

BAB 3

PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi

Selama kegiatan magang di PT. Winnicode Garuda Indonesia, posisi yang dijalankan adalah sebagai *Fullstack Developer Intern* dengan penugasan utama untuk mengembangkan proyek *website* portal berita. Dalam pelaksanaannya, proses kerja dilakukan di bawah bimbingan dan pengawasan langsung dari Pak Widyantoro selaku *supervisor* perusahaan, yang memberikan arahan terkait tahapan pengembangan sistem serta evaluasi terhadap capaian setiap minggu.

Koordinasi dan komunikasi selama magang dilakukan melalui beberapa *platform* pendukung, yaitu Google Classroom yang digunakan sebagai media pengumpulan laporan dan dokumentasi perkembangan proyek secara berkala, WhatsApp yang berfungsi sebagai sarana komunikasi langsung dan diskusi teknis dengan *supervisor*, GitHub yang digunakan sebagai repositori untuk mendokumentasikan kode sumber proyek serta mendukung kolaborasi dalam pengembangan sistem secara *versioning* dan terstruktur.

Penggunaan *platform* tersebut membantu memastikan proses pengembangan proyek berjalan secara efektif, terdokumentasi dengan baik.

3.2 Tugas yang Dilakukan

Berikut merupakan beberapa tugas yang dilaksanakan selama melaksanakan kegiatan magang di Perusahaan PT. Winnicode Garuda Indonesia selama kurang lebih 5 bulan.

3.2.1 Pembuatan proposal

Pekerjaan awal yang diberikan dalam pelaksanaan program magang adalah penyusunan proposal deskripsi proyek, yang berfungsi sebagai dasar perencanaan dalam pengembangan proyek yang akan dilaksanakan selama masa magang. Tujuan utama dari pembuatan proposal ini adalah untuk merancang dan merumuskan rencana kerja secara sistematis, sehingga kegiatan pengembangan proyek dapat berjalan dengan terarah dan sesuai dengan sasaran yang ditetapkan oleh perusahaan.

Proposal tersebut berisi penjabaran mengenai seluruh aktivitas yang akan dilakukan selama periode magang di PT. Winnicode Garuda Indonesia, termasuk penjelasan mengenai latar belakang, tujuan, serta manfaat dari proyek yang dipilih, yaitu pembuatan website portal berita. Selain itu, proposal juga memuat informasi mengenai teknologi dan *framework* yang akan digunakan dalam pengembangan sistem, seperti:

- Laravel, digunakan sebagai *framework backend* untuk pengelolaan server dan pembuatan API.
- MySQL, digunakan sebagai sistem manajemen basis data yang menyimpan informasi terkait berita, pengguna, dan komentar.
- blade, digunakan sebagai *framework frontend* untuk membangun antarmuka pengguna yang dinamis dan responsif.

Di samping penjelasan teknis, proposal deskripsi proyek juga dilengkapi dengan *timeline* perencanaan mingguan, yang mencantumkan rincian kegiatan dan target pekerjaan selama masa magang berlangsung, yaitu selama lima bulan. Dokumen ini menjadi acuan utama dalam pelaksanaan proyek serta mempermudah proses pemantauan dan evaluasi capaian kerja selama program magang.

3.2.2 Pembuatan Design rancangan Website

Tugas kedua yang dilaksanakan selama masa magang di PT. Winnicode Garuda Indonesia adalah penyusunan rancangan *design website* yang akan dikembangkan sepanjang periode magang. Kegiatan ini bertujuan untuk menghasilkan acuan visual dan struktural sebagai dasar dalam proses implementasi sistem pada tahap selanjutnya.

Perancangan dilakukan dengan menyusun desain antarmuka (*user interface*) yang mencerminkan struktur halaman, navigasi, serta elemen-elemen interaktif yang dibutuhkan dalam *website* portal berita. Desain ini juga mempertimbangkan aspek pengalaman pengguna (*user experience*) guna memastikan kemudahan dalam penggunaan dan aksesibilitas bagi pengguna akhir.

Proses desain dilaksanakan menggunakan perangkat lunak desain antarmuka seperti Figma, yang memungkinkan pembuatan prototipe secara interaktif dan kolaboratif. Hasil dari perancangan ini menjadi patokan awal dalam implementasi *frontend* serta berperan penting dalam menjaga konsistensi tampilan dan fungsi sistem selama proses pengembangan berlangsung.

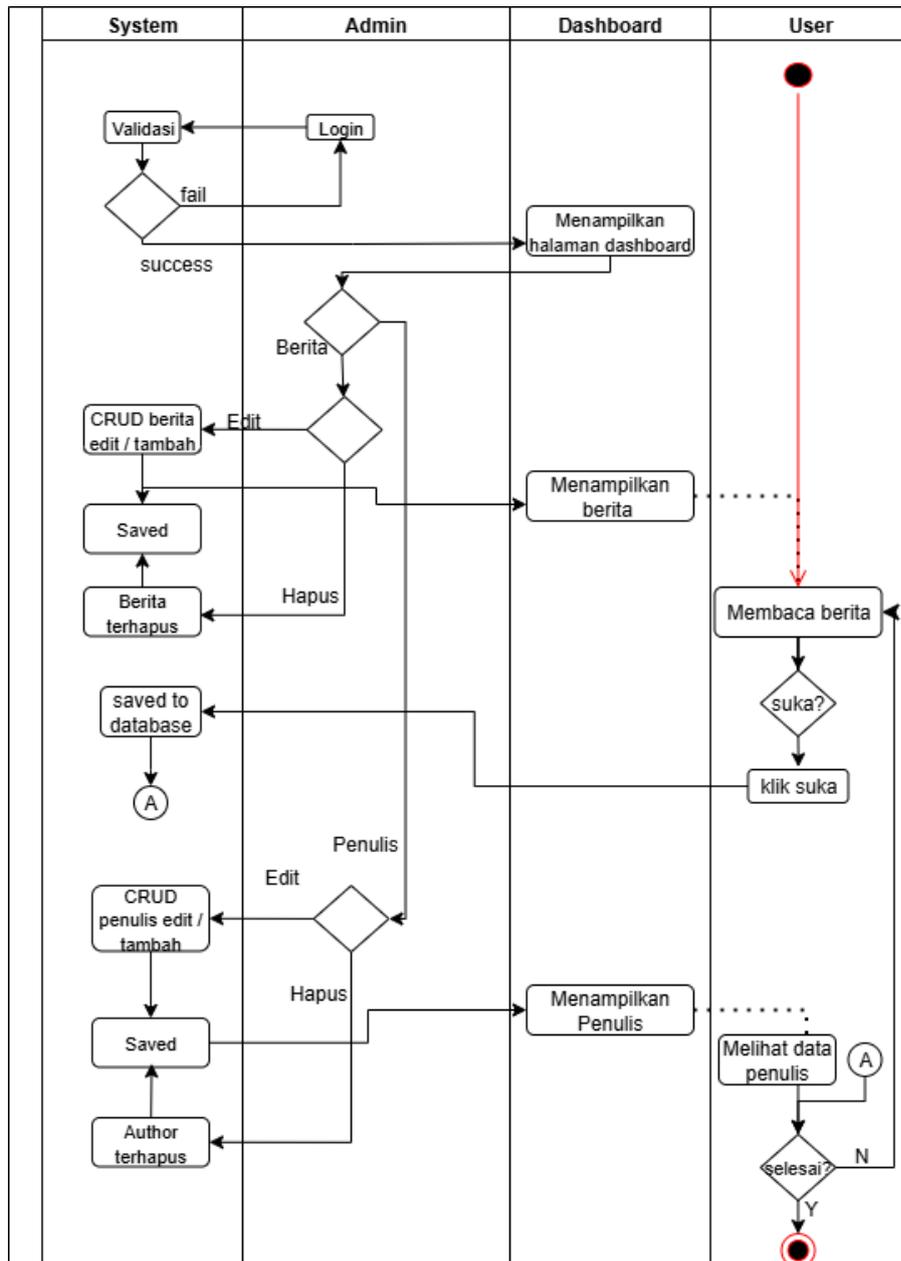
menggambarkan secara visual bagaimana peran masing-masing aktor dalam menjalankan fungsionalitas yang ada di dalam sistem.

Terdapat dua aktor utama yang terlibat dalam sistem, yaitu *User* dan *Admin*. Aktor *User* memiliki peran sebagai pengunjung atau pembaca berita, yang dapat mengakses berbagai informasi seperti melihat daftar berita terbaru, mencari berita berdasarkan kategori, serta membaca detail dari setiap berita. Di sisi lain, aktor *Admin* berfungsi sebagai pengelola sistem yang bertugas melakukan manajemen konten, seperti menambahkan berita baru, mengedit informasi berita yang sudah ada, menghapus berita yang tidak relevan, serta mengatur kategori dan penulis.

Setiap interaksi antara aktor dan sistem direpresentasikan dalam bentuk use case yang menunjukkan apa saja yang bisa dilakukan oleh masing-masing peran. Use case diagram ini juga membantu dalam proses identifikasi kebutuhan sistem, sehingga pengembangan fitur dapat dilakukan secara terstruktur dan efisien. [5]

- Activity Diagram





Gambar 3.2. Activity Diagram Admin dan User

Activity Diagram merupakan jenis pemodelan perilaku dalam Unified Modeling Language (UML) yang digunakan untuk menggambarkan alur kegiatan atau proses bisnis yang terjadi dalam suatu sistem. Diagram ini berperan dalam menunjukkan runtutan aktivitas yang dilakukan oleh aktor ketika berinteraksi dengan sistem, sekaligus memperlihatkan bagaimana sistem merespons setiap tindakan yang dilakukan oleh pengguna.

Gambar 3.2 Pada sistem informasi portal berita ini, aktor (Admin) memiliki sejumlah aktivitas penting yang berkaitan langsung dengan pengelolaan konten dan

data di dalam sistem. Aktivitas dimulai ketika admin melakukan proses login ke dalam sistem. Setelah berhasil masuk, admin akan diarahkan ke halaman dashboard, yang menjadi pusat kontrol dari seluruh fitur administrasi yang tersedia.

Dari dashboard tersebut, admin dapat memilih untuk melakukan beberapa tindakan, seperti menambahkan berita baru, memperbarui berita yang sudah ada, atau menghapus berita yang tidak lagi relevan. Selain itu, admin juga memiliki akses untuk mengelola kategori berita, termasuk membuat kategori baru atau mengedit kategori yang sudah tersedia. Pengelolaan data penulis juga menjadi bagian dari aktivitas admin, yang mencakup penambahan informasi *author* baru atau pembaruan data *author* yang telah ada.

Diagram aktivitas ini turut menggambarkan proses pengambilan keputusan dalam alur kerja admin. Sebagai contoh, apabila terdapat kekurangan data atau kesalahan saat mengisi informasi berita, maka sistem akan menampilkan pesan kesalahan dan mengarahkan pengguna kembali ke tahap pengisian data untuk diperbaiki. Sebaliknya, jika semua data valid, sistem akan menyimpan informasi ke dalam basis data dan menampilkan konfirmasi keberhasilan.

Sementara itu terdapat pada gambar 3.2, *User* sebagai aktor eksternal juga memiliki aktivitas tersendiri dalam menggunakan sistem. Activity Diagram untuk *user* menggambarkan serangkaian langkah yang dilakukan oleh pengguna umum saat mengakses dan berinteraksi dengan portal berita.

Aktivitas diawali saat user membuka halaman utama dari sistem. Pada tahap ini, sistem akan menampilkan berbagai informasi seperti daftar berita terbaru, *banner* berita unggulan, serta kategori berita yang tersedia. *User* dapat memilih untuk menjelajahi daftar berita berdasarkan minat atau kata kunci tertentu, serta mengklik salah satu judul berita untuk melihat informasi secara lebih detail.

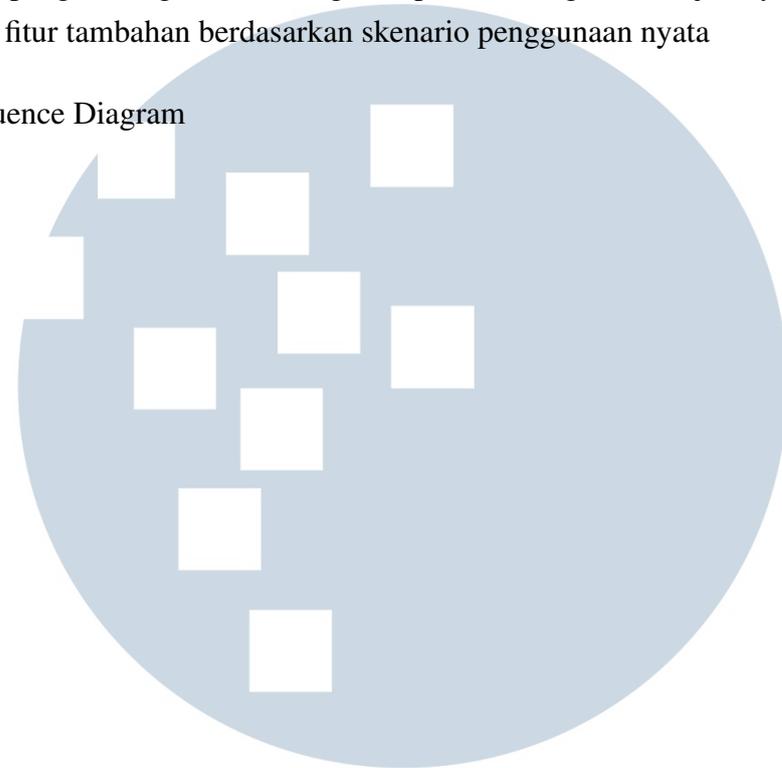
Ketika user memilih berita, sistem akan menampilkan isi berita lengkap, termasuk judul, gambar, isi artikel, kategori, dan informasi penulis. Jika user ingin membaca berita lainnya, ia dapat kembali ke daftar berita atau memilih kategori berbeda. Di samping itu, *user* juga dapat mengakses bagian “Kenali Author” untuk melihat profil penulis dan daftar berita yang ditulis oleh masing-masing penulis.

Activity diagram ini memperlihatkan alur interaksi pengguna dengan sistem secara sederhana namun terstruktur, dimulai dari awal kunjungan hingga selesai membaca berita. Alur ini penting untuk dipahami karena berhubungan langsung dengan pengalaman pengguna (*user experience*) yang ingin ditingkatkan dalam pengembangan sistem.

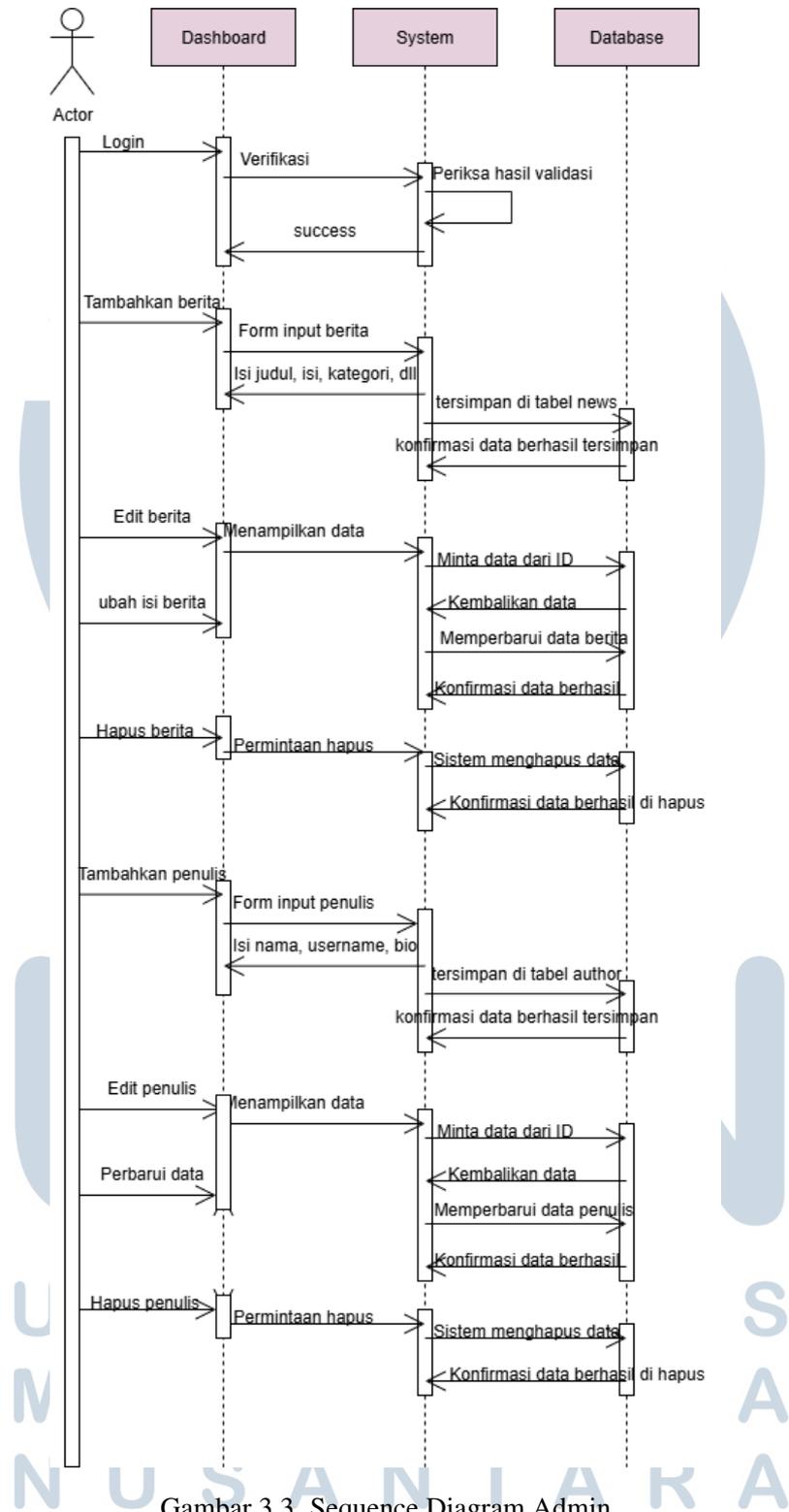
Dengan adanya pemodelan aktivitas ini, pengembangan sistem dapat lebih

terarah karena setiap proses sudah tergambar secara sistematis. Hal ini juga membantu pengembang dalam mengantisipasi kemungkinan terjadinya error atau kebutuhan fitur tambahan berdasarkan skenario penggunaan nyata

- Sequence Diagram



UMMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.3. Sequence Diagram Admin

Gambar 3.3 Sequence Diagram digunakan untuk memodelkan alur komunikasi antar objek dalam suatu skenario penggunaan sistem. Diagram ini menampilkan urutan pesan yang dikirim antar komponen sistem dalam kurun waktu tertentu, serta

menunjukkan bagaimana proses tersebut berjalan dari awal hingga akhir.

Pada skenario mengedit berita, proses diawali ketika admin sebagai aktor utama mengakses halaman daftar berita. Setelah memilih salah satu berita yang ingin diperbarui, admin akan diarahkan menuju halaman formulir edit. Sistem kemudian mengambil data berita yang bersangkutan dari database dan menampilkannya ke dalam formulir yang dapat diubah.

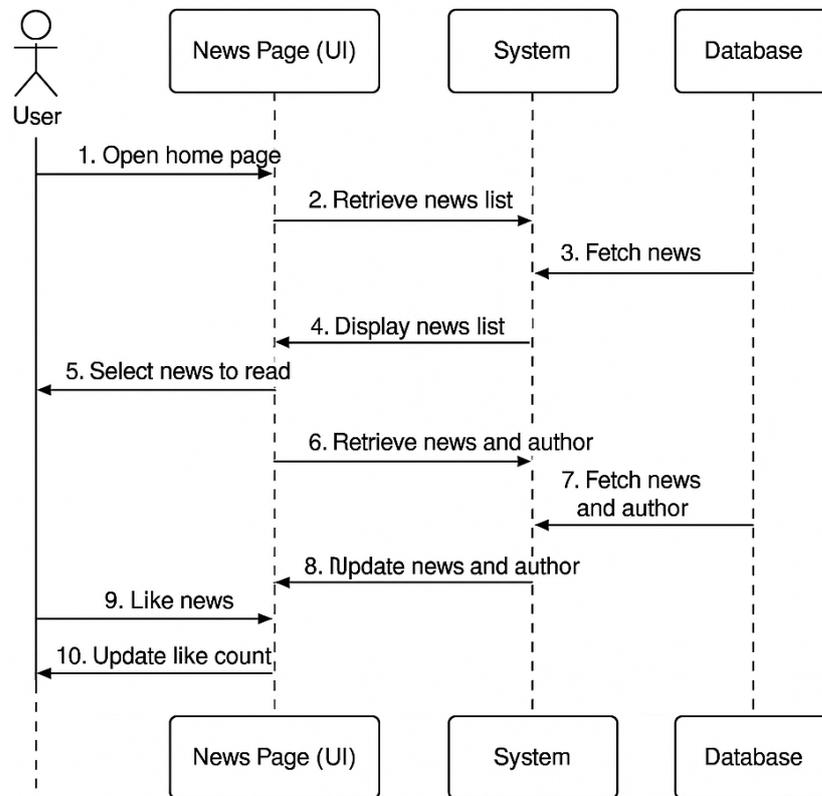
Selanjutnya, admin melakukan perubahan pada elemen berita seperti judul, isi berita, kategori, ataupun *thumbnail*. Setelah proses pengeditan selesai, admin menekan tombol simpan. Sistem akan memvalidasi data yang diinputkan. Jika data valid, sistem akan mengirimkan permintaan pembaruan ke *database* untuk menyimpan perubahan tersebut. Jika proses berhasil, sistem akan menampilkan notifikasi keberhasilan dan mengarahkan admin kembali ke halaman daftar berita.

Diagram ini mencerminkan bagaimana interaksi antar komponen seperti admin, halaman edit, *controller*, dan *database* berjalan secara berurutan dan sistematis. Penyajian alur ini bertujuan untuk memastikan bahwa setiap langkah dalam pengeditan data mengikuti prosedur yang benar, serta memfasilitasi pengecekan bila ditemukan ketidaksesuaian atau kesalahan.

Penghapusan berita merupakan salah satu fungsi dalam modul administrasi yang memungkinkan pihak admin untuk melakukan kurasi terhadap konten, khususnya yang tidak relevan lagi atau bertentangan dengan standar redaksional. Sequence Diagram untuk proses ini menggambarkan interaksi yang terjadi saat admin menghapus sebuah berita dari sistem.

Proses dimulai ketika admin membuka halaman daftar berita. Pada masing-masing entri berita, tersedia tombol atau opsi hapus yang dapat digunakan untuk menghapus data tersebut. Setelah admin memilih opsi tersebut, sistem akan memunculkan dialog konfirmasi untuk memastikan bahwa tindakan ini dilakukan secara sadar dan tidak terjadi penghapusan secara tidak sengaja.

Apabila admin mengonfirmasi penghapusan, sistem akan mengirimkan permintaan ke *controller* untuk memproses permintaan tersebut. *Controller* akan meneruskan instruksi kepada model untuk menghapus data dari database berdasarkan ID berita yang dipilih. Setelah data berhasil dihapus, sistem akan memberikan umpan balik berupa pesan notifikasi bahwa berita telah dihapus dari sistem.



Gambar 3.4. Sequence Diagram User

Gambar 3.4 Sequence diagram ini menjelaskan alur interaksi pengguna (*User*) dengan sistem portal berita, mulai dari memilih berita hingga menyukai konten. Diagram ini mencerminkan fungsi-fungsi utama yang dapat diakses oleh pengguna umum tanpa perlu login, seperti membaca berita dan melihat informasi penulis.

Proses dimulai ketika pengguna membuka halaman utama portal berita. Sistem menampilkan daftar berita yang diambil dari *database*. Selanjutnya, pengguna memilih salah satu berita untuk dibaca secara lengkap. Saat berita dibuka, sistem akan menampilkan detail isi berita beserta informasi penulisnya.

Jika tersedia, pengguna juga dapat memberikan respons positif terhadap artikel dengan menekan tombol “Suka” (*Like*). Saat tombol ini diklik, sistem akan memperbarui jumlah like pada *database*. Semua proses ini menggambarkan bagaimana pengguna dapat menikmati dan berinteraksi dengan konten tanpa perlu hak akses khusus seperti admin.

Diagram ini tidak hanya memvisualisasikan urutan proses yang terjadi,

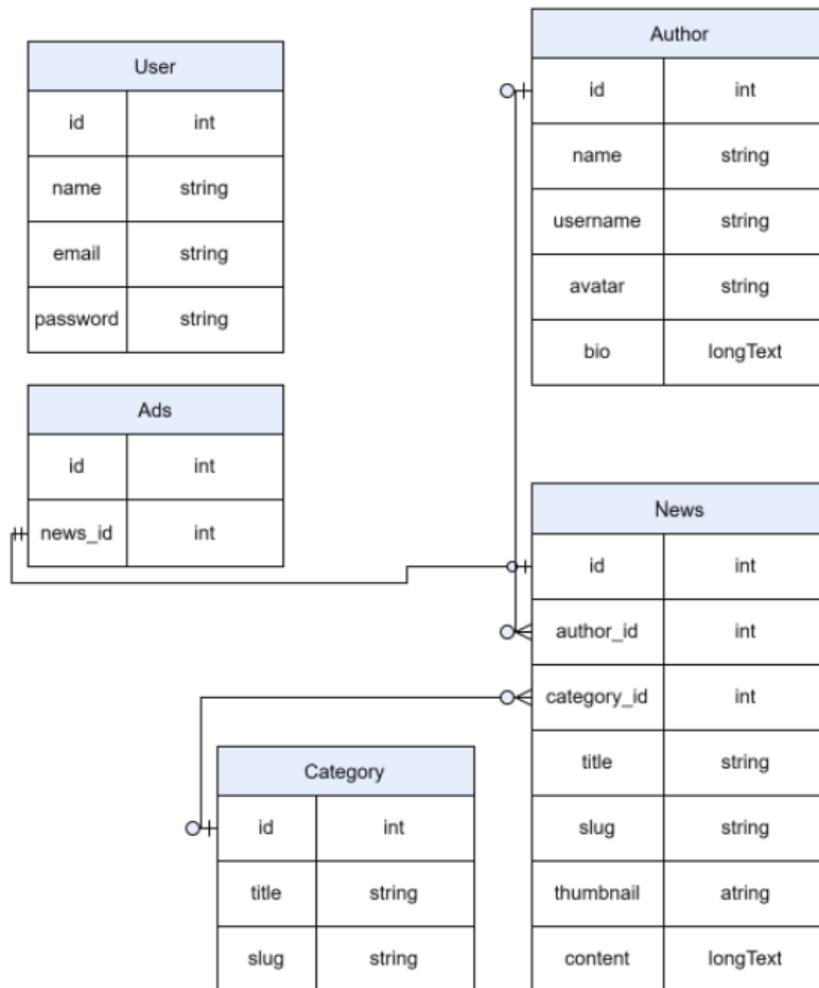
tetapi juga menekankan pentingnya tahapan verifikasi sebelum tindakan destruktif dilakukan. Upaya tersebut menjadi bagian dari implementasi standar keamanan dan keandalan dalam proses manajemen konten digital melalui *platform website*.

3.2.4 Struktur Database Website Portal Berita

Setelah tahap perancangan desain antarmuka, kegiatan selanjutnya yang dilakukan adalah pembuatan struktur basis data (*database*). Tahap ini bertujuan untuk merancang dan menyusun hubungan antar data yang akan digunakan dalam sistem, serta menentukan alur dan kebutuhan data secara keseluruhan. Struktur basis data yang dirancang menjadi fondasi penting dalam proses pengembangan *backend* sistem, karena menentukan bagaimana data disimpan, diakses, dan dikelola.

Dalam rancangan tersebut, terdapat beberapa entitas utama yang dibutuhkan dalam pengelolaan *website* portal berita, antara lain:

1. *User*: Menyimpan data pengguna sistem, baik sebagai pembaca maupun administrator.
2. *News*: Menyimpan data berita yang diterbitkan, mencakup informasi seperti judul, isi berita, waktu publikasi, serta relasi dengan entitas lain seperti penulis (*Author*) dan kategori (*Category*).
3. *Category*: Menyimpan data kategori berita yang digunakan untuk mengelompokkan konten berdasarkan topik tertentu.
4. *Author*: Menyimpan informasi terkait penulis berita yang terlibat dalam pembuatan konten.
5. *Ads*: Menyimpan data mengenai iklan yang ditampilkan pada halaman *website*, termasuk konten dan penempatan iklan.



Gambar 3.5. Struktur *Database* Portal Berita

Pada gambar 3.5 berikut menjelaskan setiap entitas memiliki relasi yang dirancang sesuai dengan kebutuhan sistem, misalnya entitas *News* memiliki relasi dengan *Author* sebagai penulis dan *Category* sebagai pengelompokan berita. Perancangan ini divisualisasikan dalam bentuk *Entity Relationship Diagram* (ERD) sebagai panduan dalam implementasi basis data menggunakan MySQL.

3.2.5 CRUD admin menggunakan filament di Laravel

Pekerjaan selanjutnya adalah menerapkan Filament, yaitu salah satu paket atau fitur yang tersedia dalam ekosistem Laravel, yang berfungsi sebagai antarmuka administratif untuk mempermudah proses pengelolaan data [2] [3]. Penerapan Filament dilakukan guna membangun modul CRUD (*Create, Read, Update, Delete*) yang diperuntukkan bagi pengguna dengan peran administrator [6][7].

Melalui pemanfaatan Filament, proses pengelolaan data seperti berita, kategori, pengguna, dan iklan dapat dilakukan secara lebih efisien melalui *dashboard* yang responsif dan mudah digunakan. Selain itu, Filament menyediakan berbagai komponen siap pakai yang memungkinkan pengembangan antarmuka admin dilakukan dengan lebih cepat, namun tetap menjaga aspek keamanan dan skalabilitas sistem [8].

Dalam penerapannya, Filament digunakan untuk membangun *dashboard* administrasi yang terdiri atas beberapa modul pengelolaan data, antara lain:

- Manajemen Data Berita (*News*)

Modul ini memungkinkan administrator untuk menambahkan, mengubah, menghapus, serta menampilkan daftar berita melalui antarmuka CRUD yang disediakan oleh Filament. Formulir input dilengkapi dengan validasi otomatis dan editor konten untuk mendukung penulisan artikel secara lebih efisien dan terstruktur [9][10].

Berikut merupakan contoh kode pada *migration news*:

```
1     public function up(): void
2     {
3         Schema::create('news', function (Blueprint $table) {
4             $table->id();
5             $table->foreignId('author_id');
6             $table->foreignId('news_category_id');
7             $table->string('title');
8             $table->string('slug')->unique();
9             $table->string('thumbnail');
10            $table->longText('content');
11            $table->timestamps();
12        });
13    }
14
```

Kode 3.1: Kode Migration News

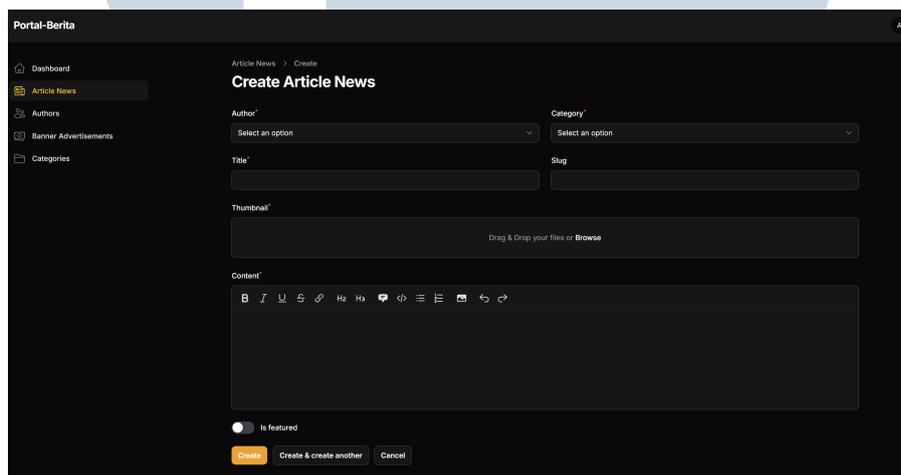
Kode migrasi 3.1 berikut digunakan untuk membuat tabel *news* dalam *database*. Tabel ini dirancang untuk menyimpan data berita yang akan ditampilkan pada sistem portal berita.

Secara garis besar, tabel ini memiliki beberapa kolom utama, yaitu:

- *id*: *primary key* sebagai identitas unik setiap berita.

- `author_id` dan `news_category_id` : relasi ke tabel penulis dan kategori berita.
- `title`, `slug`, dan `thumbnail` : menyimpan judul berita, slug URL, dan gambar thumbnail.
- `content` : menyimpan isi lengkap dari berita.
- `timestamps` : mencatat waktu pembuatan dan pembaruan data secara otomatis.

Struktur ini memungkinkan pengelolaan berita secara dinamis, terhubung dengan entitas penulis dan kategori, serta mendukung sistem manajemen konten berbasis Laravel.



Gambar 3.6. Contoh Tampilan *Dashboard Article News* Menggunakan Filament

Gambar 3.6 di atas menunjukkan tampilan antarmuka *Dashboard Article News* yang dibangun menggunakan Filament sebagai admin panel. Tampilan ini menampilkan daftar artikel berita yang terstruktur dengan baik, mencakup informasi penting seperti judul, kategori, tanggal publikasi, dan status penayangan. Dengan fitur manajemen konten berbasis tabel dan form interaktif, Filament memungkinkan pengelolaan data artikel secara efisien dan *real-time* oleh admin. Antarmuka yang sederhana namun fungsional ini mendukung proses editorial dalam sistem portal berita berbasis Laravel.

- Manajemen Kategori Berita (*Category*)

Digunakan untuk mengelola pengelompokan berita berdasarkan topik tertentu. Administrator dapat menambahkan maupun memperbarui nama serta deskripsi kategori sesuai dengan kebutuhan editorial.

```
1 public function up(): void
2 {
3     Schema::create('news_categories', function (Blueprint
4     $table) {
5         $table->id();
6         $table->string('title');
7         $table->string('slug')->unique();
8         $table->timestamps();
9     });
10 }
```

Kode 3.2: Kode Migration Kategori

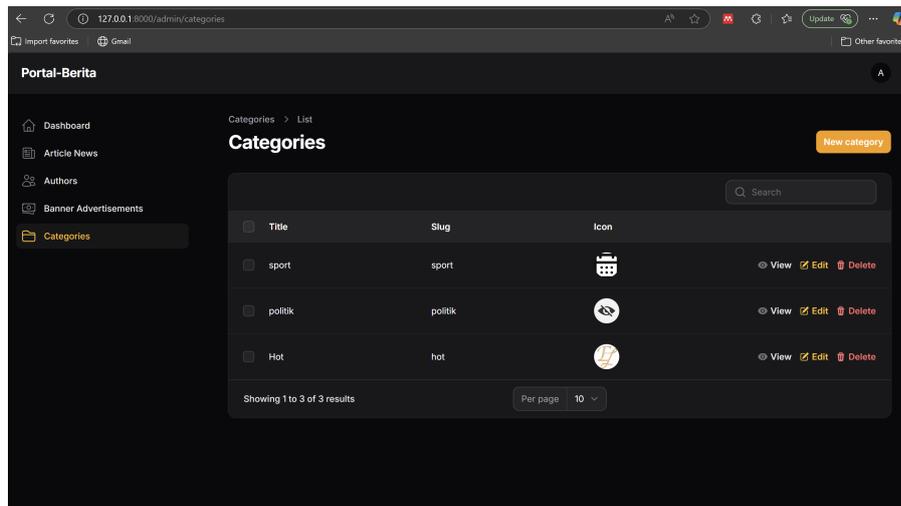
Kode migrasi 3.2 di atas digunakan untuk membuat tabel news yang menyimpan data berita. Secara umum, tabel ini mencakup:

- Identitas berita (id, title, slug, dan thumbnail).
- Isi lengkap berita dalam bentuk teks panjang.
- Relasi ke tabel penulis (author_id) dan kategori (news_category_id).
- Informasi waktu pembuatan dan pembaruan data melalui timestamps.

Struktur ini digunakan sebagai dasar pengelolaan data berita dalam sistem informasi portal berita berbasis Laravel.

Contoh tampilan antarmuka *dashboard* kategori dapat dilihat pada gambar berikut:

UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.7. Contoh Tampilan *Dashboard* Kategori Menggunakan Filament

Gambar 3.7 di atas menampilkan tampilan *Dashboard* Kategori yang dikembangkan menggunakan Filament sebagai bagian dari sistem admin portal berita. Melalui tampilan ini, admin dapat melakukan manajemen kategori berita secara efisien, seperti menambah, mengedit, atau menghapus data kategori. Antarmuka yang responsif dan terstruktur memungkinkan proses pengelompokan artikel berdasarkan kategori menjadi lebih mudah, serta memastikan konten tersusun dengan rapi sesuai klasifikasi topik yang ditentukan.

- Manajemen Penulis (*Author*)

Modul ini digunakan untuk menyimpan dan mengelola informasi penulis berita. Setiap data penulis memiliki relasi langsung dengan entitas berita yang diterbitkan, sehingga memudahkan dalam pelacakan dan pengelompokan konten berdasarkan kontributor [8].

```

1     public function up(): void
2     {
3         Schema::create('authors', function (Blueprint $table) {
4             $table->id();
5             $table->string('name');
6             $table->string('username')->unique();
7             $table->string('avatar');
8             $table->longText('bio');
9             $table->timestamps();
10        });
11    }

```

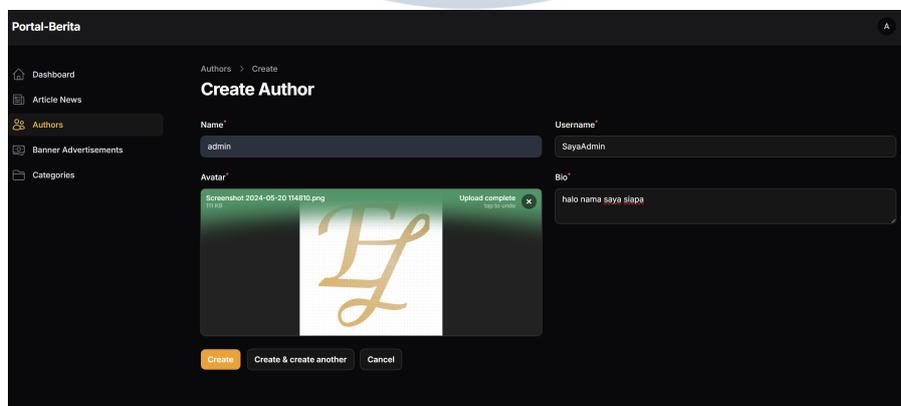
Kode 3.3: Kode Migration Author

Kode migrasi 3.3 berikut digunakan untuk membuat tabel `authors`, yaitu tabel yang menyimpan data penulis berita dalam sistem. Tabel ini terdiri atas:

- `id` : sebagai *primary key* untuk identitas unik penulis.
- `name` dan `username` : nama lengkap dan username unik penulis.
- `avatar` : menyimpan foto profil penulis.
- `bio` : deskripsi atau biodata penulis dalam bentuk teks panjang.
- `timestamps` : untuk mencatat waktu pembuatan dan pembaruan data.

Tabel ini terhubung secara relasional dengan tabel `news` dalam sistem, di mana setiap berita memiliki satu penulis.

Contoh tampilan antarmuka *dashboard author* atau penulis dapat dilihat pada gambar berikut:



Gambar 3.8. Contoh Tampilan *Dashboard Author* Menggunakan Filament

Gambar 3.8 di atas memperlihatkan tampilan *Dashboard Author* pada panel admin yang dibangun menggunakan Filament. Dashboard ini berfungsi untuk mengelola data penulis atau kontributor artikel dalam sistem portal berita. Informasi yang ditampilkan mencakup nama penulis, email, peran, serta status keaktifan. Melalui fitur CRUD yang tersedia, admin dapat dengan mudah menambahkan, memperbarui, maupun menghapus data penulis, sehingga pengelolaan konten menjadi lebih terstruktur dan terorganisasi.

- Manajemen Pengguna (*User*)

Memfasilitasi pengelolaan data pengguna sistem, baik dari sisi pembaca umum maupun pengguna internal dengan hak akses tertentu. Modul ini juga mendukung pemantauan aktivitas dan peran dari masing-masing pengguna [11].

```
1 Schema::create('users', function (Blueprint $table) {
2     $table->id();
3     $table->string('name');
4     $table->string('email')->unique();
5     $table->timestamp('email_verified_at')->nullable();
6     $table->string('password');
7     $table->rememberToken();
8     $table->timestamps();
9 });
10
```

Kode 3.4: Kode Migration User

Kode 3.4 tersebut digunakan untuk membuat tabel `users` yang merupakan bagian dari sistem autentikasi Laravel. Tabel ini menyimpan data pengguna seperti nama, email, dan password.

Secara umum, kolom-kolom yang terdapat pada tabel ini meliputi:

- `id` : identitas unik pengguna (sebagai *primary key*).
- `name` dan `email` : menyimpan nama dan alamat email pengguna. Kolom `email` bersifat unik.
- `email_verified_at` : menyimpan waktu ketika email pengguna terverifikasi. Kolom ini dapat bernilai kosong.
- `password` : menyimpan password yang telah dienkripsi.
- `rememberToken` : digunakan untuk fitur "ingat saya" saat login.
- `timestamps` : mencatat waktu pembuatan dan pembaruan data pengguna.

Tabel ini secara *default* dibuat oleh Laravel untuk mendukung fitur login dan manajemen pengguna.

- Manajemen Iklan (*Banner*)

Digunakan untuk mengatur konten iklan yang akan ditampilkan pada halaman website. Fitur ini mencakup pengunggahan materi iklan, pengaturan jadwal tayang, serta pengelompokan berdasarkan lokasi atau halaman penempatan iklan [7].

```
1     public function up(): void
2     {
3         Schema::create('banners', function (Blueprint $table) {
4             $table->id();
5             $table->foreignId('news_id');
6             $table->timestamps();
7         });
8     }
9
```

Kode 3.5: Kode Migration Banner

Kode 3.5 memuat tampilan yang digunakan untuk mengelola konten iklan (*banner*) yang akan ditayangkan pada halaman website portal berita. Sistem ini memungkinkan admin untuk menyesuaikan iklan secara dinamis, mendukung strategi promosi dan monetisasi konten secara efektif [7].

Tabel *banners* dibuat untuk menyimpan data *banner* yang akan ditampilkan pada halaman utama portal berita. Setiap *banner* merepresentasikan satu berita yang ditandai melalui relasi ke tabel *news*.

Kolom-kolom utama dalam tabel ini adalah:

- *id*: identitas unik banner sebagai *primary key*.
- *news_id*: kolom *foreign key* yang merujuk ke berita yang ditampilkan sebagai *banner*.
- *timestamps*: mencatat waktu pembuatan dan pembaruan data.

Dengan struktur ini, sistem dapat mengelola daftar berita unggulan atau berita pilihan yang ditampilkan dalam bentuk banner di antarmuka pengguna.

Setiap modul dikembangkan menggunakan Filament *Resources*, yang mempermudah proses konfigurasi tabel data, formulir input, serta aksi-aksi khusus seperti filter, pencarian, dan ekspor data [12].

Tampilan antarmuka Filament bersifat modern, responsif, dan terintegrasi secara otomatis dengan sistem autentikasi Laravel, sehingga hanya pengguna yang memiliki hak akses tertentu yang dapat mengakses halaman *dashboard* admin [13].

3.2.6 Pembuatan Frontend antarmuka

Pembuatan antarmuka pengguna (*frontend*) dilakukan menggunakan *Blade Template Engine* bawaan dari Laravel. Pada halaman utama, antarmuka dirancang agar dapat menampilkan beberapa bagian penting seperti *banner* berita, berita unggulan, berita terbaru, daftar penulis, serta pilihan berita dari para penulis. Seluruh data ini ditampilkan secara dinamis berdasarkan hasil *query* dari *controller* dan diproses di *view* menggunakan *Blade syntax*. Berikut merupakan potongan kode Blade yang digunakan pada halaman utama (*home page*):

```
1 @foreach ($banners as $banner)
2   <a href="{{ route('news.show', $banner->news->slug) }}" class="
   block">
3     <div class="relative bg-cover" style="background-image: url
   ('{{ asset('storage/' . $banner->news->thumbnail) }}')">
4       <div class="absolute inset-0 bg-gradient-to-t from-black/40
   to-transparent"></div>
5       <div class="relative z-10 text-white px-4">
6         <div class="bg-primary text-xs rounded px-3 py-1">
7           {{ $banner->news->newsCategory->title }}
8         </div>
9         <p class="text-2xl font-semibold mt-1">{{ $banner->news->
   title }}</p>
10        <div class="flex items-center gap-1 mt-1">
11          
12          <p class="text-xs">{{ $banner->news->author->name }}</p>
13          ...
14 @endforeach
```

Kode 3.6: Kode Blade untuk Banner Berita

Kode 3.6 Pada bagian '@foreach (\$banners as \$banner)', data banner ditampilkan dalam bentuk *slider* yang interaktif menggunakan plugin Swiper JS. Setiap *banner* merupakan berita yang ditampilkan dengan gambar latar (*background*) yang diambil dari *storage* menggunakan fungsi 'asset()'.
UNIVERSITAS
MULTIMEDIA
NUSANTARA

Bagian lain dalam antarmuka meliputi:

1. Berita Unggulan – menampilkan berita pilihan untuk pengguna.
2. Berita Terbaru – menampilkan berita terkini yang diproses secara dinamis dari database.

3. Kenali Author – menampilkan daftar penulis aktif.
4. Pilihan Author – menampilkan berita yang direkomendasikan oleh para penulis.

Semua bagian tersebut dirancang responsif dan konsisten secara visual menggunakan kombinasi Tailwind CSS dan struktur grid layout dari HTML. Blade syntax digunakan untuk memanggil data relasional seperti kategori berita, penulis, dan tanggal yang telah diformat dengan ‘Carbon‘.

Selain itu, penggunaan komponen seperti ‘@foreach‘, ‘ ‘, dan ‘@section‘ menandakan integrasi Blade yang bersih dan modular, memungkinkan pemisahan tampilan dan logika backend Laravel secara efisien.

3.3 Uraian Pelaksanaan Magang

Pelaksanaan kegiatan kerja magang di PT Winnicode Garuda Indonesia dilakukan secara *Work From Home* (WFH) dengan jam kerja yang telah ditentukan, yaitu pukul 08.00 hingga 18.00 WIB setiap hari kerja. Meskipun dilakukan secara daring, peserta magang tetap wajib mengikuti jadwal dan tanggung jawab yang diberikan secara disiplin.

Media yang digunakan dalam pelaksanaan magang adalah WhatsApp dan Google Classroom. WhatsApp digunakan sebagai sarana komunikasi langsung antara peserta magang dan *supervisor*, khususnya untuk kebutuhan konsultasi, diskusi teknis, serta penyampaian kendala yang dihadapi selama pengerjaan proyek. Sementara itu, Google Classroom digunakan sebagai platform untuk melakukan update progres proyek secara mingguan, termasuk pengumpulan laporan, dokumentasi, dan catatan capaian kerja.

Kegiatan magang berlangsung selama 5 bulan, di mana peserta magang diberikan kepercayaan penuh untuk menyelesaikan satu proyek secara mandiri (*end-to-end project*). Proyek tersebut mencakup seluruh tahapan pengembangan, mulai dari analisis kebutuhan, perancangan antarmuka dan sistem, implementasi *backend* dan *frontend*, pengujian sistem, hingga *deployment*. Peran yang dijalankan adalah sebagai *Full Stack Developer*, dengan tanggung jawab mencakup pengembangan aplikasi secara menyeluruh.

Seluruh pelaksanaan kerja magang dilakukan secara mandiri namun tetap dalam pengawasan *supervisor* melalui konsultasi dan pelaporan berkala.

Pelaksanaan kerja magang diuraikan seperti pada Tabel 3.1.

Tabel 3.1. Pekerjaan yang dilakukan tiap minggu selama pelaksanaan kerja magang

Minggu Ke -	Pekerjaan yang dilakukan
1	Pengenalan mengenai perusahaan dan kerja magang yang dilaksanakan
2	Pembuatan Design UI dan UX untuk website portal berita
3	Mempersiapkan kebutuhan <i>software</i> serta memastikan agar <i>environment</i> berjalan semestinya
4	Pembuatan <i>database migration</i> serta ERD
5-6	Pembuatan CRUD pada filament
7 -8	Pembuatan awal tampilan <i>website Frontend</i>
9	Pembuatan awal tampilan <i>website Frontend</i>
10	Pembuatan tampilan <i>website Frontend</i>
11	Pembuatan tampilan <i>website Frontend</i>
12-13	Pembuatan tampilan <i>Pembuatan tampilan isi berita dan author</i>
14	Pembuatan fitur-fitur pada <i>website</i>
15	Pembuatan fitur-fitur pada <i>website</i>
16	Pembuatan fitur dan isi pada <i>website</i>

Tabel 3.1 Rincian kegiatan mingguan yang dilakukan selama masa magang di PT Winnicode Garuda Indonesia, mencakup tahapan perencanaan, pengembangan, hingga *development* proyek pembuatan website portal berita.

3.4 Kendala dan Solusi Selama Pelaksanaan Magang

Selama menjalani kegiatan magang di PT Winnicode Garuda Indonesia, terdapat beberapa kendala yang dihadapi dalam pelaksanaan tugas, antara lain:

1. Minimnya pemahaman terhadap teknologi yang digunakan.
Peserta magang mengalami kesulitan dalam memahami *framework* utama yang digunakan dalam proyek, yaitu *VueJS*, karena belum memiliki pengalaman sebelumnya serta keterbatasan sumber belajar mengenai *framework* tersebut.
2. Komunikasi yang belum optimal dengan *supervisor*.
Interaksi antara peserta magang dan pembimbing belum berjalan secara maksimal, sehingga terkadang informasi terkait pekerjaan kurang

tersampaikan dengan baik dan menimbulkan kendala dalam pengerjaan tugas.

Untuk mengatasi permasalahan yang dihadapi, beberapa solusi yang diterapkan antara lain:

- Mencari referensi secara mandiri melalui berbagai platform seperti *Google*, *YouTube*, dan *StackOverflow* untuk memperdalam pemahaman terhadap *framework* yang digunakan.
- Menjalin komunikasi rutin dengan *supervisor* dalam rangka mendiskusikan hambatan-hambatan teknis dan administratif yang dihadapi selama pengerjaan proyek.
- Mendokumentasikan permasalahan yang muncul dan membahasnya dalam sesi diskusi agar bisa diatasi.

