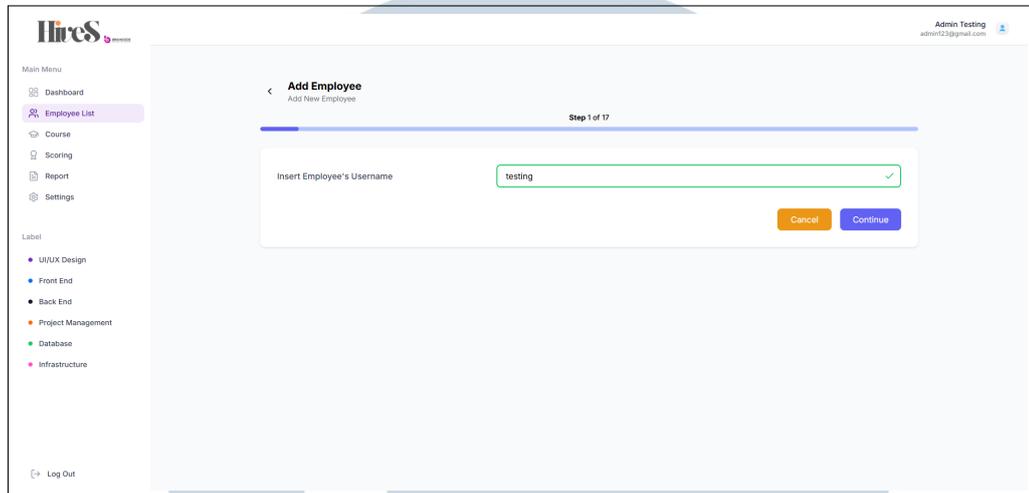
E Halaman Add Employee - (Admin)

Halaman *Add Employee* memungkinkan *admin* untuk menambahkan data karyawan baru ke dalam sistem *LMS HIVES*. Formulir ini terdiri dari beberapa bagian penting seperti pengisian *username*, nama lengkap, email, nomor induk karyawan (NIK), posisi pekerjaan, tipe pekerjaan, dan sebagainya. Setiap input divalidasi untuk memastikan integritas data, dan beberapa kolom seperti *job position* serta *work type* menggunakan *dropdown* yang bersumber dari data dinamis *API*. Proses penambahan data ini dirancang agar intuitif, responsif, dan terintegrasi langsung dengan database melalui layanan *endpoint* khusus.

E.1 Halaman Add Employee - Username

Halaman *Add Employee - Username* menyediakan validasi *real-time* untuk memastikan username tersedia dan sesuai format. Saat admin mengetik, sistem memeriksa ketersediaan melalui API dan menampilkan centang hijau jika valid. Terdapat tombol *Cancel* untuk kembali dan *Continue* untuk melanjutkan proses.

Proses ini terintegrasi dengan registrasi multi-step menggunakan penyimpanan *state* sementara.

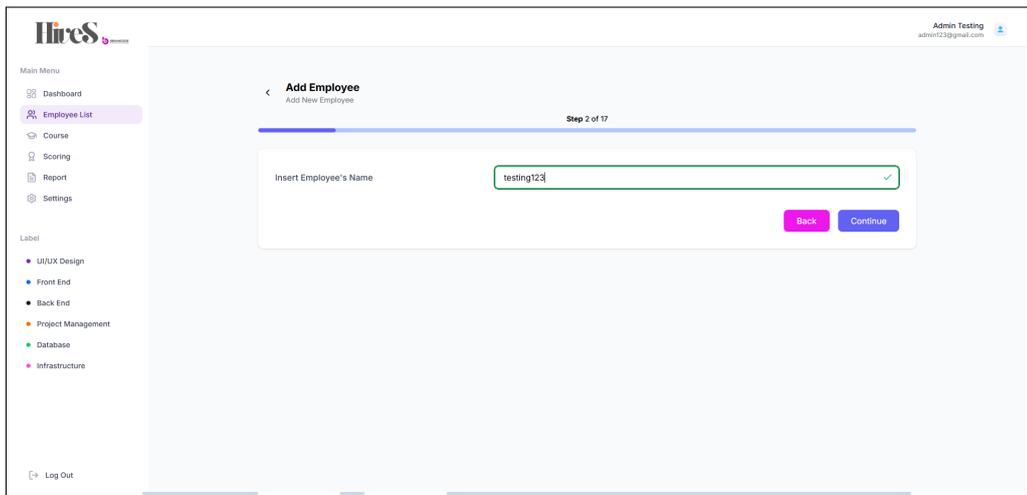


Gambar 3.10. Halaman add employee - username

E.2 Halaman Add Employee - Name

Halaman *Add Employee - Name* ini memungkinkan *admin* memasukkan nama karyawan dengan validasi *real-time*, memastikan format nama sesuai aturan (tidak kosong atau mengandung karakter khusus). Tersedia indikator visual seperti centang hijau saat nama valid, serta tombol *Back* dan *Continue* untuk navigasi antar langkah. Data dari langkah sebelumnya tetap tersimpan, mendukung navigasi dua arah tanpa kehilangan informasi. Proses ini terintegrasi dengan *session ID* dan sistem penyimpanan *state* untuk kelanjutan pendaftaran.

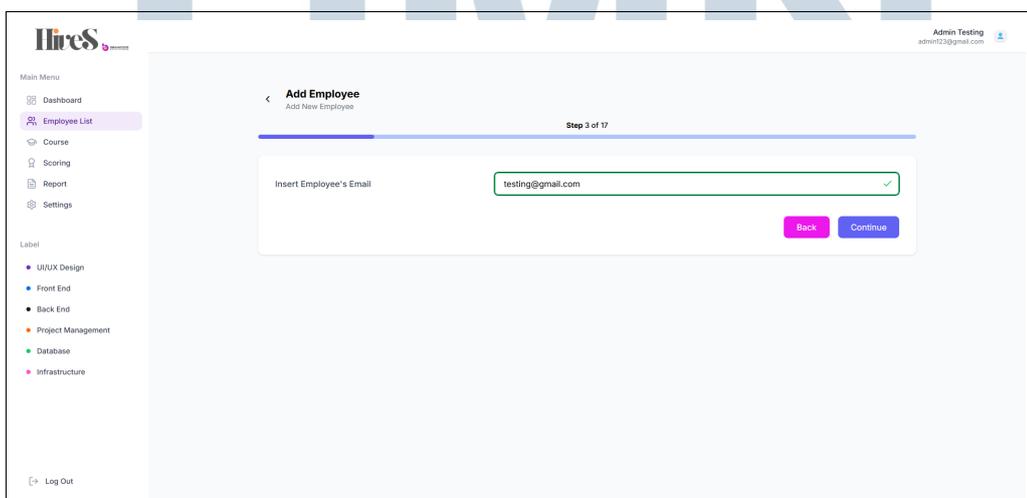
U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A



Gambar 3.11. Halaman add employee - name

E.3 Halaman Add Employee - Email

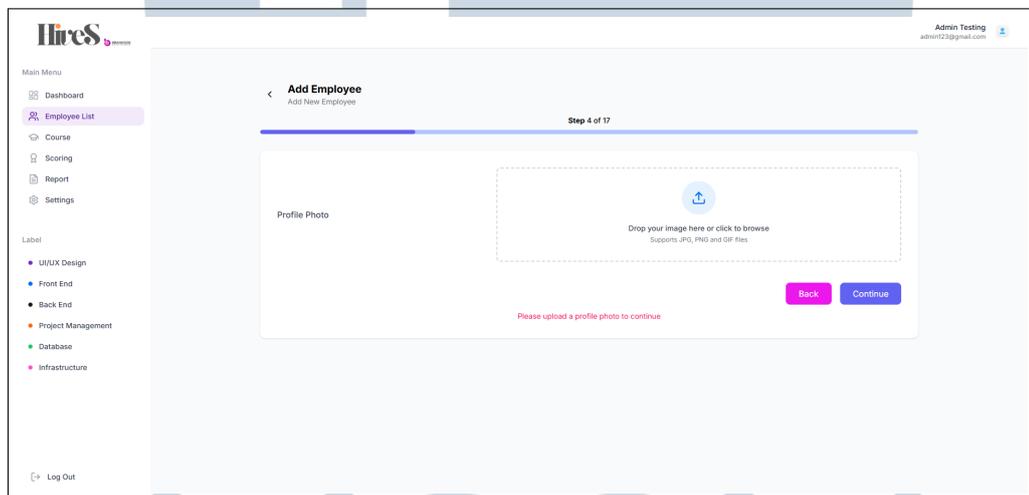
Halaman *Add Employee - Email* ini memungkinkan *admin* memasukkan email karyawan dengan validasi *real-time*, mencakup format standar, keunikan, dan domain yang sah. Indikator visual seperti centang hijau muncul saat email lolos validasi, sementara navigasi dua arah melalui tombol *Back* dan *Continue* memungkinkan fleksibilitas antar langkah. Email yang telah diinput ditambahkan ke state form, dan sistem memanfaatkan session ID untuk menjaga alur registrasi tetap sinkron. Validasi dilakukan baik di sisi frontend maupun backend untuk memastikan integritas data.



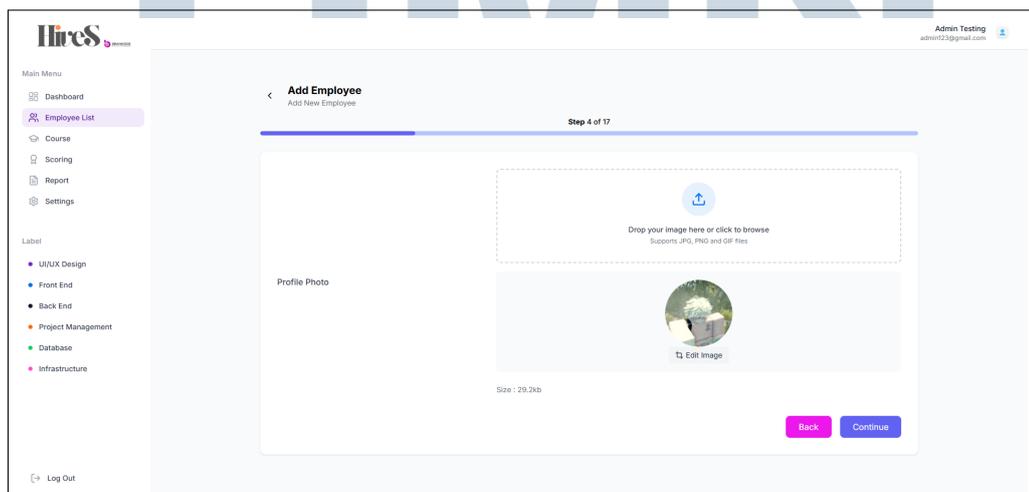
Gambar 3.12. Halaman add employee - email

E.4 Halaman Add Employee - Profile Photo

Halaman *Add Employee - Profile Photo* ini memungkinkan *admin* mengunggah foto profil karyawan melalui metode *drag & drop* atau *browse file*, dengan validasi terhadap tipe file (*JPG, PNG, GIF*) dan ukuran maksimal 500KB. Jika file tidak valid atau belum diunggah, sistem menampilkan *warning* dan menonaktifkan tombol *Continue*. Setelah foto valid berhasil diunggah, sistem menampilkan *preview* dan mengirimkannya ke server melalui API. Navigasi tersedia dengan tombol *Back* untuk kembali tanpa kehilangan data, dan *Continue* untuk melanjutkan jika validasi terpenuhi.



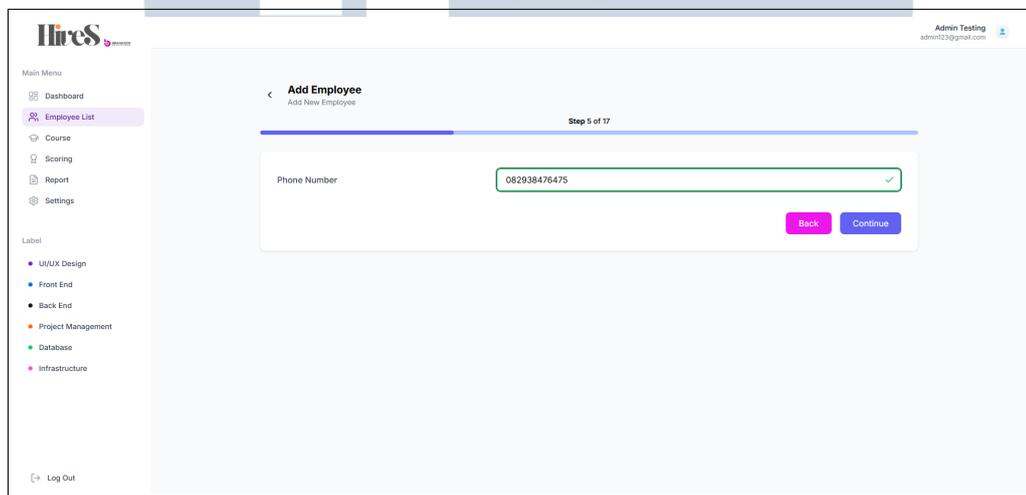
Gambar 3.13. Halaman add employee - profile photo (tampilan awal)



Gambar 3.14. Halaman add employee - profile photo (setelah upload)

E.5 Halaman Add Employee - Phone Number

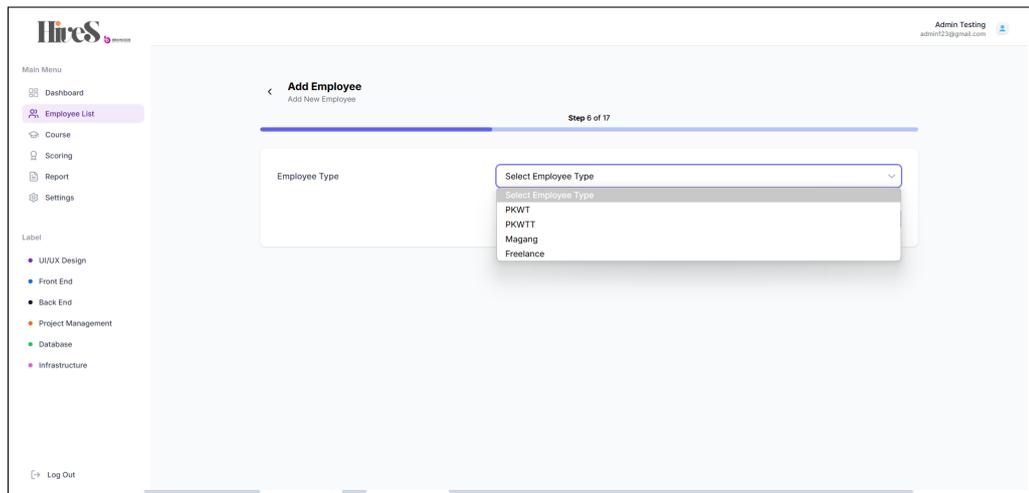
Halaman *Add Employee - Profile Photo* ini digunakan untuk menginput dan memvalidasi nomor telepon karyawan dengan format sesuai standar Indonesia, yaitu hanya angka, panjang 10–13 digit. Validasi dilakukan secara *real-time*, dengan indikator visual (centang hijau) jika input sesuai. Data nomor telepon akan disimpan dalam *state* form sebagai bagian dari sesi registrasi *multi-step*, bersama data sebelumnya. Tombol *Back* memungkinkan kembali ke langkah sebelumnya tanpa kehilangan data, sementara *Continue* hanya aktif jika input valid dan belum terdaftar dalam sistem.

The screenshot shows a web application interface for adding a new employee. On the left is a sidebar menu with options like Dashboard, Employee List, Course, Scoring, Report, and Settings. The main content area is titled 'Add Employee' and 'Add New Employee'. It features a progress bar indicating 'Step 5 of 17'. Below the progress bar is a form with a 'Phone Number' label and a text input field containing '082938476475'. A green checkmark is visible to the right of the input field, signifying successful validation. At the bottom right of the form are two buttons: a purple 'Back' button and a blue 'Continue' button. The top right corner of the page shows the user 'Admin Testing' with the email 'admin123@gmail.com'. A 'Log Out' link is located at the bottom left of the sidebar.

Gambar 3.15. Halaman add employee - phone number

E.6 Halaman Add Employee - Employee Type

Pada halaman *Add Employee - Employee Type* ini, *admin* diminta memilih tipe karyawan melalui *dropdown selector* yang berisi empat kategori: *PKWT*, *PKWTT*, *Magang*, dan *Freelance*. Pemilihan ini wajib dilakukan karena akan berdampak langsung pada pengelolaan hak akses, kebijakan kerja, dan alur administrasi karyawan di sistem. Sistem akan memvalidasi input dan hanya mengaktifkan tombol *Continue* jika salah satu tipe telah dipilih, sementara tombol *Back* tetap tersedia untuk navigasi ke langkah sebelumnya. Informasi tipe karyawan ini disimpan dalam *state* dan dapat memengaruhi form lanjutan seperti penentuan durasi kontrak pada tipe *PKWT*.

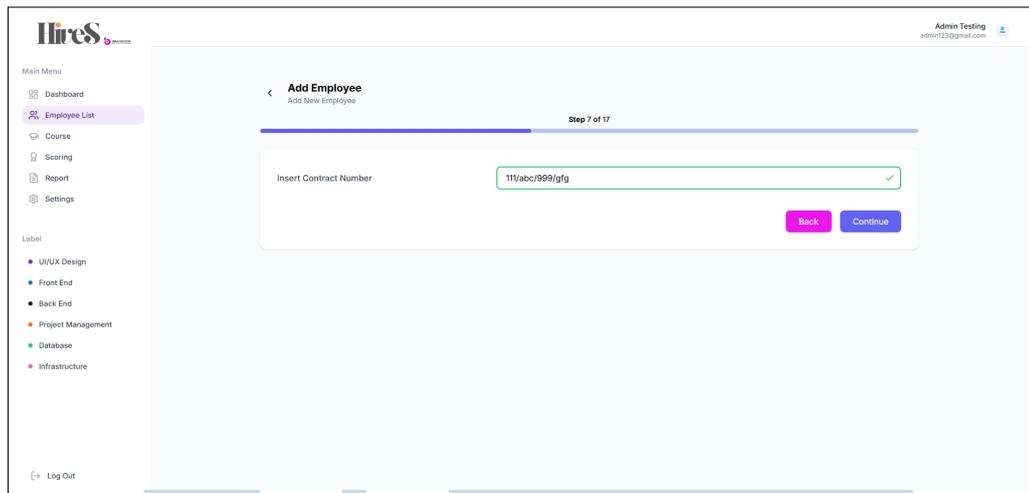


Gambar 3.16. Halaman add employee - employee type

E.7 Halaman Add Employee - Contract Number

Pada halaman *Add Employee - Contract Number*, *admin* diminta memasukkan *contract number* karyawan yang mengikuti format standar perusahaan, seperti kombinasi angka, huruf, dan simbol khusus (misalnya “111/abc/999/gfg”). Validasi dilakukan secara *real-time* untuk memastikan format yang sesuai serta keunikan nomor tersebut dalam database. Field ini bersifat *adaptive*, hanya muncul untuk tipe karyawan tertentu seperti *PKWT* atau *PKWTT*, dan dapat bersifat opsional bagi *Magang* atau *Freelance*. Navigasi antar langkah tetap konsisten, dengan tombol *Back* untuk kembali ke *Employee Type* dan *Continue* untuk melanjutkan ke langkah berikutnya, hanya aktif jika input sudah valid. Data nomor kontrak yang lolos validasi akan disimpan ke dalam *state* registrasi sebagai bagian dari informasi karyawan.

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A

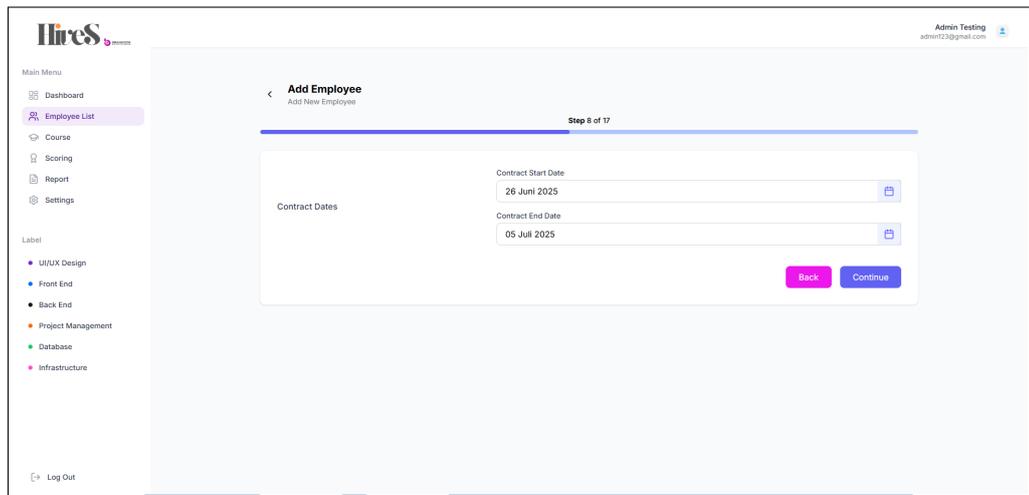


Gambar 3.17. Halaman add employee - contract number

E.8 Halaman Add Employee - Contract Dates

Pada halaman *Add Employee - Contract Number*, *admin* diminta mengisi *tanggal mulai* dan *tanggal akhir kontrak* dengan format lokal Indonesia (misalnya “26 Juni 2025”), menggunakan *date picker* yang memudahkan pemilihan secara visual. Field ini bersifat *adaptive* tergantung tipe karyawan—misalnya, untuk tipe *PKWT* kedua tanggal wajib diisi dan tervalidasi agar tanggal akhir lebih besar dari tanggal mulai, sedangkan untuk *PKWTT*, tanggal akhir bisa opsional atau bahkan tidak muncul. Validasi juga memastikan bahwa tanggal mulai tidak berada di masa lalu. Setelah validasi berhasil, data tanggal disimpan dalam *form state* dan digunakan untuk memperbarui *session* registrasi karyawan. Navigasi dilakukan melalui tombol *Back* untuk kembali ke *Contract Number* dan *Continue* ke langkah berikutnya.

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A

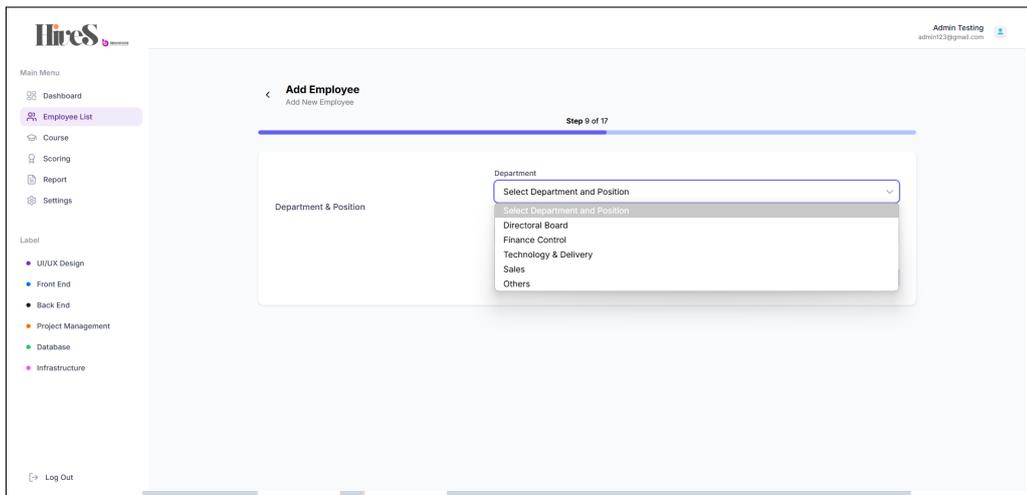


Gambar 3.18. Halaman add employee - contract date

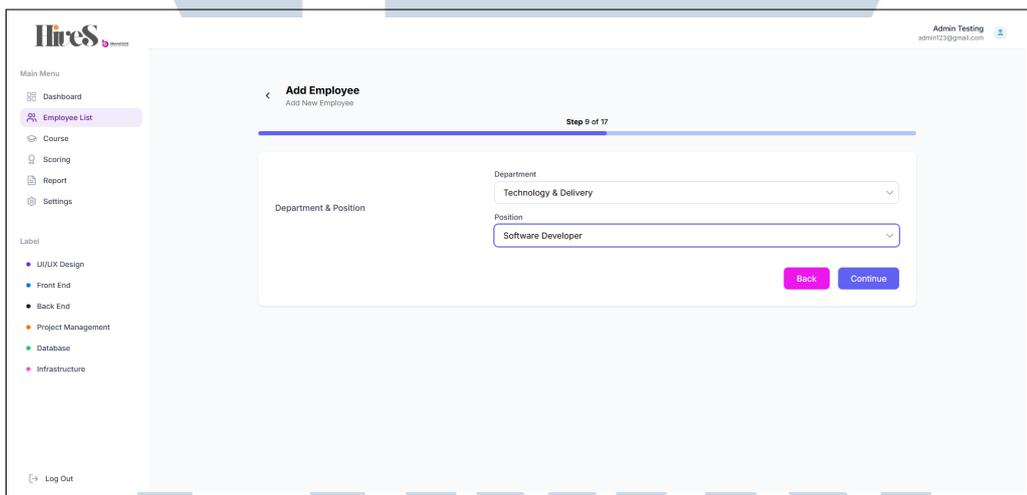
E.9 Halaman Add Employee - Department and Position

Pada halaman *Add Employee - Department and Position*, admin diminta memilih *Department* dan *Position* untuk karyawan baru melalui dua dropdown yang saling terkait (*cascading dropdowns*). Dropdown departemen menampilkan daftar departemen yang telah dikonfigurasi dalam sistem, dan setelah admin memilih misalnya “Technology & Delivery”, dropdown posisi akan otomatis menampilkan opsi yang relevan seperti “Software Developer”. Kedua field ini wajib diisi untuk melanjutkan ke tahap berikutnya, dan pilihan yang dipilih akan disimpan dalam *form state* untuk memengaruhi hak akses serta jalur pembelajaran karyawan. Tombol *Back* dan *Continue* tetap disediakan untuk navigasi dua arah antar langkah registrasi multi-step.

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A



Gambar 3.19. Halaman add employee - department and position (bagian department)

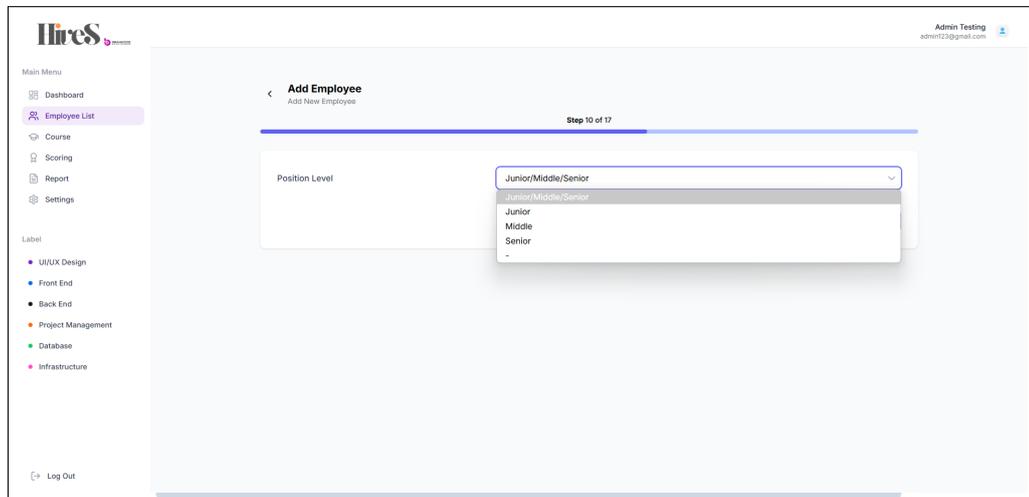


Gambar 3.20. Halaman add employee - department and position (bagian position)

E.10 Halaman Add Employee - Position Level

Pada halaman *Add Employee - Position Level*, admin memilih *Position Level* atau tingkat senioritas dari karyawan, seperti *Junior*, *Middle*, atau *Senior*, melalui *dropdown selector*. Pilihan ini melengkapi posisi yang dipilih sebelumnya, sehingga membentuk *job title* lengkap seperti “Senior Software Developer”. Level ini tidak hanya memperjelas klasifikasi pekerjaan, tetapi juga menentukan jalur pembelajaran, ekspektasi kinerja, dan tanggung jawab dalam sistem. Sistem menyimpan pilihan level dalam *form state* dan hanya mengizinkan lanjut ke langkah berikutnya jika level yang dipilih valid serta sesuai dengan kombinasi posisi

sebelumnya.

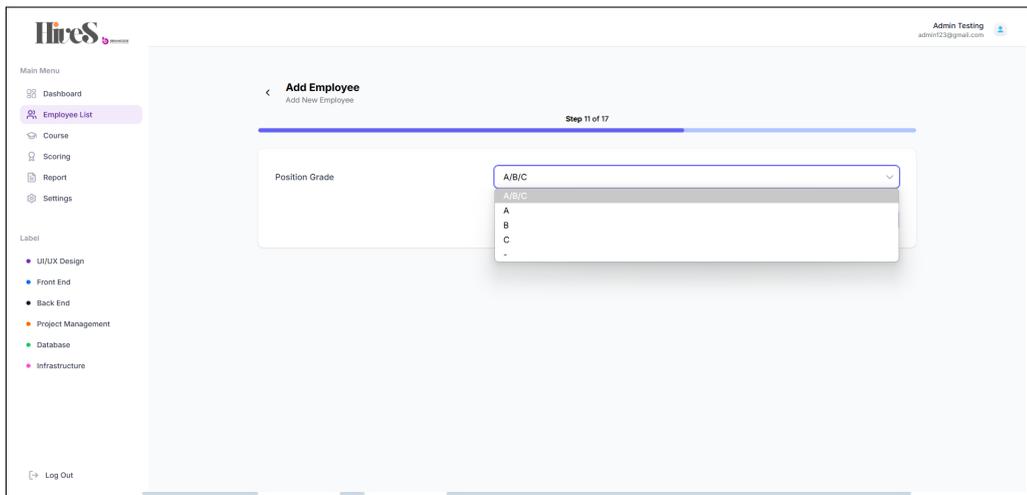


Gambar 3.21. Halaman add employee - position level

E.11 Halaman Add Employee - Position Grade

Pada halaman *Add Employee - Position Grade*, admin memilih *Position Grade* melalui dropdown dengan opsi seperti A, B, C, atau - untuk jabatan yang belum ditentukan. Grade memberikan klasifikasi tambahan terhadap jabatan berdasarkan tanggung jawab, tingkat keahlian, serta struktur kompensasi. Kombinasi antara posisi, level, dan grade membentuk identitas jabatan yang sangat spesifik dalam organisasi, misalnya *Senior Software Developer Grade A*. Sistem menyimpan pilihan ini dalam *form state*, memverifikasi kombinasi valid, dan hanya mengizinkan lanjut ke langkah berikutnya jika pemilihan grade sesuai dengan struktur organisasi dan aturan validasi sistem.

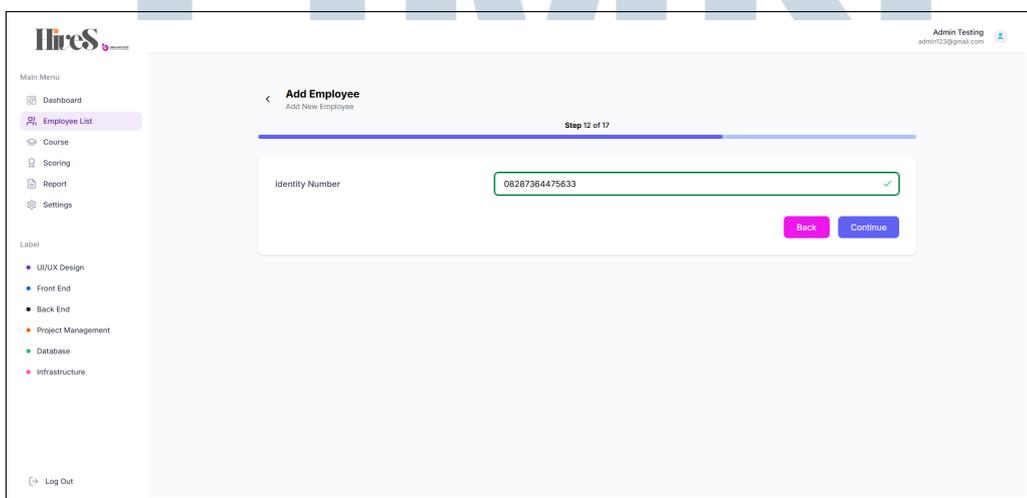
U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A



Gambar 3.22. Halaman add employee - position grade

E.12 Halaman Add Employee - Identity Number

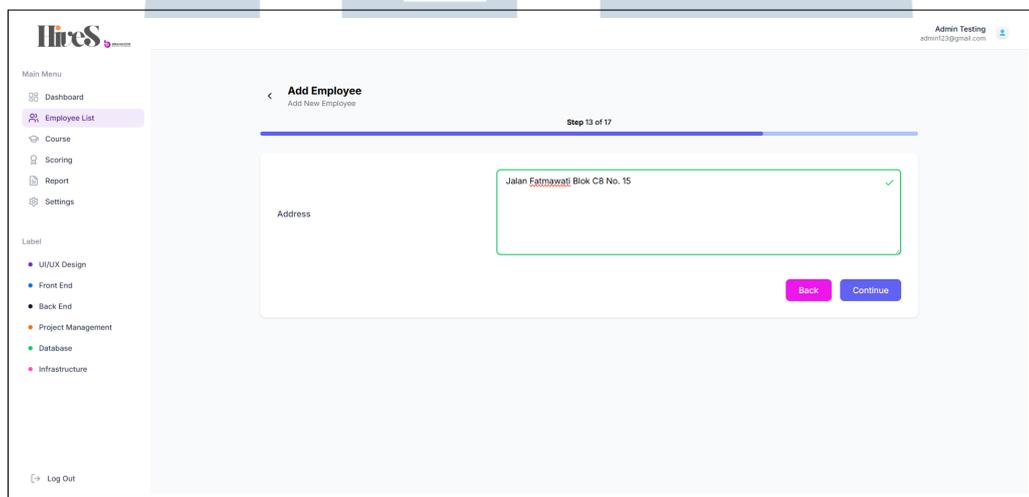
Pada halaman *Add Employee - Identity Number*, *admin* diminta mengisi *identity number* resmi karyawan, seperti NIK atau NPWP, melalui field input yang divalidasi secara *real-time*. Sistem memverifikasi kevalidan format numerik dan keunikan nomor untuk mencegah duplikasi data. Nomor identitas yang sah ditandai dengan indikator visual berupa centang hijau dan akan disimpan dalam *form state* sebagai bagian dari data karyawan. Informasi ini dikumpulkan untuk memenuhi kebutuhan kepatuhan hukum dan pelaporan administratif, dan sistem hanya mengizinkan lanjut ke langkah berikutnya jika validasi berhasil.



Gambar 3.23. Halaman add employee - identity number

E.13 Halaman Add Employee - Address

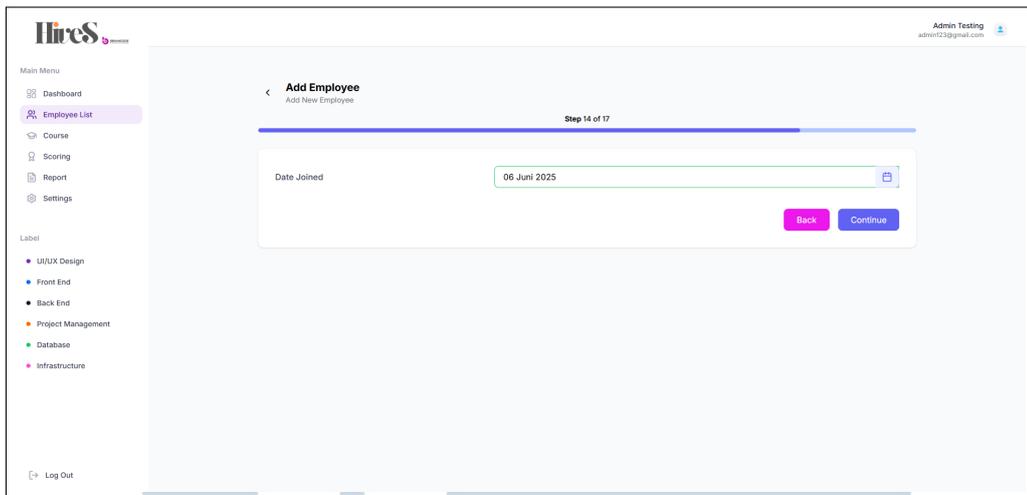
Pada halaman *Add Employee - Address*, *admin* diminta mengisi *alamat lengkap* karyawan melalui field *textarea* yang mendukung input multi-baris. Sistem melakukan validasi *real-time* untuk memastikan alamat tidak kosong. Input yang valid akan ditandai dengan centang hijau dan disimpan dalam *form state* sebagai bagian dari data kontak karyawan. Informasi ini penting untuk keperluan komunikasi, dokumentasi, dan melengkapi data personal sebelum berlanjut ke langkah berikutnya.



Gambar 3.24. Halaman add employee - address

E.14 Halaman Add Employee - Date Joined

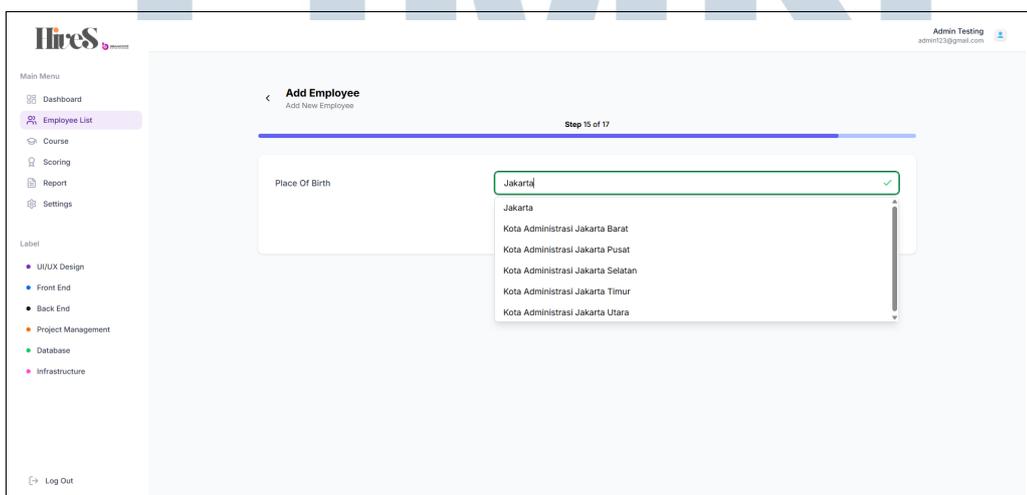
Pada halaman *Add Employee - Date Joined*, *admin* diminta mengisi *tanggal bergabung* karyawan menggunakan field dengan *date picker* berformat lokal (DD Bulan YYYY). Validasi dilakukan secara *real-time* untuk memastikan tanggal valid, tidak di masa depan yang ekstrem, serta konsisten dengan tanggal kontrak yang telah diisi sebelumnya. Tanggal bergabung menjadi penanda resmi dimulainya masa kerja karyawan, serta digunakan untuk perhitungan benefit dan jalur pembelajaran awal. Setelah input valid, data disimpan dalam *form state* dan proses dapat dilanjutkan ke langkah berikutnya.



Gambar 3.25. Halaman add employee - date joined

E.15 Halaman Add Employee - Place of Birth

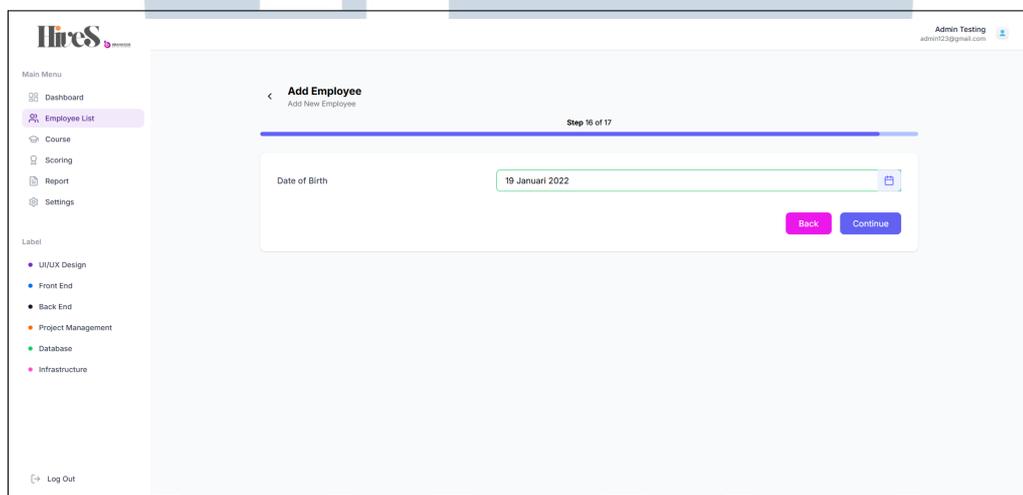
Pada halaman *Add Employee - Place of Birth*, admin mengisi *tempat lahir* karyawan melalui field input dengan fitur *autocomplete* yang menampilkan daftar kota di Indonesia. Sistem secara *real-time* memfilter lokasi berdasarkan input yang diketik, menampilkan saran yang relevan dan hanya memperbolehkan lokasi valid dari database. Indikator visual seperti centang hijau muncul untuk memastikan tempat lahir sudah sesuai. Data yang valid akan ditambahkan ke *form state*, dan *admin* dapat menavigasi ke langkah selanjutnya atau kembali ke langkah sebelumnya dengan tetap mempertahankan data yang telah dimasukkan.



Gambar 3.26. Halaman add employee - place of birth

E.16 Halaman Add Employee - Date of Birth

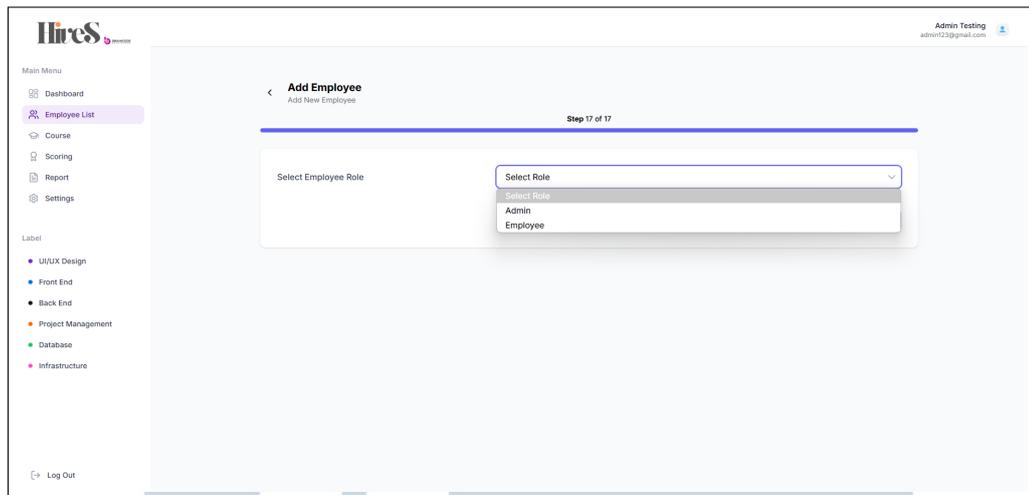
Pada halaman *Add Employee - Date of Birth*, *admin* mengisi tanggal lahir karyawan melalui field input bertipe tanggal dengan tampilan yang user-friendly dan dilengkapi *date picker* untuk mempermudah pemilihan. Sistem melakukan validasi *real-time* terhadap format tanggal, memastikan tanggal tidak berada di masa depan. Informasi ini melengkapi data identitas personal seperti *tempat lahir* dari langkah sebelumnya dan menjadi bagian dari *form state*. Setelah valid, *admin* dapat melanjutkan ke langkah terakhir atau kembali ke langkah sebelumnya tanpa kehilangan data.

The screenshot shows a web application interface for 'HireS'. On the left is a 'Main Menu' sidebar with options: Dashboard, Employee List (highlighted), Course, Scoring, Report, and Settings. Below the menu is a 'Label' section with categories: UI/UX Design, Front End, Back End, Project Management, Database, and Infrastructure. At the bottom left is a 'Log Out' button. The main content area is titled 'Add Employee' with a subtitle 'Add New Employees'. It shows 'Step 16 of 17' in a progress bar. The 'Date of Birth' field is active, displaying '19 Januari 2022' with a calendar icon. Below the field are 'Back' and 'Continue' buttons. The top right corner shows the user 'Admin Testing' with email 'admin123@gmail.com'.

Gambar 3.27. Halaman add employee - date of birth

E.17 Halaman Add Employee - Role

Pada halaman *Add Employee - Role*, *admin* diminta memilih *role* sistem untuk karyawan yang akan dibuat, dengan opsi *Admin* atau *Employee* melalui dropdown selector. Role ini menentukan tingkat akses pengguna dalam sistem LMS — *Admin* memiliki hak untuk mengelola user, konten, dan pengaturan sistem, sedangkan *Employee* hanya memiliki akses sebagai learner. Mekanisme ini menerapkan *Role-Based Access Control (RBAC)* untuk memastikan keamanan dan segmentasi akses. Setelah role dipilih, data akhir akan divalidasi dan proses pendaftaran diselesaikan dengan pembuatan akun serta penugasan hak akses sesuai role yang ditentukan.



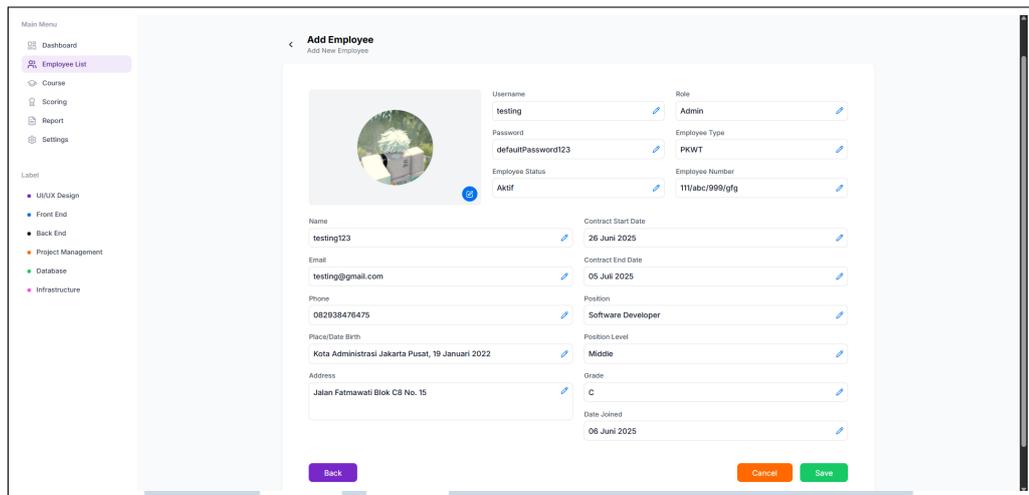
Gambar 3.28. Halaman add employee - role

E.18 Halaman Add Employee - Employee Resume

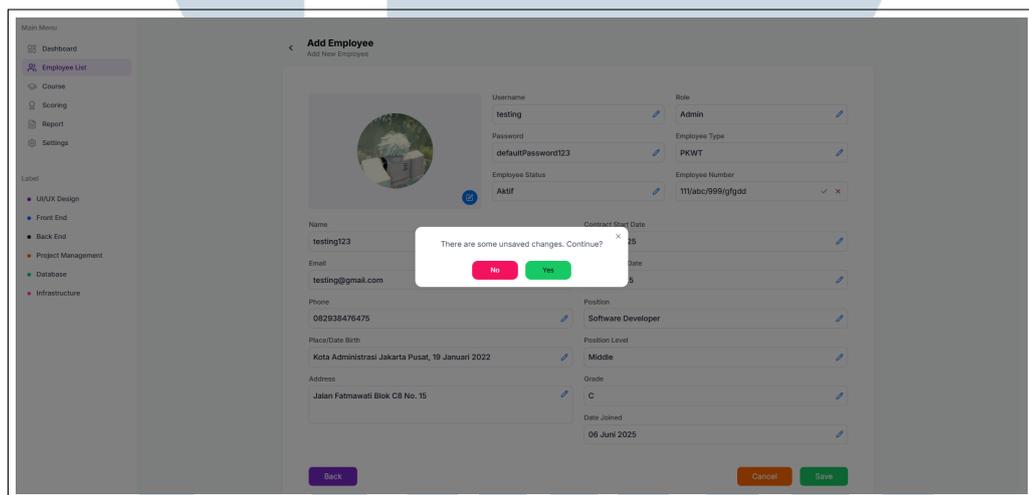
Pada tahap terakhir dari proses *Add Employee* yaitu halaman *Employee Resume*, sistem menampilkan *resume* atau ringkasan lengkap seluruh data karyawan yang telah diisi dari langkah 1 hingga 17. Informasi ditampilkan secara terstruktur berdasarkan kategori, memudahkan admin untuk melakukan pengecekan akhir. Fitur *inline editing* memungkinkan setiap field dapat diedit langsung dengan ikon pensil, disertai validasi *real-time*, notifikasi kesalahan, serta opsi *save* dan *cancel*.

Admin juga dapat mengunggah dan mengedit foto profil karyawan menggunakan editor foto interaktif yang mendukung *crop*, *zoom*, dan *rotate*. Setelah semua data direview dan valid, admin dapat mengklik tombol *Save* untuk melakukan *final submission*. Sistem kemudian akan melakukan validasi menyeluruh dan mengirim data ke API untuk menyimpan informasi karyawan dan foto profil, lalu menampilkan notifikasi sukses dan mengarahkan pengguna ke halaman *Employee List*.

UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.29. Halaman add employee - employee resume



Gambar 3.30. Halaman add employee - employee resume (warning modal)

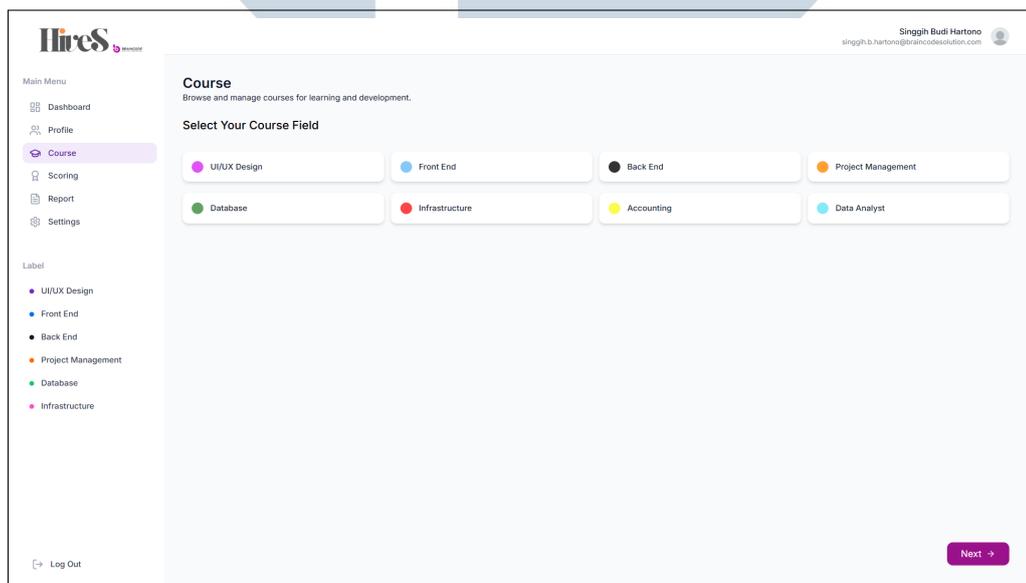
F Halaman Course (Employee)

Pada bagian ini, ditampilkan antarmuka yang digunakan oleh karyawan untuk mengakses daftar kursus yang tersedia. Halaman ini dirancang untuk memberikan pengalaman belajar yang intuitif dan terstruktur, sehingga memudahkan pengguna dalam menjelajahi materi pembelajaran sesuai bidang atau tingkatan yang relevan. Setiap komponen halaman memiliki peran penting dalam menyajikan informasi kursus secara interaktif, mulai dari daftar bidang pembelajaran, daftar kursus per bidang, hingga detail masing-masing kursus.

F.1 Halaman Course Field

Halaman *Course Field* merupakan langkah awal dalam proses penelusuran kursus, di mana pengguna diminta untuk memilih bidang atau kategori kursus sesuai minat atau keahliannya. Setiap bidang ditampilkan dalam bentuk *visual card* berwarna yang mewakili masing-masing kategori, memberikan pengalaman visual yang menarik dan intuitif.

Pemilihan bidang ini bersifat penting karena digunakan sebagai *filter* utama untuk menampilkan daftar kursus yang relevan di langkah berikutnya. Sistem menyimpan pilihan pengguna dalam *state* menggunakan custom hook *useCourseFilter*, yang kemudian diteruskan sebagai parameter ke API saat memuat daftar kursus. Setelah pengguna memilih salah satu bidang dan menekan tombol *Next*, sistem akan menavigasi ke halaman daftar kursus yang telah dipersonalisasi berdasarkan pilihan tersebut. Dengan pendekatan ini, pengalaman belajar menjadi lebih terarah dan sesuai kebutuhan pengguna [9].



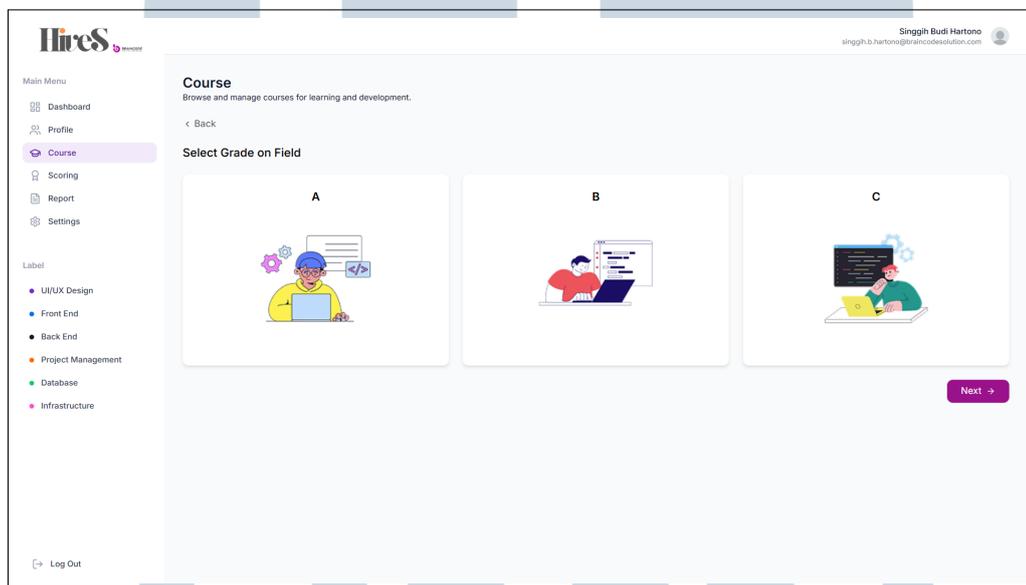
Gambar 3.31. Halaman course field

F.2 Halaman Course Grade

Halaman *Course Grade* merupakan langkah kedua dalam proses penyaringan kursus, di mana pengguna diminta untuk memilih tingkat kesulitan atau *grade* kursus yang sesuai dengan tingkat keahlian mereka. Setiap *grade* ditampilkan dalam bentuk *visual card* dengan ilustrasi representatif, mulai dari

Grade C untuk pemula, Grade B untuk tingkat menengah, hingga Grade A untuk tingkat mahir.

Pemilihan *grade* ini akan dikombinasikan dengan pilihan bidang (*field*) dari langkah sebelumnya untuk menghasilkan daftar kursus yang telah dipersonalisasi. Sistem menyimpan kombinasi *field* dan *grade* dalam *state* menggunakan *useCourseFilter*, yang kemudian digunakan untuk melakukan permintaan data ke API. Setelah pengguna memilih salah satu *grade* dan menekan tombol *Next*, sistem akan mengarahkan ke halaman daftar kursus yang telah difilter, memastikan materi yang ditampilkan relevan dengan tingkat pengalaman pengguna.

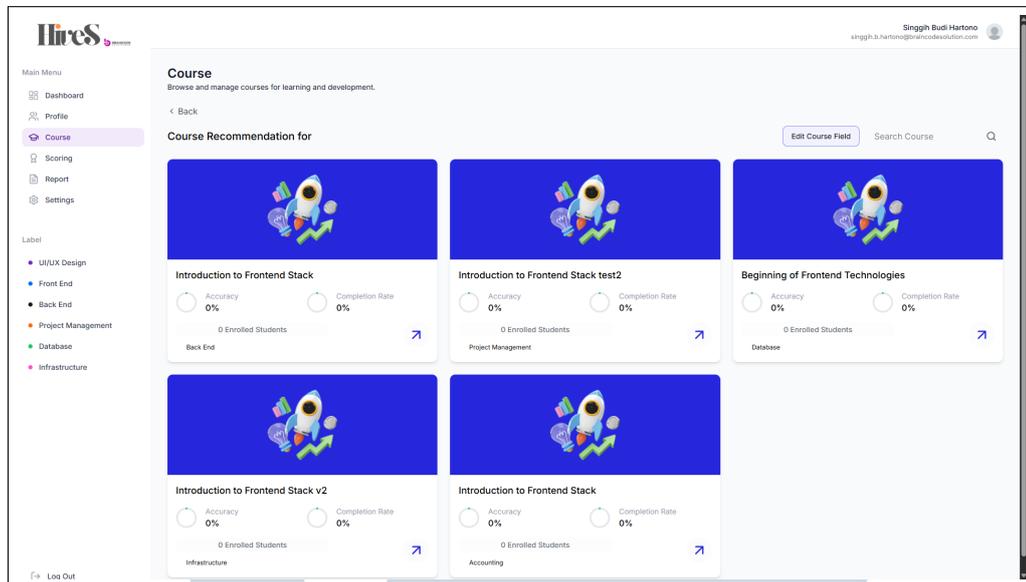


Gambar 3.32. Halaman course grade

E.3 Halaman Course List

Halaman *Course List* menampilkan katalog kursus dalam format *visual card* yang menarik, menyesuaikan dengan filter *field* dan *grade* yang telah dipilih sebelumnya oleh pengguna. Masing-masing kursus ditampilkan beserta informasi penting seperti judul, deskripsi singkat, serta metrik performa seperti tingkat penyelesaian (*completion rate*), jumlah siswa terdaftar, dan akurasi pembelajaran. Pengguna dapat menggunakan *search bar* untuk mencari kursus berdasarkan kata kunci, atau mengedit preferensi bidang melalui tombol *Edit Course Field*. Ketika pengguna mengklik salah satu kartu kursus, mereka akan diarahkan ke halaman detail kursus. Semua interaksi ini dikelola menggunakan *state* yang menyimpan filter aktif serta daftar kursus yang ditampilkan, dan sistem terintegrasi dengan

berbagai *API service* untuk menyajikan data yang relevan secara *real-time* [10].

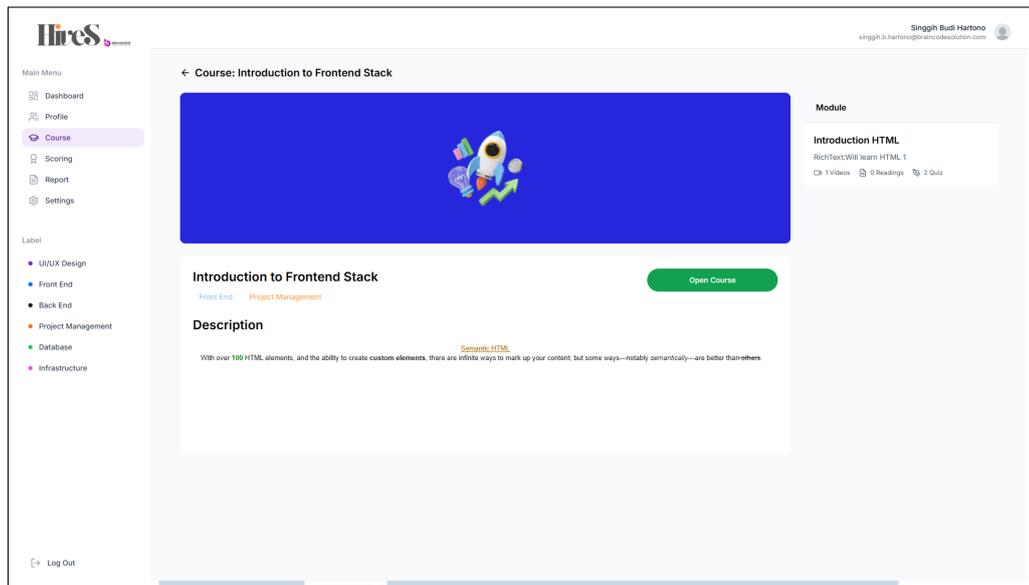


Gambar 3.33. Halaman course list

F.4 Halaman Course Detail

Halaman *Course Detail* menyajikan informasi lengkap mengenai sebuah kursus, termasuk deskripsi mendalam, struktur modul, jenis konten seperti *video*, *readings*, dan *quiz*, serta tag kategori yang mencerminkan bidang terkait seperti *Front End* atau *Project Management*. Pengguna dapat menelusuri isi kursus melalui tampilan *preview* yang informatif, kemudian mengklik tombol *Enroll Course* untuk mendaftar atau tombol *Open Course* jika sebelumnya sudah pernah mendaftar untuk langsung memulai proses pembelajaran.

Sistem akan memverifikasi status pendaftaran pengguna, menyesuaikan tombol aksi, dan jika pengguna belum terdaftar, akan membuat data *enrollment* baru di database sebelum mengarahkan ke modul pertama. *Sidebar* kanan menampilkan struktur pembelajaran yang terorganisir, sementara *breadcrumb* membantu navigasi kembali ke halaman sebelumnya. Data ditampilkan berdasarkan *course ID* yang dipilih, dan seluruh proses terintegrasi dengan berbagai *API*, memastikan pengalaman belajar yang informatif, personal, dan lancar.

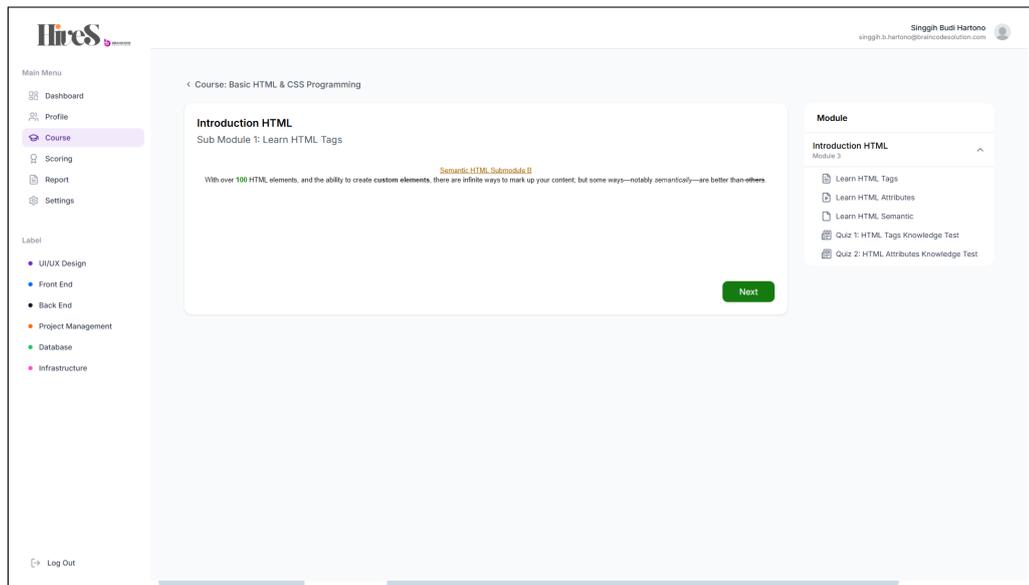


Gambar 3.34. Halaman course detail

F.5 Halaman SubModule - FreeText

Halaman *Submodule - FreeText* menampilkan konten pembelajaran berbasis teks yang disusun secara hierarkis dalam struktur *Course* ; *Module* ; *Sub Module* ; *Content Item*. Materi ditampilkan dalam format *rich text* dengan dukungan *formatting* seperti *bold*, *italic*, dan *hyperlink*, termasuk kemampuan menyisipkan tautan internal ke submodul lain yang relevan. Navigasi pembelajaran disediakan melalui *breadcrumb*, *sidebar* modul, serta tombol *Next* untuk alur linier. Saat pengguna mengakses halaman ini, sistem memuat konten dari database berdasarkan ID, merender dalam HTML yang aman, dan mencatat progress belajar melalui *API*.

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A

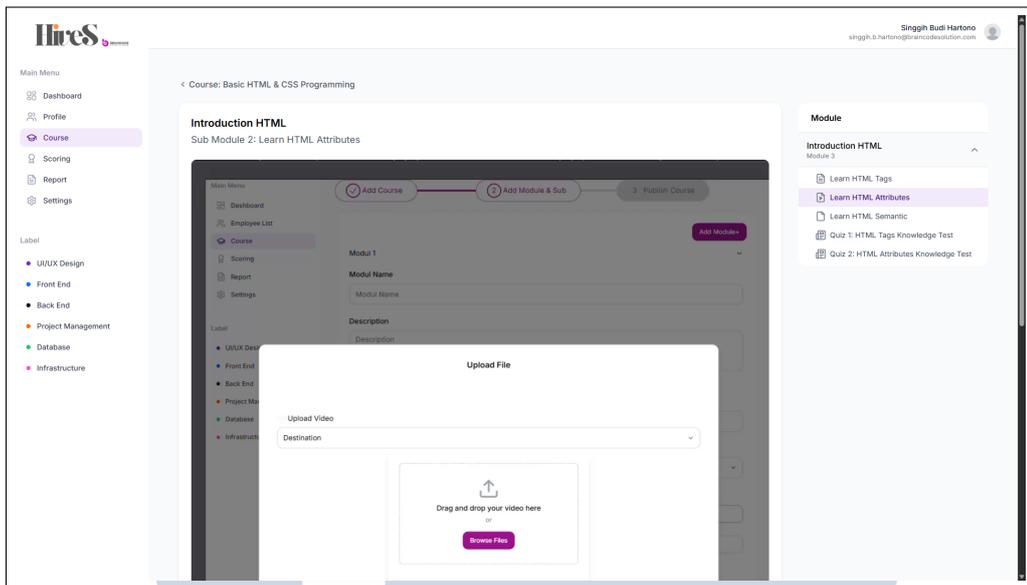


Gambar 3.35. Halaman submodule freetext

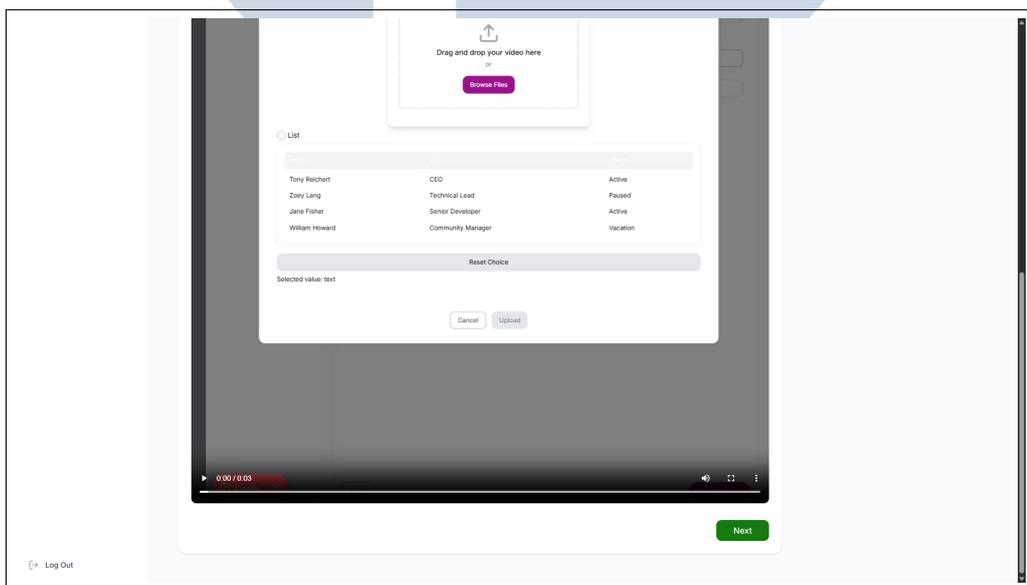
F.6 Halaman SubModule - Video

Halaman *Submodule - Video* menyajikan materi pembelajaran dalam bentuk video yang terintegrasi dalam alur kursus. Kontennya diputar melalui *video player* dengan kontrol standar. Navigasi pembelajaran didukung *breadcrumb*, *sidebar*, dan tombol *Next* agar pengguna dapat melanjutkan ke submodul berikutnya secara linier. Semua data dan status ditangani melalui berbagai endpoint API, memastikan pengalaman belajar berbasis video yang mulus, terstruktur, dan dapat dilanjutkan kapan saja.

U M M N
U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A



Gambar 3.36. Halaman submodule video - tampilan atas

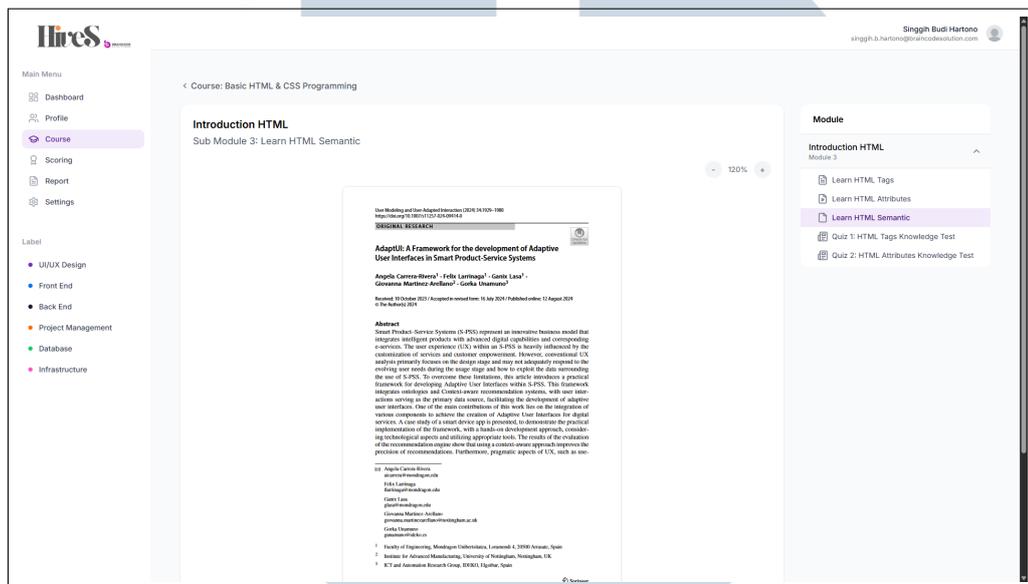


Gambar 3.37. Halaman submodule video - tampilan bawah

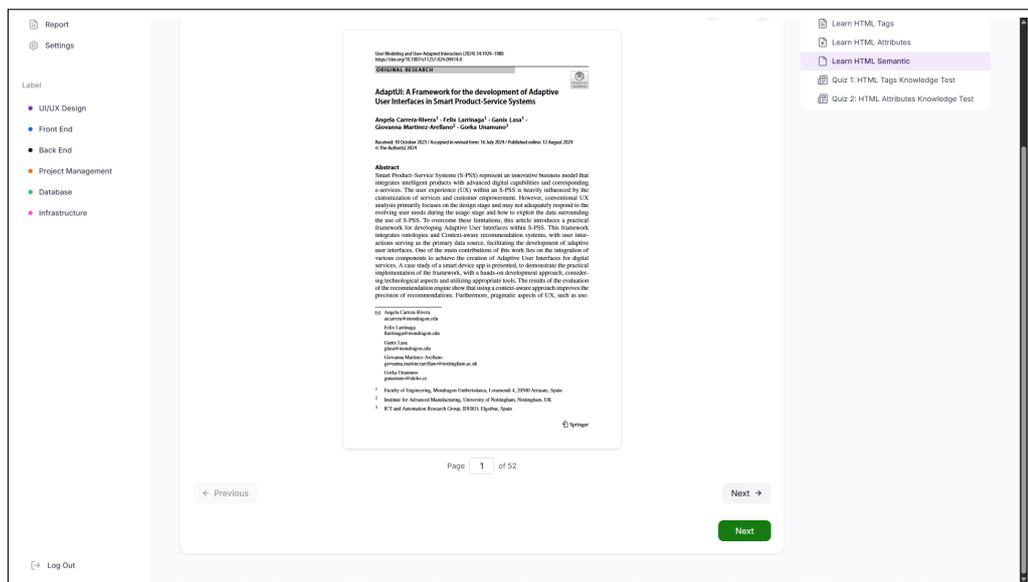
F.7 Halaman SubModule - PDF

Halaman *Submodule - PDF* menyediakan pengalaman membaca dokumen langsung dalam platform melalui *PDF viewer* terintegrasi dengan kualitas rendering tinggi dan dukungan fitur zoom. Pengguna dapat menavigasi dokumen multi-halaman menggunakan kontrol *pagination*, tombol *Previous/Next*, serta indikator

halaman saat ini. Sebagai bagian dari alur pembelajaran terstruktur, konten PDF digunakan untuk referensi penting atau bacaan wajib. Selain itu, dari sisi teknis, dokumen dimuat secara progresif dari server serta dirender dengan *library* seperti *PDF.js* untuk performa optimal.



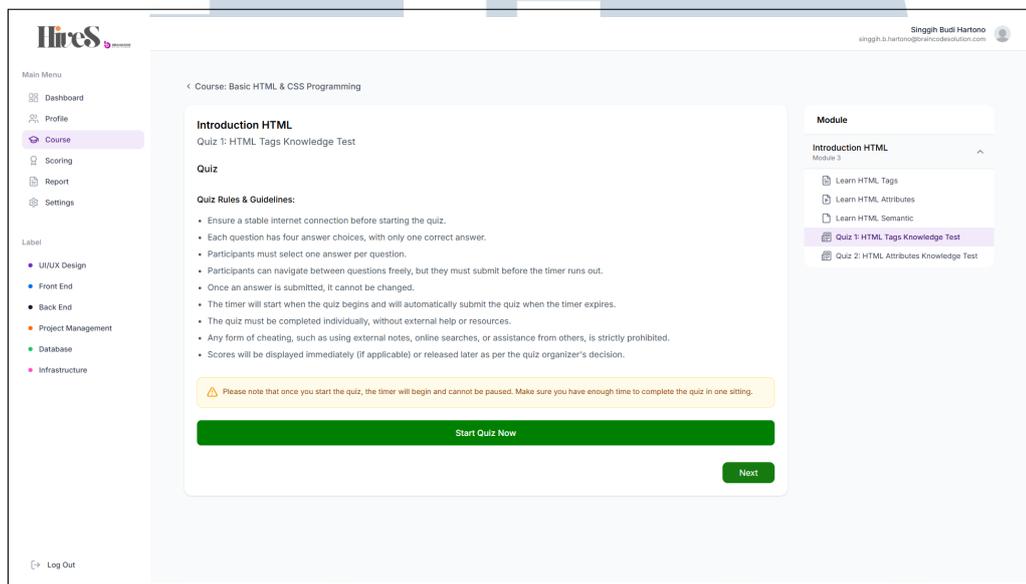
Gambar 3.38. Halaman submodule pdf - tampilan atas



Gambar 3.39. Halaman subModule pdf - tampilan bawah

F.8 Halaman Quiz Instruction

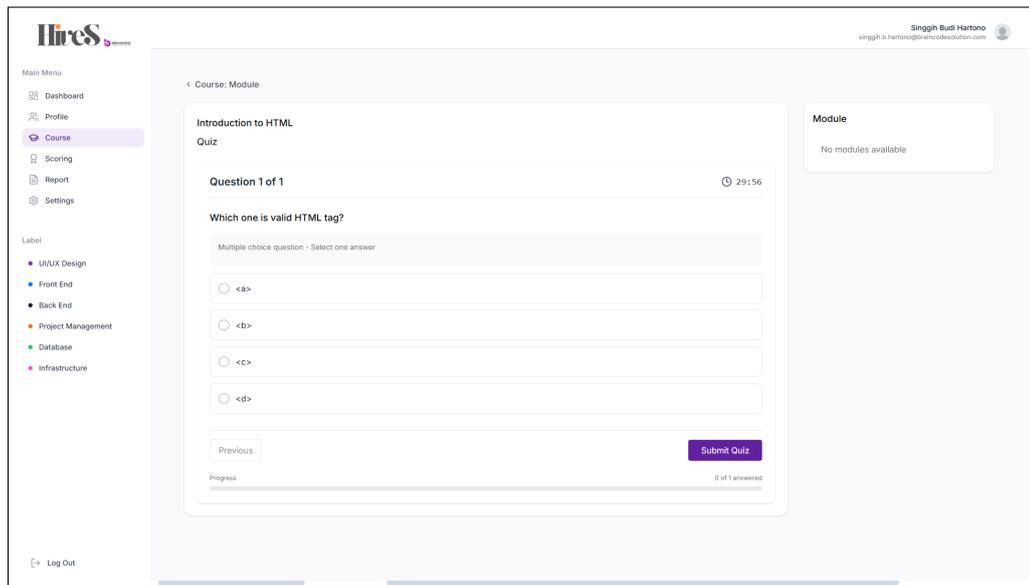
Halaman *Quiz Instruction* menampilkan aturan dan petunjuk penting sebelum memulai *quiz*, seperti durasi, larangan kecurangan, dan kewajiban menyelesaikan dalam satu sesi. Instruksi ini membantu mempersiapkan pengguna secara mental dan teknis. Setelah menekan "*Start Quiz Now*", sistem memulai *timer*, membuat *quiz session*, dan menampilkan soal. Halaman ini terintegrasi dengan API untuk memastikan kesiapan quiz.



Gambar 3.40. Halaman quiz instruction

F.9 Halaman Quiz - Multiple Choice

Halaman *Quiz - Multiple Choice* menyajikan pertanyaan dalam format pilihan ganda dengan antarmuka yang intuitif. *Timer* di bagian atas halaman menampilkan waktu tersisa secara *real-time* dan akan mengaktifkan *auto-submit* saat waktu habis. Navigasi antar pertanyaan didukung dengan tombol *Previous* dan *Next*. Sistem melacak jawaban pengguna dalam *state lokal* dan menyinkronkannya secara periodik ke server melalui API seperti `saveAnswer` dan `submitQuiz`, memastikan data tersimpan meskipun terjadi gangguan. Proses pengambilan soal dilakukan melalui `getQuizQuestions`, menjadikan pengalaman ujian ini stabil, terukur, dan bebas kecurangan.

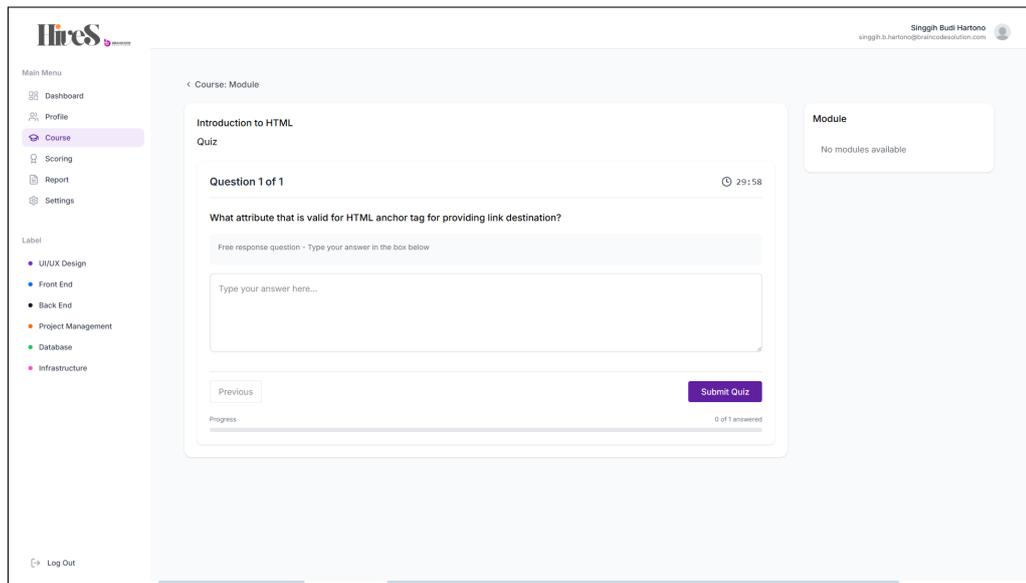


Gambar 3.41. Halaman quiz multiple choice

F.10 Halaman Quiz - Essay

Halaman *Quiz - Essay* menampilkan antarmuka soal dengan format jawaban bebas (*free response*) di mana pengguna dapat mengetikkan jawaban panjang melalui area teks yang luas dan fleksibel. Mekanisme *timer* berjalan secara real-time untuk menampilkan sisa waktu, dan sistem akan melakukan *auto-submit* saat waktu habis, sesuai aturan yang telah ditentukan. Setiap pertanyaan ditampilkan secara individual dengan indikator posisi, serta *progress bar* di bagian bawah yang menunjukkan status jawaban. Setelah pengguna mengklik *Submit Quiz*, maka data yang berisi jawaban pengguna akan di kirim ke server lewat API.

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A

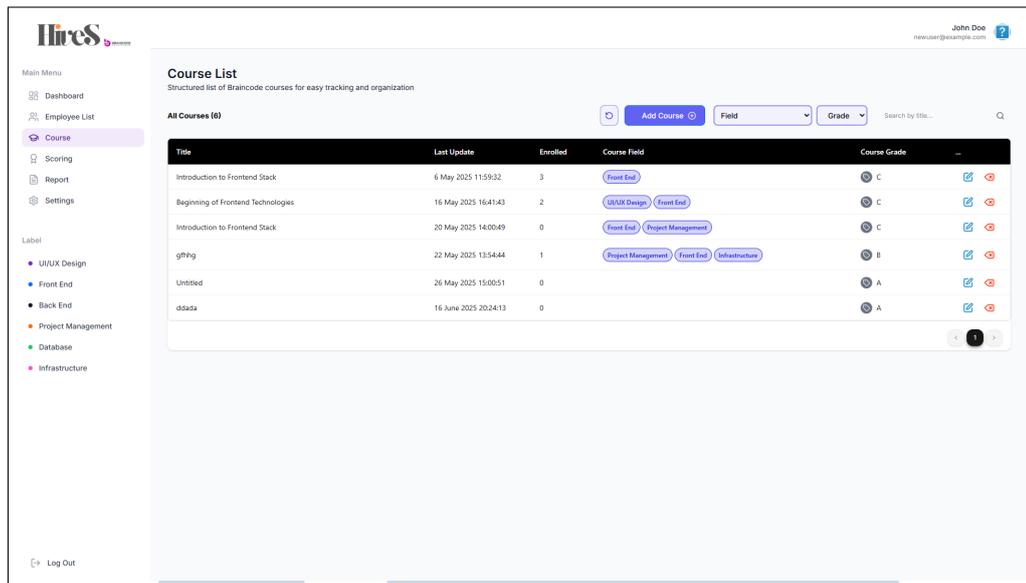


Gambar 3.42. Halaman quiz essay

G Halaman Course List (Admin)

Halaman *Course List (Admin)* menyediakan antarmuka *manajemen katalog kursus* yang komprehensif dalam format tabel, memungkinkan admin untuk *menambah, mengedit, dan menghapus kursus* dengan mudah. Dilengkapi dengan fitur *filtering* berdasarkan *field* (seperti Front End atau UI/UX Design), *grade* (A, B, C), serta *pencaharian cepat* berdasarkan judul kursus, halaman ini juga menampilkan *jumlah pendaftar (enrolled)*, *waktu update terakhir*, dan *label kategori* yang divisualisasikan dengan tag berwarna.

Proses kerja melibatkan pemanggilan data dari API saat halaman dimuat dan setiap perubahan *filter*, *search*, atau *pagination* akan memicu `fetchCourses()` untuk memperbarui tampilan tabel. *Admin* dapat membuat kursus baru melalui tombol *Add Course*, melakukan pengeditan yang diarahkan ke halaman khusus, serta menghapus kursus melalui *dialog konfirmasi*. Navigasi antar halaman kursus difasilitasi oleh kontrol *pagination*.



Gambar 3.43. Halaman course list (admin)

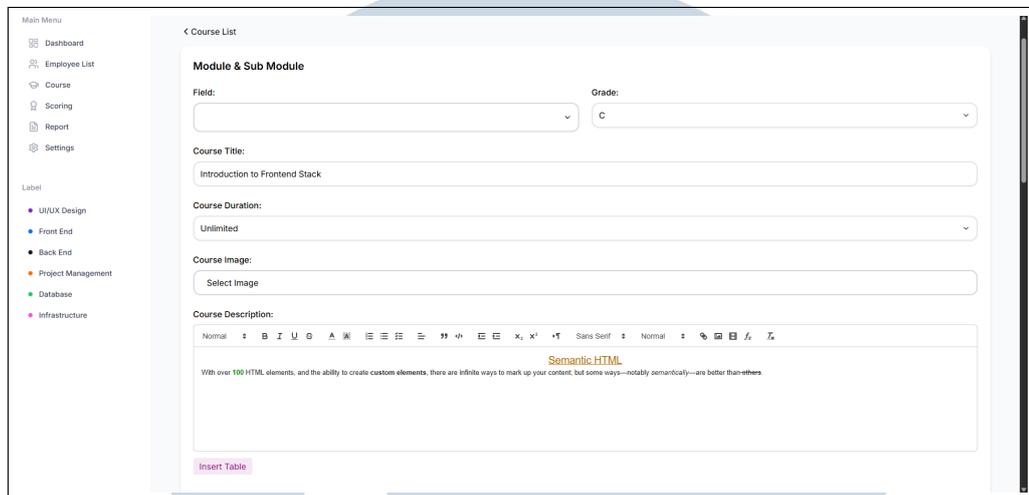
H Halaman Course Edit (Admin)

Halaman *Course Edit* pada sisi *admin* merupakan pusat pengelolaan konten kursus yang telah dibuat dalam sistem *LMS HIVES*. Di halaman ini, *admin* memiliki akses untuk mengedit berbagai aspek penting dari kursus, mulai dari metadata dasar seperti judul, deskripsi, dan durasi kursus, hingga struktur pembelajaran yang mencakup modul dan submodul dengan berbagai jenis konten. Pengeditan dilakukan secara modular, yang memudahkan *admin* untuk menavigasi antar bagian dan menerapkan perubahan secara bertahap. Setiap bagian dalam halaman ini dirancang untuk mendukung fleksibilitas dalam pengelolaan materi ajar serta memastikan pengalaman pengguna yang efisien dan intuitif.

H.1 Bagian Course Detail

Halaman *Course Edit (Bagian Course Detail)* merupakan antarmuka pengelolaan metadata kursus secara menyeluruh, yang memungkinkan admin untuk mengatur judul, deskripsi, durasi, *field*, dan *grade* dari kursus. Informasi ini disajikan dalam *form input* yang intuitif, termasuk fitur unggah gambar thumbnail dan *rich text editor* yang mendukung format teks terstruktur serta HTML semantik. Data kursus akan dimuat dari *localStorage* terlebih dahulu, dan jika tidak tersedia, akan diambil menggunakan *course_id* dari URL melalui API. Integrasi dengan API juga mencakup pengambilan daftar *grade* dan *field*, pengunggahan media, serta

pembaruan modul dan sub-modul, menjadikan halaman ini pusat kontrol penting dalam proses pembuatan dan pengelolaan kursus.

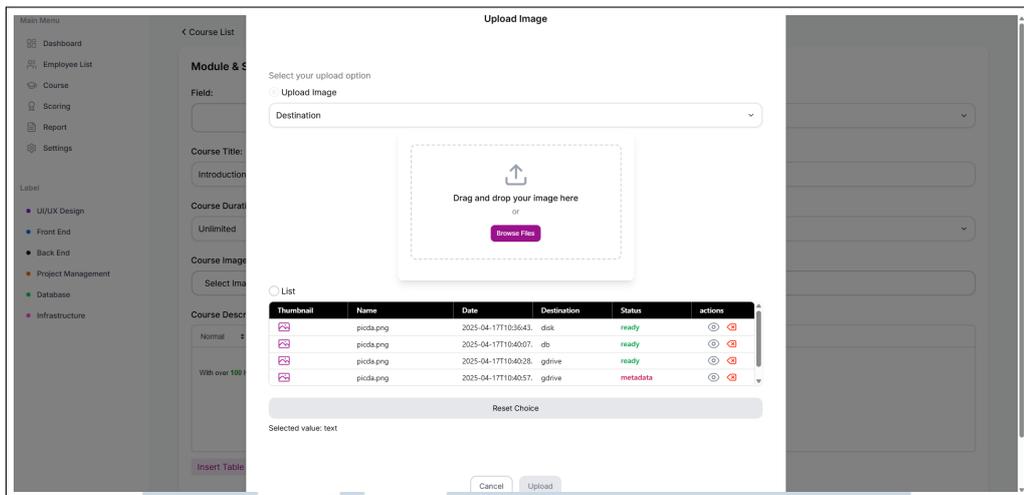
The image shows a web application interface for managing courses. On the left is a 'Main Menu' sidebar with options like Dashboard, Employee List, Course, Scoring, Report, and Settings. Below the menu are labels for UI/UX Design, Front End, Back End, Project Management, Database, and Infrastructure. The main content area is titled '< Course List' and contains a 'Module & Sub Module' form. This form has several input fields: 'Field' (a dropdown menu), 'Grade' (a dropdown menu with 'C' selected), 'Course Title' (a text input field containing 'Introduction to Frontend Stack'), 'Course Duration' (a dropdown menu with 'Unlimited' selected), 'Course Image' (a button labeled 'Select Image'), and 'Course Description' (a rich text editor with a toolbar and a sample paragraph about Semantic HTML). At the bottom of the form is an 'Insert Table' button.

Gambar 3.44. Bagian course detail

H.2 Modal Upload Image

Bagian *Modal Upload Image* merupakan bagian integral dari fitur pengelolaan kursus, yang memungkinkan admin memilih atau mengunggah gambar untuk digunakan sebagai *thumbnail* kursus. Modal ini menyediakan dua tab utama: *Upload Image* dan *List*, dengan tab unggah aktif secara default. Pengguna dapat menyeret dan menjatuhkan file gambar atau menggunakan tombol *Browse Files*, lalu memilih destinasi penyimpanan seperti *Google Drive*, *Disk Server*, atau *Database* sebelum menekan tombol *Upload*.

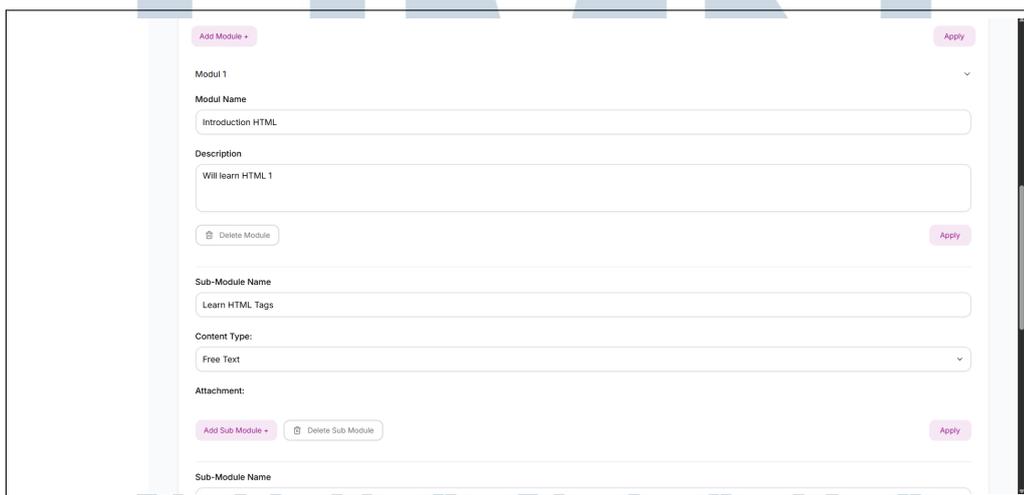
Selama proses berlangsung, *progress bar* dan status waktu nyata (*Uploading*, *Upload Complete*) akan ditampilkan. Pada tab *List*, admin dapat memilih dari gambar yang sudah tersedia berdasarkan *preview* dan *metadata*. Setelah gambar dipilih atau berhasil diunggah, modal akan tertutup secara otomatis dan gambar tersebut akan dikirim ke komponen induk untuk memperbarui *form* edit kursus. Proses ini terintegrasi langsung dengan API untuk menjamin pengelolaan file yang efisien dan terorganisir.



Gambar 3.45. Modal upload image

H.3 Bagian SubModule - FreeText

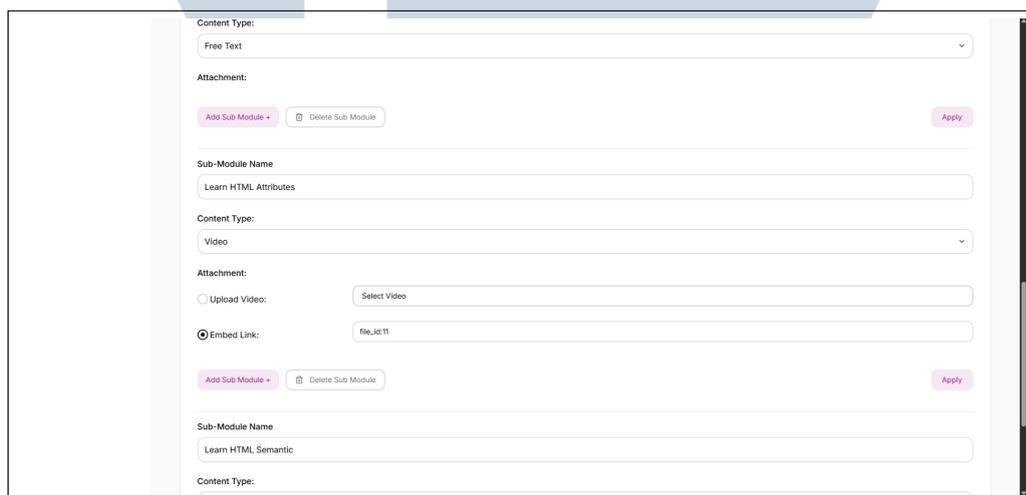
Pada bagian *SubModule - FreeText* di halaman *Course Edit* memungkinkan *admin* untuk mengelola konten pembelajaran berbasis teks. Submodul bertipe *Free Text* dipilih melalui dropdown, dan setelah konten siap, tombol “Apply” digunakan untuk menyimpan perubahan, yang selanjutnya dikirim ke API melalui fungsi `handleApplySubModule`. Proses ini juga mencakup validasi input dan umpan balik melalui *modal* untuk memastikan keandalan dan keamanan data yang disimpan.



Gambar 3.46. Bagian submodule - freetext

H.4 Bagian SubModule - Video

Pada bagian *Submodule - Video* Halaman *Course Edit* memberikan fleksibilitas tinggi bagi admin untuk menambahkan materi video melalui dua metode: *upload* langsung atau *embed link* dari platform eksternal seperti YouTube. Pemilihan metode dilakukan melalui *radio buttons*, yang secara otomatis menyesuaikan tampilan UI dan memperbarui *state* submodul. Untuk metode *upload*, pengguna diarahkan ke *modal upload video*, sementara itu, metode *embed link* memungkinkan pengguna langsung memasukkan URL video yang kemudian disimpan sebagai *subModule.content* tanpa format *prefix* khusus. Setelah konten ditentukan, tombol *Apply* digunakan untuk menyimpan data, yang kemudian dikirim ke API melalui fungsi *handleApplySubModule* untuk disimpan di server sebagai referensi video atau URL yang valid.



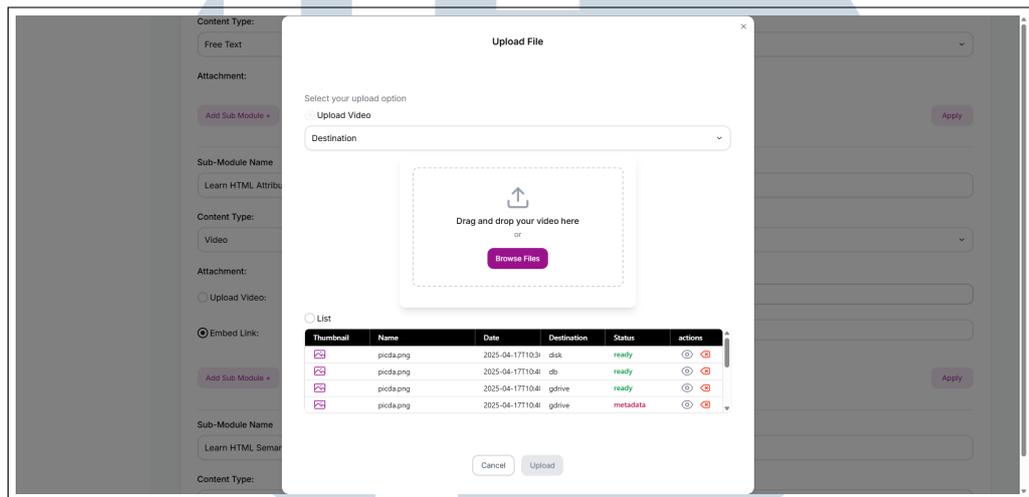
Gambar 3.47. Bagian submodule - video

H.5 Modal Upload Video

Bagian *Modal Upload Video* adalah komponen penting yang memungkinkan admin mengunggah video pembelajaran melalui dua metode utama: *drag-and-drop* atau file browser konvensional. Pengguna dapat memilih lokasi penyimpanan yang diinginkan, seperti *Google Drive*, *Disk Server*, atau *Database*, untuk mengoptimalkan manajemen media.

Proses *upload* dilengkapi dengan *preview* file, validasi format dan ukuran, serta indikator *progress bar* yang menunjukkan status unggahan secara *real-time*. Selain mengunggah file baru, pengguna juga dapat berpindah ke tab *List* untuk

memilih video yang sudah ada di sistem melalui integrasi dengan *video repository*, yang menyediakan filter, pencarian, dan metadata file. Video yang dipilih atau diunggah akan diteruskan ke komponen induk untuk dikaitkan dengan *submodule* kursus melalui API, menjadikan modal ini bagian integral dari proses pengelolaan konten video pada platform *LMS*.

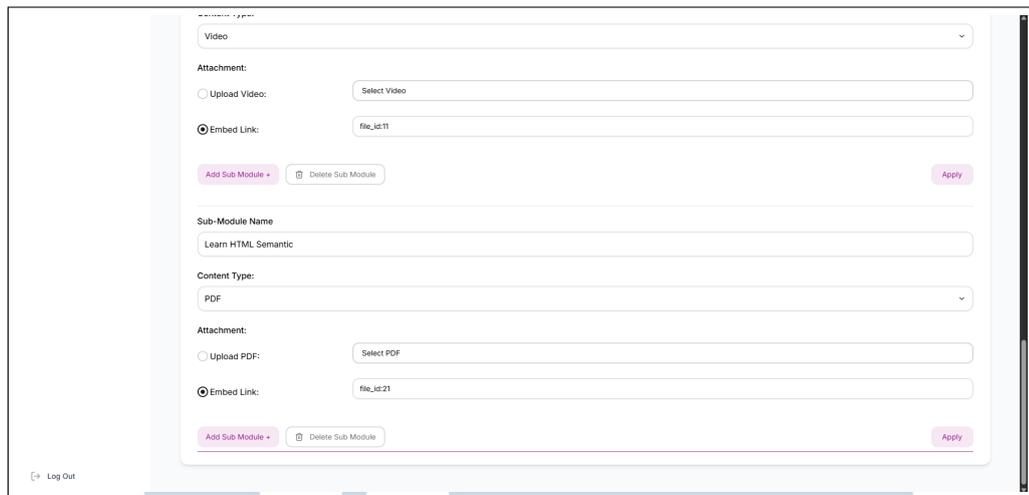


Gambar 3.48. Modal upload video

H.6 Bagian SubModule - PDF

Pada bagian *Submodule - PDF* di halaman *Course Edit* menyediakan dua metode input yaitu *upload langsung* melalui modal atau *embed link* dari sumber eksternal, yang dapat dipilih melalui *radio buttons*. Setelah file dipilih atau link dimasukkan, data akan disimpan dalam state lokal dan diteruskan ke API saat tombol *Apply* ditekan.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

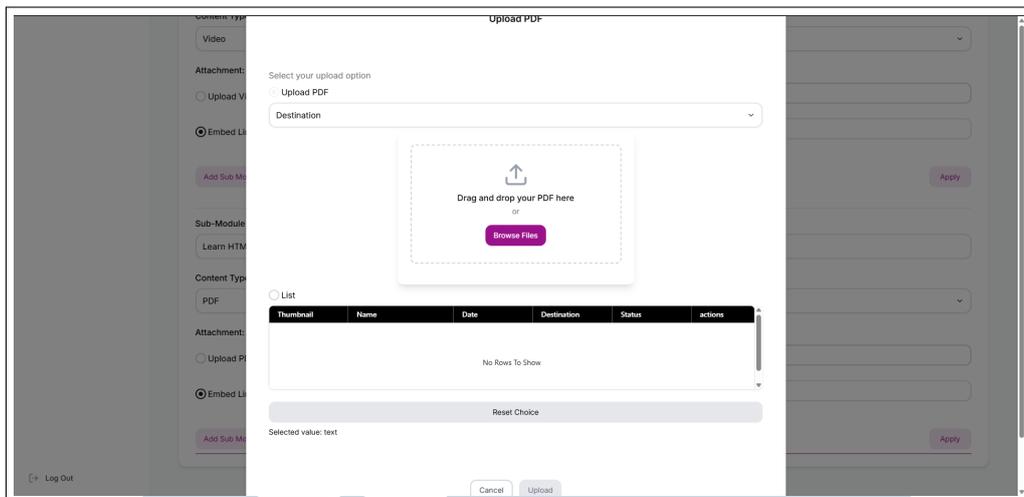


Gambar 3.49. Bagian submodule - pdf

H.7 Modal Upload PDF

Bagian *Modal Upload PDF* ini berfungsi sebagai antarmuka unggahan file yang memungkinkan admin mengunggah dokumen PDF ke sistem pembelajaran secara efisien. Terdapat dua tab utama, yaitu *Upload PDF* dan *List*, yang dapat dipilih melalui *radio button*. Pada tab *Upload PDF*, pengguna dapat mengunggah file melalui mekanisme *drag & drop* atau *file browser* tradisional, dengan pratinjau nama dan ukuran file yang ditampilkan sebelum proses unggah. Proses ini mendukung validasi agar hanya file berformat PDF yang diterima. Destinasi penyimpanan dapat dipilih antara *Google Drive*, *Disk Server*, atau *Database*, tergantung kebutuhan.

Selama proses berlangsung, pengguna akan melihat *progress bar* yang menunjukkan status unggahan secara real-time hingga muncul notifikasi *Upload Complete*. Alternatifnya, pada tab *List*, pengguna dapat memilih file PDF yang sudah pernah diunggah sebelumnya dari *repository* yang tersedia, lengkap dengan metadata dan status. File yang dipilih atau diunggah kemudian akan diteruskan ke komponen induk untuk disimpan sebagai referensi dalam submodule kursus. Proses ini diintegrasikan langsung dengan API untuk memastikan manajemen konten berjalan dengan lancar dan terstruktur.



Gambar 3.50. Modal upload pdf

3.4 Kendala dan Solusi yang Ditemukan

Pada proses pengembangan sistem *LMS HIVES* ini, berbagai tantangan muncul yang mencerminkan kompleksitas dalam membangun aplikasi yang skalabel dan terintegrasi dengan baik. Tantangan tersebut tidak hanya berasal dari sisi teknis pengembangan antarmuka dan integrasi sistem, tetapi juga dari sisi pengelolaan alur kerja dan komunikasi antar komponen dalam lingkungan pengembangan yang dinamis. Oleh karena itu, diperlukan identifikasi kendala secara menyeluruh serta strategi pemecahan masalah yang efektif agar proses pengembangan tetap berjalan optimal dan menghasilkan sistem yang stabil serta dapat digunakan dengan baik oleh pengguna.

3.4.1 Kendala

Selama proses pengembangan sistem *admin panel HIVES*, beberapa kendala signifikan muncul baik dari sisi teknis maupun non-teknis. Secara teknis, salah satu kendala utama adalah pengelolaan *state* dinamis pada fitur yang melibatkan struktur data bertingkat seperti modul dan submodul kursus. Masalah seperti *state* tidak tersinkronisasi, data yang tidak *render* secara *real-time*, atau tidak terkirim ke *API* terjadi cukup sering saat melakukan operasi tambah, ubah, dan hapus secara simultan. Selain itu, integrasi antara berbagai *modal upload* (gambar, video, PDF) dengan komponen utama seringkali menimbulkan konflik *prop drilling* dan pengelolaan nilai yang tidak konsisten, terutama pada saat pengguna memilih file

dari *repository*.

Dari sisi UI/UX, terdapat tantangan dalam menjaga konsistensi dan kejelasan antarmuka ketika banyak *modal* ditampilkan secara bersamaan. *Modal* yang saling tumpang tindih tanpa kontrol visual yang jelas membuat navigasi terasa membingungkan. Sementara itu, dalam pengelolaan data tabel (Employee List dan Course List), sinkronisasi antara *filter* dan *search*, dengan parameter *API* sempat menjadi masalah karena perubahan *state* yang tidak langsung memicu refresh data.

Kendala lain yang cukup mempengaruhi adalah pada sisi komunikasi data antara komponen *frontend* dengan *backend*, terutama ketika struktur data yang diharapkan tidak sepenuhnya konsisten, misalnya penamaan *property* atau format *response*. Hal ini menyebabkan proses *debugging* menjadi lebih kompleks dan memperlambat iterasi pengembangan.

3.4.2 Solusi

Untuk mengatasi permasalahan pengelolaan *state*, dilakukan *refactoring* menyeluruh dengan menerapkan prinsip *controlled component* dan *immutable update pattern*. Penggunaan *callback* yang terstruktur dan *state management* lokal berbasis React Hooks memudahkan pelacakan perubahan data dan meningkatkan keandalan komponen dalam menyimpan dan menampilkan informasi terbaru. Komunikasi antar komponen juga diperjelas dengan memanfaatkan *props drilling* yang lebih terkontrol serta pemisahan tanggung jawab komponen (*component decoupling*).

Permasalahan tampilan *modal* yang saling menumpuk diselesaikan dengan menggunakan komponen *modal* berbasis *portal* untuk memastikan elemen selalu dirender di luar hirarki DOM utama. Selain itu, pengelolaan *state* modal dilakukan secara terpusat untuk mencegah tumpang tindih antar modal. Untuk menghindari kesalahan saat mengambil data dari *API*, dibuat fungsi khusus yang secara otomatis menyusun parameter permintaan berdasarkan kondisi *filter* dan halaman yang sedang aktif di aplikasi.

Dari sisi integrasi *API*, dilakukan pembersihan dan standarisasi struktur *payload request* serta penyesuaian *frontend* agar selalu sesuai dengan skema data *backend*. Selain itu, *testing* manual dan simulasi *error* ditambahkan pada fase integrasi untuk mempercepat identifikasi *bug*. Dengan pendekatan iteratif dan kolaboratif, sebagian besar kendala berhasil diatasi dan menghasilkan sistem yang lebih stabil, fungsional, dan mudah dikembangkan di masa mendatang.