

BAB III

PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi

Selama magang di PT Winnicode Garuda Teknologi, berperan sebagai *Fullstack Developer* di bawah Departemen *Development*. Kegiatan dilaksanakan dengan arahan langsung dari Bapak Muhammad Widyantoro selaku *Supervisor* Divisi. *Supervisor* bertanggung jawab memberikan bimbingan teknis serta mengevaluasi setiap progres pekerjaan yang dilakukan.

Posisi ini memberikan ruang eksplorasi teknologi yang cukup luas, khususnya dalam penerapan *JavaScript* untuk pengembangan sistem.^[4] Terdapat keleluasaan untuk mengusulkan solusi maupun pendekatan baru dalam setiap proyek yang dikerjakan. Pengalaman ini sangat bermanfaat dalam memahami penerapan teknologi secara langsung di dunia industri. Namun, untuk setiap perubahan tetap harus dikonsultasikan kepada *supervisor* guna menjaga stabilitas dan kesesuaian sistem.

3.2 Tugas yang dilakukan

Sepanjang mengikuti program magang sebagai *Fullstack Developer*, fokus utama pekerjaan terletak pada pengembangan *website* portal berita, yang merupakan bagian dari upaya pengembangan fitur portal berita di PT. Winnicode Garuda Teknologi.

Adapun tahapan-tahapan utama yang menjadi fokus dalam proyek ini meliputi:

1. Perancangan desain antarmuka pengguna (UI/UX) untuk *website* portal berita dengan memanfaatkan *Figma*.
2. Transformasi desain *Figma* menjadi kode program melalui implementasi *frontend* meliputi HTML, CSS, *JavaScript*.

3. Integrasi API *open source* melalui studi dokumentasi teknis untuk keperluan menampilkan konten berita secara dinamis pada *website*.
4. Mengembangkan berbagai fitur fungsional yang dibutuhkan untuk mendukung aktivitas pengguna.
5. Mengeksplorasi dan membangun sistem *backend* yaitu PHP yang mendukung pembuatan API serta sistem manajemen konten (CMS).

3.3 Uraian Pelaksanaan Magang

Proyek pengembangan *website* portal berita ini dilaksanakan secara mandiri dengan bimbingan rutin dari *supervisor* yang diberikan setiap minggu melalui *platform Google Classroom*. Meskipun jadwal pengerjaan bersifat fleksibel dan disusun secara individu, tanggung jawab dalam perancangan dan pengembangan proyek tetap menjadi kewajiban utama yang harus diselesaikan secara bertahap sesuai arahan dan standar yang ditetapkan oleh perusahaan. Pelaksanaan kegiatan magang didokumentasikan secara mingguan sebagaimana tercantum dalam Tabel 3.1.

Tabel 3.1. Aktivitas kerja setiap minggu

WAKTU	AKTIVITAS KERJA
Minggu 1	Menentukan ide dan teknologi yang akan digunakan, pencarian referensi desain, penyusunan proposal.
Minggu 2	Penentuan fitur <i>UI/UX</i> , riset teknologi <i>frontend</i> .
Minggu 3	Asistensi <i>supervisor</i> , implementasi desain ke dalam kode (<i>slicing</i>).
Minggu 4	Struktur <i>file</i> proyek, perancangan <i>login & signup page</i> , penyusunan komponen.
Minggu 5	Perancangan <i>homepage</i> , implementasi <i>layout</i> responsif.

Tabel 3.1. Aktivitas kerja setiap minggu (lanjutan)

WAKTU	AKTIVITAS KERJA
Minggu 6	Penambahan fitur-fitur <i>homepage</i> , optimasi tampilan responsif.
Minggu 7	Perancangan <i>categories, contact, about</i> , dan <i>profile page</i> .
Minggu 8	Perancangan halaman <i>admin</i> untuk pengelolaan konten, serta merapikan halaman <i>user</i> agar terstruktur & responsif.
Minggu 9	Perancangan antarmuka dan logika <i>search page</i> untuk menampilkan hasil yang relevan berdasarkan kata kunci pengguna.
Minggu 10	Finalisasi <i>frontend</i> , pembelajaran <i>backend</i> (PHP) dan memahami arsitektur <i>database</i> .
Minggu 11	Integrasi <i>frontend</i> dan <i>backend</i> (PHP & MySQL), serta pengujian CRUD untuk artikel berita.
Minggu 12	Penyempurnaan dan <i>debugging</i> sistem <i>backend</i> agar lebih stabil dan siap digunakan.

3.4 Tools yang digunakan

Selama periode magang di PT. Winnicode Garuda Teknologi sebagai *Fullstack Developer*, terdapat tanggung jawab untuk menguasai berbagai teknologi penting dalam pengembangan *website*. Pada bagian *Frontend*, dikenalkan dengan teknologi dasar seperti *JavaScript*, HTML, dan CSS yang menjadi fondasi utama dalam membangun antarmuka pengguna yang estetik dan responsif. HTML digunakan untuk membentuk struktur dasar halaman *website* agar konten dapat tersusun dengan baik. CSS digunakan untuk mempercantik tampilan halaman melalui pengaturan warna, tata letak, dan gaya visual lainnya. *JavaScript* berperan dalam menambahkan

fungsi interaktif agar pengguna dapat berinteraksi secara dinamis dengan halaman *website*^[5].

Pada sisi *backend*, pengembangan sistem dapat dilakukan dengan memanfaatkan PHP sebagai bahasa pemrograman *server-side* yang didukung oleh *MySQL* sebagai *database management system*. PHP berfungsi untuk menjalankan proses-proses *server*, seperti menangani permintaan (*request*) dari pengguna, memproses data, serta mengelola logika *website* di sisi *server*^[6]. *MySQL* digunakan untuk menyimpan, mengambil, dan mengelola data yang dibutuhkan oleh *website* secara efisien^[7]. Berikut merupakan perangkat yang menjadi sarana pendukung selama magang di PT. Winnicode Garuda Teknologi:

A. Perangkat Lunak

Dalam perancangan dan pembuatan *website* portal berita, perangkat lunak yang digunakan meliputi:

- *Visual Studio Code* sebagai *tools* utama untuk pengembangan kode program.
- *Github* berfungsi sebagai repositori penyimpanan *source code*.
- *Figma* untuk merancang tampilan antarmuka pengguna (UI/UX).
- XAMPP untuk menjalankan server lokal sebagai *backend* (PHP) dan *database*.
- *phpMyAdmin* untuk mengelola *database* secara visual melalui *browser*.
- *Google Classroom* untuk melaporkan progress setiap minggunya dan menerima arahan dari mentor.
- *WhatsApp* sebagai media komunikasi utama dengan tim PT. Winnicode Garuda Teknologi.

B. Perangkat Keras

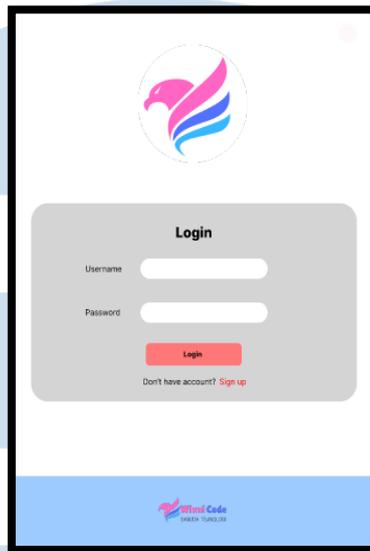
Dalam pelaksanaan proyek pengembangan *website* portal berita, beberapa perangkat keras utama yang digunakan antara lain:

1. Lenovo Ideapad Slim 3
 - Prosesor: Intel i3 10110U
 - Memory: 8GB
 - Sistem Operasi: Windows 11
 - Storage: 512GB
2. Jaringan Internet (*Wi-Fi*) untuk mengakses referensi dan pengujian *website*.

3.5 Project Utama

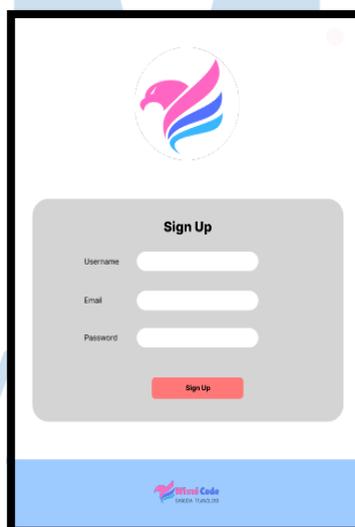
Proyek utama yang dikembangkan adalah sebuah *website* portal berita yang dibagi menjadi dua bagian utama berdasarkan peran pengguna, yaitu *user* dan *admin*. *User* merupakan pihak yang dapat mengakses *website* portal berita untuk membaca berbagai artikel yang telah dipublikasikan, sementara *admin* memiliki akses ke sistem *Content Management System* (CMS) yang berfungsi untuk mengelola konten berita. Melalui CMS, *admin* dapat menambah dan juga mengedit yang nantinya akan ditampilkan di halaman utama.

Tahap awal pengembangan difokuskan pada pembuatan *frontend* dari *website* utama yang bertugas menampilkan berbagai data yang tersimpan di dalam *database*. Karena ini merupakan tahap awal, maka data yang digunakan masih bersifat sementara atau *dummy* sebagai pengganti data asli. Data *dummy* tersebut diperoleh dari *opensource*, yaitu *NewsAPI*. Penggunaan data dari *NewsAPI* ini bertujuan agar memudahkan proses serta membantu menentukan tata letak desain yang sesuai dan efektif pada tampilan *website*^[8]. Dengan pendekatan ini, pengembangan *frontend* dapat berjalan lebih terstruktur sebelum akhirnya mengintegrasikan data sesungguhnya. Berikut ini merupakan tampilan *wireframe* dari *website* portal berita yang terdiri dari beberapa halaman utama, antara lain halaman *Login*, *Sign Up*, *Home*, *Categories*, *Contact*, *About*, *Profile*, dan *Admin*. Tampilan *wireframe login* dapat dilihat pada Gambar 3.1.



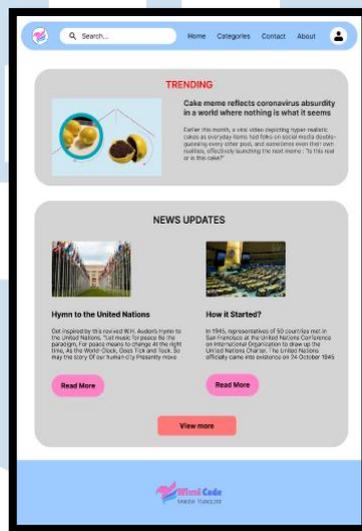
Gambar 3.1. Wireframe Login

Gambar 3.1 menampilkan antarmuka *login* yang memuat *field* untuk memasukkan identitas pengguna, yaitu *username* dan *password*, beserta *button* untuk proses *login*. Tujuannya untuk memberikan akses cepat dan langsung bagi pengguna yang telah terdaftar untuk masuk ke *website*. Untuk mendukung proses autentikasi pengguna, terdapat pula tampilan *wireframe* halaman *Sign Up* yang divisualisasikan dalam Gambar 3.2.



Gambar 3.2. Wireframe Sign Up

Pada Gambar 3.2, pengguna baru dapat membuat akun dengan mengisi data seperti *username*, *email*, dan *password*. Struktur halaman ini dibuat ringkas agar memudahkan pengguna dalam proses registrasi, dengan tombol *sign up* untuk mengirim data pendaftaran dan navigasi ke halaman *login* jika sudah memiliki akun. Selanjutnya, tampilan *wireframe* halaman *home* divisualisasikan dalam Gambar 3.3.



Gambar 3.3. Wireframe Home

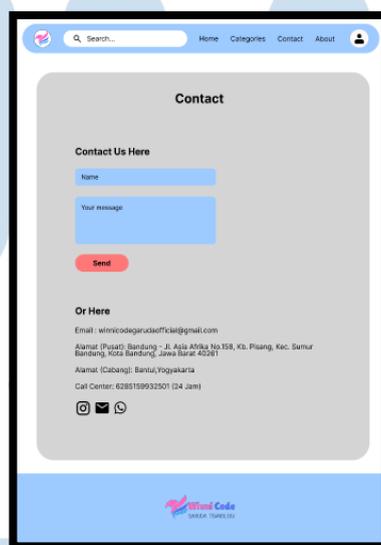
Gambar 3.3 merupakan beranda utama dari *website* portal berita, yang menampilkan konten berita terkini dan populer. Bagian ini dirancang dengan layout yang responsif dan terstruktur, mencakup komponen seperti *trending news* dan *news updates* agar pengguna langsung dapat mengakses informasi penting saat membuka *website*. Selain itu, terdapat pula tampilan *wireframe* dari halaman *Categories* yang dapat dilihat pada Gambar 3.4.

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 3.4. Wireframe Categories

Gambar 3.4 yang merupakan halaman *categories* menampilkan berbagai pilihan topik berita, seperti teknologi, olahraga, dan politik. Tata letak kategori dirancang sederhana agar mudah diakses dan dipahami, memungkinkan pengguna untuk memilih berita sesuai minat. Di samping halaman *categories*, terdapat pula *wireframe* halaman *contact* yang dapat dilihat pada Gambar 3.5.



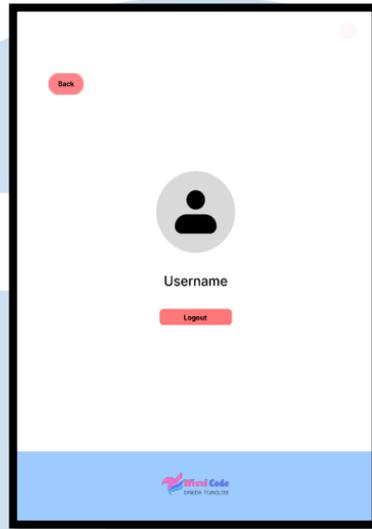
Gambar 3.5. Wireframe Contact

Pada Gambar 3.5, ditampilkan halaman *contact* yang berisi formulir isian berupa kolom nama dan pesan, yang memungkinkan pengguna mengirim pertanyaan, saran, atau masukan. Selain itu, tersedia informasi kontak seperti *email*, alamat kantor, dan ikon media sosial sebagai alternatif komunikasi. Halaman ini bertujuan untuk memfasilitasi interaksi dua arah antara pengguna dan pihak perusahaan secara mudah dan efisien. Terdapat pula tampilan *wireframe* untuk halaman *about* yang dapat dilihat pada Gambar 3.6.



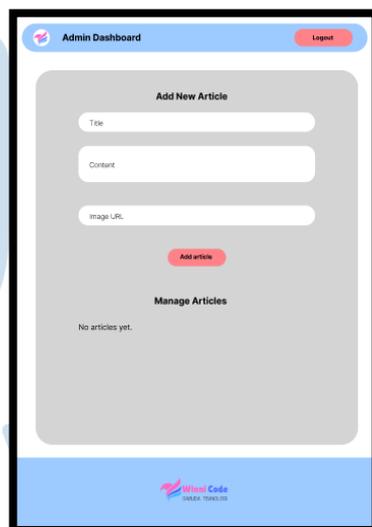
Gambar 3.6. Wireframe About

Gambar 3.6 menampilkan halaman About yang dirancang untuk menyampaikan berbagai informasi penting terkait pengembangan *website*, meliputi penjelasan mengenai latar belakang, visi misi, serta tujuan dari pengembangan *website* ini. Tujuannya adalah agar pengguna dapat mengenal lebih jauh nilai dan identitas perusahaan sebagai pengelola situs berita. Selain itu, terdapat pula *wireframe* halaman *profile* yang dapat dilihat pada Gambar 3.7.



Gambar 3.7. Wireframe Profile

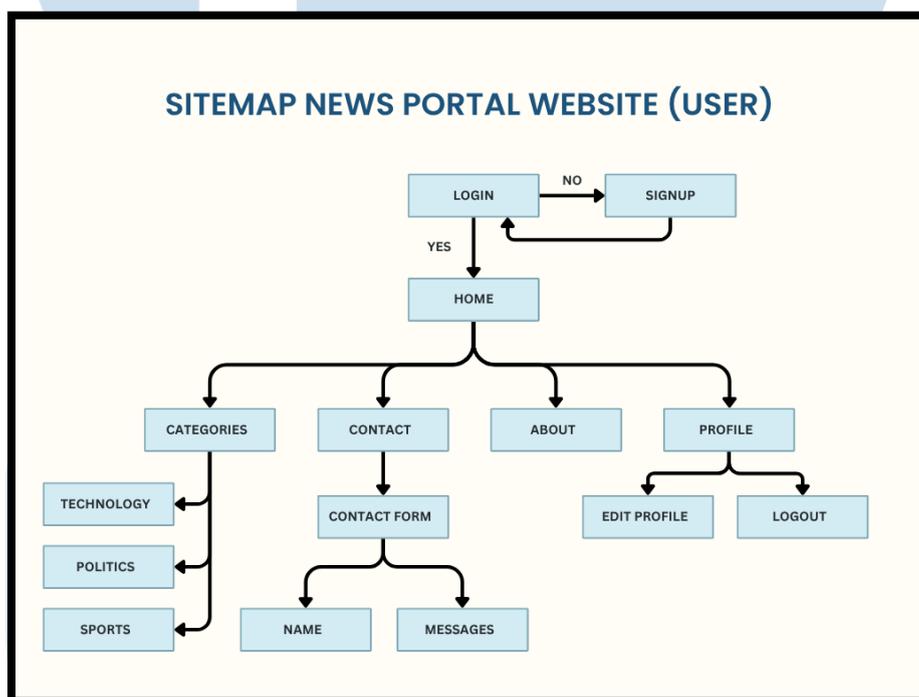
Pada Gambar 3.7, halaman profil memberikan ruang bagi pengguna untuk melihat dan mengelola akun. Halaman ini berisi informasi seperti nama pengguna dan tombol *logout*. Adapun untuk kebutuhan pengelolaan konten, disediakan halaman khusus untuk *admin*. Tampilan *wireframe* halaman *admin* dapat dilihat pada Gambar 3.8.



Gambar 3.8. Wireframe Admin

Pada Gambar 3.8, halaman *admin* ini dirancang khusus untuk kebutuhan pengelola *website* dalam melakukan pengelolaan konten berita. Melalui halaman ini, *admin* memiliki kendali penuh atas konten, mencakup CRUD (*Create, Read, Update, Delete*) terhadap data artikel, termasuk menambahkan berita baru, mengedit isi berita, dan menghapus berita. Desain halaman admin bersifat fungsional dan terfokus pada efisiensi manajemen data.

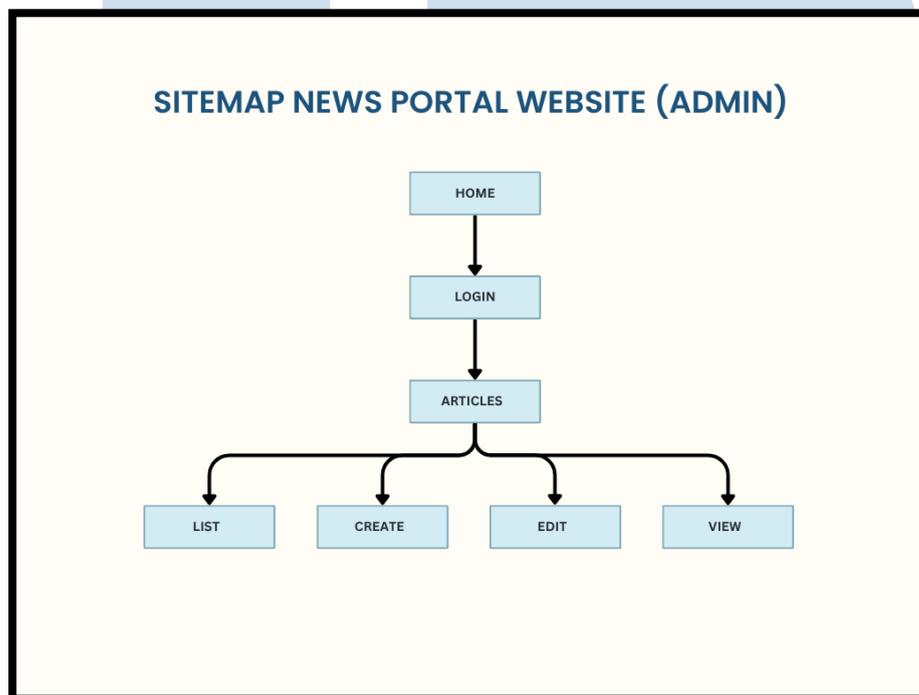
Adapun tampilan untuk *sitemap user* dan *sitemap admin* pada Gambar 3.9 dan Gambar 3.10



Gambar 3.9. Sitemap User

Pada Gambar 3.9 merupakan *sitemap user* untuk menggambarkan struktur halaman yang hanya dapat diakses oleh pengunjung maupun pengguna yang telah terdaftar. Alur dimulai dari halaman *login* dan *sign up* sebagai autentikasi pengguna, dilanjutkan dengan menuju halaman *home* yang

menampilkan berita-berita terbaru. Dari *home*, pengguna dapat mencari berbagai kategori berita, melihat detail setiap berita, serta mengakses halaman profil untuk melihat informasi akun pribadi. Tersedia pula halaman tambahan seperti *contact* dan *about* yang menyediakan sarana komunikasi dan latar belakang perusahaan secara singkat.

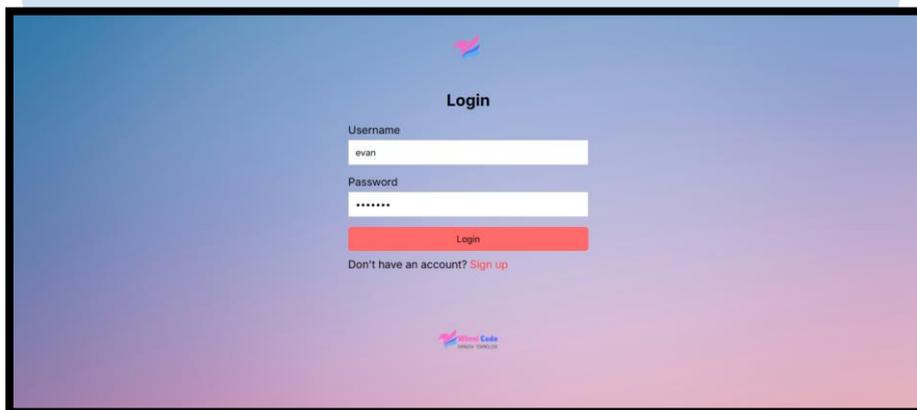


Gambar 3.10. Sitemap Admin

Gambar 3.10 menampilkan *sitemap admin* untuk mencerminkan arsitektur antarmuka yang berkaitan dengan pengelolaan sistem secara internal. Setelah proses *login* sebagai *admin*, akan diarahkan ke *dashboard* utama yang berisi fitur-fitur manajemen, seperti penambahan, pengeditan, dan penghapusan berita. Struktur ini dirancang untuk mendukung efisiensi kerja *admin* dalam mengelola konten berita (CRUD) sehingga seluruh informasi yang ditampilkan pada *website* tetap *up-to-date* dan mudah diatur sesuai kebutuhan.

A. Halaman Login

Halaman *login* merupakan tampilan awal yang pertama kali dilihat saat mengakses *website* portal berita. Halaman ini berfungsi untuk memverifikasi identitas melalui input *username* dan *password* sebelum memberikan akses ke fitur-fitur dalam sistem. Selain sebagai pintu masuk utama, *login page* juga berperan penting dalam menjaga keamanan data dan memastikan hanya yang terdaftar yang dapat mengakses konten dan layanan yang tersedia di dalam *website*. Tampilan halaman *login* dapat dilihat pada Gambar 3.11.



Gambar 3.11. Tampilan Halaman Login

Pada Gambar 3.11, *user* harus mengisi kolom *username* dan *password* sesuai data yang telah diregistrasi. Setelah mengisi formulir, *user* dapat menekan tombol *login* untuk mengakses sistem. Setelah proses autentikasi berhasil dengan data yang terdaftar dalam sistem, pengguna akan dialihkan secara otomatis ke antarmuka *dashboard*. Namun, apabila terdapat kesalahan pada *username* atau *password*, sistem akan menampilkan pesan *error*. Desain halaman *login* dibuat sederhana dengan latar belakang gradasi warna, guna memberikan tampilan yang menarik namun tetap fokus pada fungsi utama, yaitu keamanan dan kemudahan akses ke dalam sistem. Potongan kode *login.html* dapat dilihat pada Kode 3.1.

```

10 <div class="login-container">
11 <div class="logo">
12 
13 </div>
14
15 <div class="login-box">
16 <h2>Login</h2>
17 <form id="loginForm">
18 <label for="username">Username</label>
19 <input type="text" id="username" placeholder="Username" required />
20
21 <label for="password">Password</label>
22 <input type="password" id="password" placeholder="Password" required />
23
24 <button type="submit">Login</button>
25 </form>
26 <p>Don't have an account?&nbsp;<a href="signup.html">Sign up</a></p>
27 </div>

```

Kode 3.1. Login.html

Kode 3.1 merupakan struktur dasar halaman *login* pada sebuah *website*. Elemen utama dibungkus dalam `<div class="login-container">`, yang di dalamnya terdapat logo (``) dan *form login*. *Form login* ditandai dengan tag `<form id="loginForm">` yang berisi dua input: pertama untuk *username* (`<input type="text" id="username" ... />`) dan kedua untuk *password* (`<input type="password" id="password" ... />`), masing-masing disertai label agar mudah dikenali oleh pengguna. Tombol `<button type="submit">Login</button>` digunakan untuk mengirimkan data *input* ke *server* untuk proses autentikasi. Di bagian bawah *form* terdapat paragraf dengan tautan ke halaman pendaftaran (`signup.html`) bagi pengguna yang belum memiliki akun. Seluruh struktur ini dibuat agar halaman *login* fungsional dan mudah digunakan, dengan setiap input diberi atribut *required* untuk memastikan pengguna mengisi semua data sebelum *login*. Selain itu, terdapat pula potongan Kode 3.2 yang merupakan `login.js` sebagai validasi dan pengambilan input.

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

```

13     const username = document.getElementById("username").value.trim();
14     const password = document.getElementById("password").value.trim();
15
16     if (!username || !password) {
17         alert("Silakan isi username dan password!");
18         return;
19     }
20
21     if (username.length < 3 || password.length < 4) {
22         alert("Username minimal 3 karakter dan password minimal 4 karakter!");
23         return;
24     }

```

Kode 3.2. Login.js sebagai validasi dan pengambilan input

Kode 3.2 berfungsi sebagai skrip validasi input pada proses *login* di sisi klien (*client-side*). Baris pertama dan kedua mengambil nilai dari elemen input dengan id="username" dan id="password", lalu menggunakan `.trim()` untuk menghapus spasi di awal dan akhir teks. Selanjutnya, blok `if (!username || !password)` digunakan untuk memeriksa apakah kedua input telah diisi; jika salah satu kosong, maka akan muncul pesan peringatan melalui *alert* yang meminta pengguna untuk mengisi keduanya. Setelah itu, terdapat validasi panjang karakter pada *username* dan *password* menggunakan kondisi `if (username.length < 3 || password.length < 4)`; jika panjang *username* kurang dari 3 karakter atau *password* kurang dari 4 karakter, maka sistem akan menampilkan pesan peringatan bahwa panjang minimum harus dipenuhi. Jika kedua kondisi tersebut lolos, maka input akan dianggap *valid* dan proses *login* dapat dilanjutkan. Kode ini penting untuk memastikan bahwa pengguna memasukkan data dengan benar sebelum dikirim ke server. Terdapat pula potongan kode `login.js` sebagai user yang dapat dilihat pada Kode 3.3.

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

```

40     const userData = JSON.parse(localStorage.getItem("user_" + username));
41
42     if (userData && userData.password === password) {
43         localStorage.setItem(
44             "currentUser",
45             JSON.stringify({
46                 username: userData.username,
47                 email: userData.email,
48                 role: "user",
49             })
50         );
51         window.location.href = "home.html";
52     } else {
53         alert("Username atau password salah!");
54     }

```

Kode 3.3. Login.js sebagai user

Kode 3.3 merupakan bagian dari proses autentikasi pengguna pada halaman *login*, yang memanfaatkan *localStorage* sebagai media penyimpanan data akun. Baris pertama mengambil data pengguna dari *localStorage* menggunakan `getItem("user_" + username)`, lalu mengubah data tersebut dari format string JSON menjadi objek *JavaScript* dengan `JSON.parse()`. Kemudian, blok `if` memeriksa apakah data pengguna (`userData`) ada dan apakah *password* yang dimasukkan cocok dengan *password* yang tersimpan. Jika *valid*, maka data pengguna saat ini disimpan ke dalam *localStorage* dengan nama `currentUser`, berisi *username*, *email*, dan peran sebagai "user" dalam format JSON. Jika validasi berhasil, pengguna akan dialihkan ke halaman *home.html*. Sebaliknya, apabila validasi gagal, baik karena *username* tidak terdaftar maupun *password* tidak sesuai, sistem akan menampilkan notifikasi peringatan melalui `alert("Username atau password salah!")`.

C. Halaman Sign Up

Halaman *sign up* merupakan fitur yang memungkinkan pendaftaran akun baru pada *website* portal berita. Pada halaman ini, perlu mengisi data seperti *email*, *username*, dan *password* sebagai informasi dasar untuk membuat akun. Desain halaman dibuat sederhana dan mudah diakses guna memastikan proses registrasi berjalan lancar dan efisien bagi pengguna.

Fungsi utama dari halaman *sign up* adalah memberikan akses untuk *login*, serta memanfaatkan fitur-fitur interaktif lainnya yang tersedia di dalam sistem. Tampilan halaman *sign up* dapat dilihat pada Gambar 3.12.

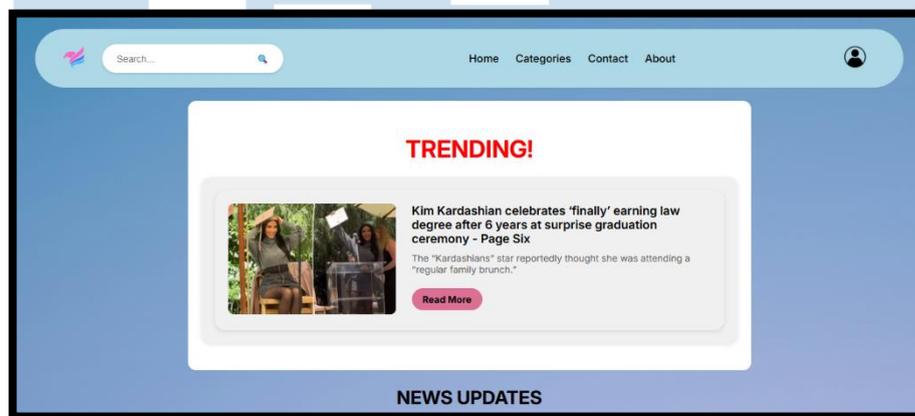
Gambar 3.12. Tampilan Halaman Sign Up

Gambar 3.12, terdapat antarmuka *sign up* yang berfungsi sebagai formulir pendaftaran akun baru bagi pengguna. Terdapat tiga field utama, yaitu *username*, alamat *email*, dan kata sandi, di mana setiap data yang dimasukkan akan divalidasi sebelum disimpan ke dalam sistem. Setelah pengguna mengisi ketiga kolom tersebut dan menekan tombol *sign up*, informasi pengguna akan disimpan di *backend server* agar dapat digunakan saat *login* berikutnya. Terdapat pula tautan teks di bagian bawah yang mengarahkan pengguna ke halaman *login* jika sudah memiliki akun.

D. Halaman Utama (Landing Page)

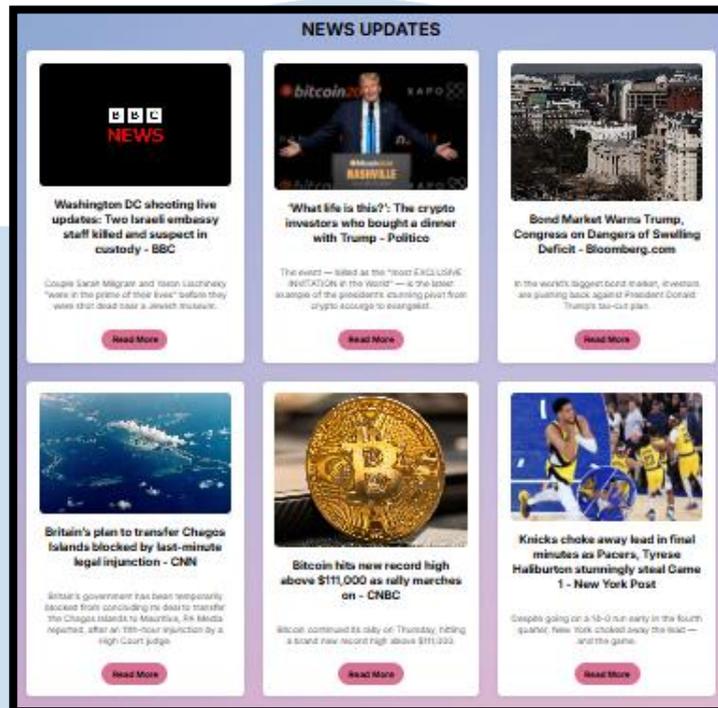
Halaman ini merupakan tampilan yang muncul setelah berhasil *login* ke dalam *website* portal berita. Desain halaman ini bertujuan memberikan kesan awal yang menarik serta menyampaikan informasi penting secara langsung. Proses pengembangan dilakukan dengan menggunakan HTML sebagai struktur utama, CSS untuk mengatur tampilan visual, dan *JavaScript* untuk menambahkan elemen interaktif serta fungsionalitas dinamis. Kombinasi ketiga teknologi ini menghasilkan pengalaman

penggunaan yang sederhana, responsif, dan mudah diakses. Beberapa bagian penting ditampilkan pada halaman utama, seperti *section Trending News* yang berisi berita terkini dan sedang populer, disusun secara acak agar variasi informasi tetap segar. Selain itu, *section News Updates* menampilkan kumpulan berita terbaru yang juga diacak secara berkala untuk menjaga konten tetap dinamis dan menarik. Tampilan *Landing Page* tersaji dalam Gambar 3.13 dan Gambar 3.14.



Gambar 3.13 Tampilan Trending News

Gambar 3.13 menampilkan halaman utama dari *website* portal berita, terutama pada bagian *Trending News*. Di bagian atas terdapat *header* yang memuat logo, *search bar*, serta menu navigasi utama seperti *Home*, *Categories*, *Contact*, dan *About*, yang memudahkan pengguna dalam mengakses berbagai halaman. Di sebelah kanan atas juga terdapat ikon profil yang menunjukkan bahwa pengguna telah *login*. Pada bagian konten utama terdapat segmen khusus yang menampilkan berita *Trending*, ditandai dengan judul merah mencolok, berisi highlight berita yang sedang populer, lengkap dengan gambar, judul, kutipan singkat, serta tombol *Read More* yang mengarahkan ke detail berita.



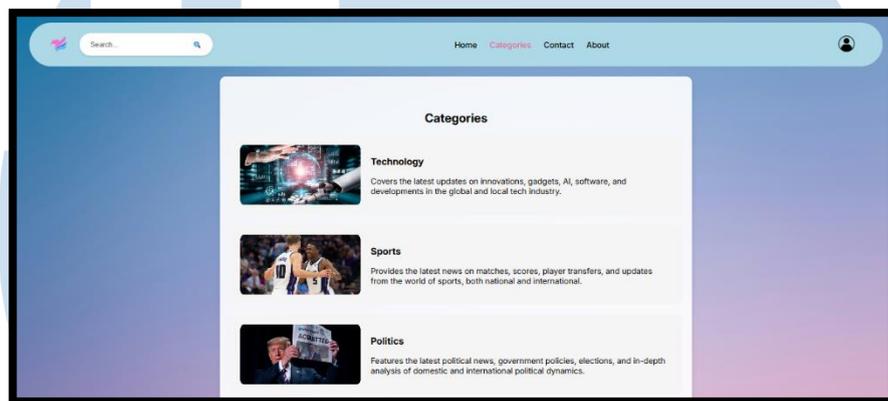
Gambar 3.14. Tampilan News Updates

Gambar 3.14 menampilkan bagian *News Updates* yang terletak di bawah bagian *Trending News*. Setiap bagian terdiri dari *thumbnail* berita, judul berita, ringkasan singkat, serta tombol *Read More* yang memungkinkan pengguna membaca berita secara lengkap. Desain *layout* dibuat dalam *grid* tiga kolom dengan tampilan yang rapi dan responsif, sehingga memudahkan pembaca dalam menavigasi dan memilih berita yang ingin dibaca.

E. Halaman Kategori Berita

Halaman kategori menampilkan berbagai jenis berita yang dikelompokkan berdasarkan topik tertentu, seperti *Technology*, *Sports*, *Politics* yang dipilih berdasarkan topik yang paling populer dan sering diakses. Setiap kategori dilengkapi dengan gambar serta penjelasan singkat yang menggambarkan isi dari jenis berita yang tersedia. Gambar membantu memperkuat konteks dari masing-masing kategori, sedangkan deskripsi singkat memberikan gambaran umum mengenai topik yang dibahas.

Penyajian ini bertujuan untuk memberikan informasi awal yang jelas dan memudahkan dalam memilih kategori yang sesuai dengan kebutuhan atau minat. Tampilan halaman kategori dapat dilihat pada Gambar 3.15.



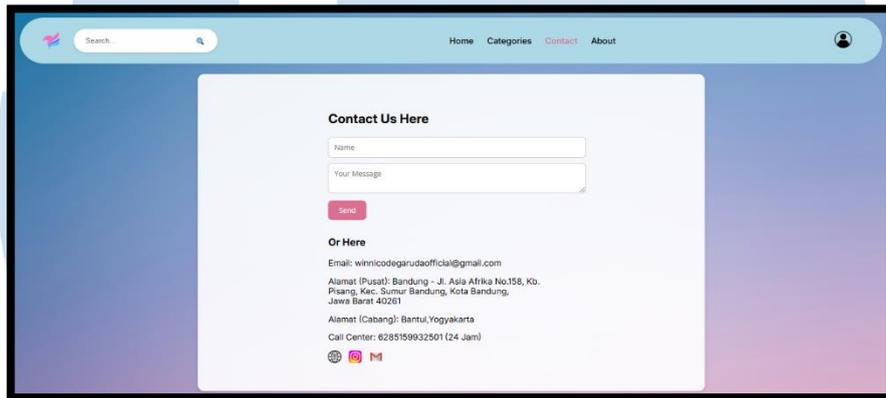
Gambar 3.15. Tampilan Kategori Berita

Gambar 3.15 menampilkan masing-masing kategori dilengkapi dengan gambar ilustratif di sebelah kiri dan deskripsi singkat di sebelah kanan. Kategori *Technology* mencakup informasi terkait inovasi, *gadget*, *AI*, *software*, serta perkembangan teknologi lainnya. *Sports* menyajikan berita seputar pertandingan, skor, transfer pemain, dan pembaruan dari dunia olahraga nasional dan internasional. Sementara *Politics* berfokus pada berita politik, kebijakan pemerintah, pemilu, dan lainnya.

F. Halaman Kontak

Halaman kontak berfungsi sebagai sarana komunikasi yang memungkinkan pengiriman pesan secara langsung kepada pihak pengelola, yaitu Winnicode Garuda Teknologi. Pada halaman ini, tersedia formulir yang terdiri dari kolom nama dan pesan yang dapat digunakan untuk menyampaikan pertanyaan, saran, maupun masukan lainnya. Selain formulir, halaman ini juga menyediakan informasi kontak alternatif seperti alamat *email*, alamat kantor pusat dan cabang, serta nomor *call center* yang aktif selama 24 jam. Untuk memudahkan akses lebih lanjut, ditampilkan

pula ikon media sosial dan *email* yang dapat langsung dihubungi. Tata letak halaman dirancang sederhana dan mudah dipahami agar proses penyampaian informasi dapat berjalan dengan cepat dan efisien. Tampilan halaman kontak dapat dilihat pada Gambar 3.16.



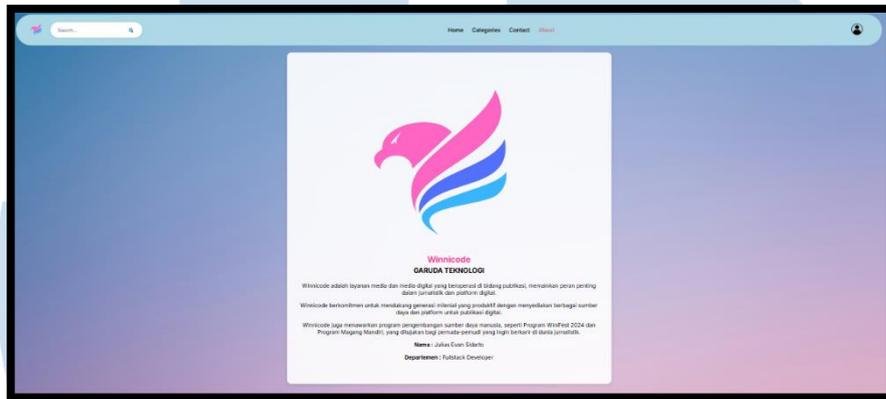
Gambar 3.16. Tampilan Halaman Kontak

Gambar 3.16 menampilkan sebuah antarmuka yang dirancang secara sederhana namun profesional, dengan tampilan visual yang bersih dan latar gradasi warna biru ke ungu yang memberikan kesan tenang dan ramah pengguna. Seluruh elemen pada halaman ini disusun secara rapi, jelas, dan mudah dipahami, sehingga mendukung komunikasi dua arah yang cepat, efisien, dan terpercaya antara pengguna dan pengelola situs.

G. Halaman About

Halaman ini menyajikan informasi mengenai Winnicode Garuda Teknologi sebagai pengelola *platform* berita digital yang berfokus pada publikasi dan pengembangan jurnalisme digital. Halaman ini menjelaskan peran Winnicode dalam menyediakan sumber daya serta mendukung generasi muda melalui program seperti WiniFest 2024 dan Program Magang Mandiri. Selain itu, informasi mengenai nama dan departemen yang bertanggung jawab dicantumkan sebagai bentuk penghargaan atas

kontribusi yang telah diberikan dalam proses pengembangan. Desain halaman *about* dapat diamati pada Gambar 3.17.



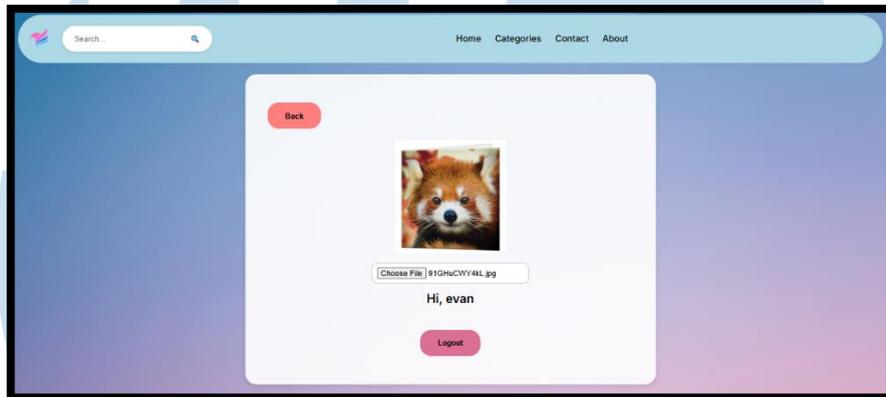
Gambar 3.17. Tampilan Halaman About

Gambar 3.17 menampilkan logo ikonik berbentuk kepala burung garuda dengan kombinasi warna pink dan biru yang mencerminkan semangat dan inovasi. Deskripsi pada halaman menjelaskan bahwa Winnicode bertujuan menjadi sarana edukatif dan informatif dalam dunia teknologi industri, serta menyediakan berbagai konten dan sumber daya untuk mendukung pengembangan keterampilan digital. Tampilan halaman ini menunjukkan bagaimana identitas visual dan misi perusahaan ditampilkan secara konsisten dan profesional dalam sebuah *website*.

H. Halaman Profile

Halaman ini menyajikan tampilan sederhana yang berisi nama pengguna serta foto profil yang dapat diperbarui melalui fitur unggah gambar. Desainnya dibuat minimalis dengan fokus pada identitas pengguna, memungkinkan personalisasi tampilan akun secara mudah dan membantu meminimalisir kebingungan saat mengakses fitur yang tersedia. Selain itu, terdapat *button* untuk kembali ke halaman sebelumnya serta opsi *logout*

sebagai kontrol utama dalam pengelolaan sesi akun. Untuk melihat tampilan halaman *profile*, dapat merujuk pada Gambar 3.18.



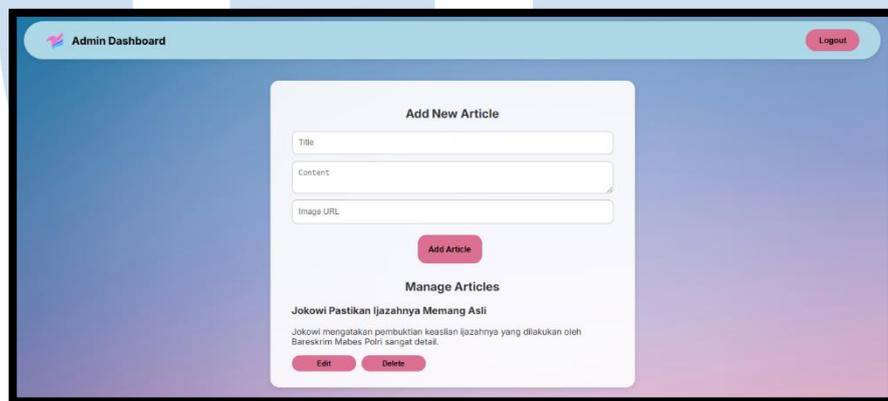
Gambar 3.18. Tampilan Halaman Profile

Gambar 3.18 menampilkan halaman profil pengguna *website* Winnicode yang dirancang dengan antarmuka sederhana dan responsif. Halaman ini menampilkan foto profil pengguna, tombol untuk mengganti gambar (*upload file*), sapaan personal “Hi, evan” yang menandakan pengguna telah *login*, serta tombol *logout* untuk keluar dari akun. Terdapat opsi kembali berupa tombol *back* yang diletakkan di area atas halaman. Desain halaman ini mencerminkan fungsionalitas dasar dari sistem autentikasi pengguna, termasuk pengenalan identitas, personalisasi tampilan, dan kontrol akun secara langsung melalui antarmuka yang *user-friendly*.

I. Halaman Admin

Halaman *admin* dirancang sebagai pusat kendali yang memungkinkan pengelolaan artikel secara menyeluruh melalui fitur CRUD (*Create, Read, Update, Delete*). Melalui tampilan yang minimalis dan *user-friendly*, *admin* dapat dengan mudah menambahkan artikel baru hanya dengan mengisi judul, konten, serta URL gambar, lalu mempublikasikannya

secara langsung. Setiap artikel yang sudah ditambahkan juga dapat diperbarui atau dihapus sesuai kebutuhan, sehingga kontrol terhadap konten tetap terjaga dan dinamis. Untuk menjaga keamanan akses, halaman ini hanya diperuntukkan oleh *admin* resmi dengan *username* admin dan *password* admin123. Dengan sistem ini, pengelolaan konten dapat dilakukan secara cepat, aman, dan terorganisir. Tampilan panel *admin* dapat diamati pada Gambar 3.19.



Gambar 3.19. Tampilan Halaman Admin

Pada Gambar 3.19, terdapat label “*Admin Dashboard*” serta tombol *logout* untuk keluar dari sesi admin. Bagian utama halaman terbagi menjadi dua: fitur *Add New Article* dan *Manage Articles*. Pada bagian *Add New Article*, *admin* dapat menambahkan artikel baru dengan mengisi tiga input *field*: judul (*Title*), isi konten (*Content*), dan URL gambar (*Image URL*), lalu menekan tombol *Add article* untuk menyimpan data. Di bawahnya, pada bagian *Manage Articles*, ditampilkan daftar artikel yang telah dipublikasikan lengkap dengan judul dan isi singkat, serta dua tombol aksi yaitu *edit* dan *delete* yang memungkinkan *admin* untuk memperbarui atau menghapus artikel tersebut. Halaman ini menunjukkan fungsionalitas sistem manajemen konten (CMS) sederhana yang memberikan kemudahan bagi *admin* dalam melakukan kontrol penuh terhadap publikasi artikel secara efisien dan terstruktur.

J. Header

Desain *header (navigation)* pada *website* portal berita ini dirancang untuk memberikan kemudahan akses ke semua fitur yang tersedia. *Header* ini mencakup beberapa elemen, diantaranya logo *website*, kemudian menu *home*, *categories*, *contact*, *about*, serta fitur pencarian berita (*search bar*). Selain itu, terdapat *profile* dimana berfungsi untuk *logout* dan *edit profile*. Penempatan elemen ini tidak hanya fokus ke fungsionalitas saja, tetapi juga berfokus pada kemudahan akses dan desain yang simetris serta sederhana. Untuk visualisasi *header* dapat dilihat pada Gambar 3.20, sementara potongan kode terdapat dalam Kode 3.4.



Gambar 3.20. Tampilan header pada website

```
10 <body>
11   <div class="header">
12     <div class="header-left">
13       <div class="homelogo">
14         
15       </div>
16
17       <div class="search-bar">
18         <form id="searchForm" action="#" method="GET">
19           <input type="text" placeholder="Search..." id="searchInput" name="search" />
20           <button type="submit" id="searchBtn">🔍</button>
21         </form>
22       </div>
23     </div>
24
25     <nav>
26       <ul>
27         <li><a href="home.html">Home</a></li>
28         <li><a href="categories.html">Categories</a></li>
29         <li><a href="contact.html">Contact</a></li>
30         <li><a href="about.html">About</a></li>
31       </ul>
32     </nav>
```

Kode 3.4. Header

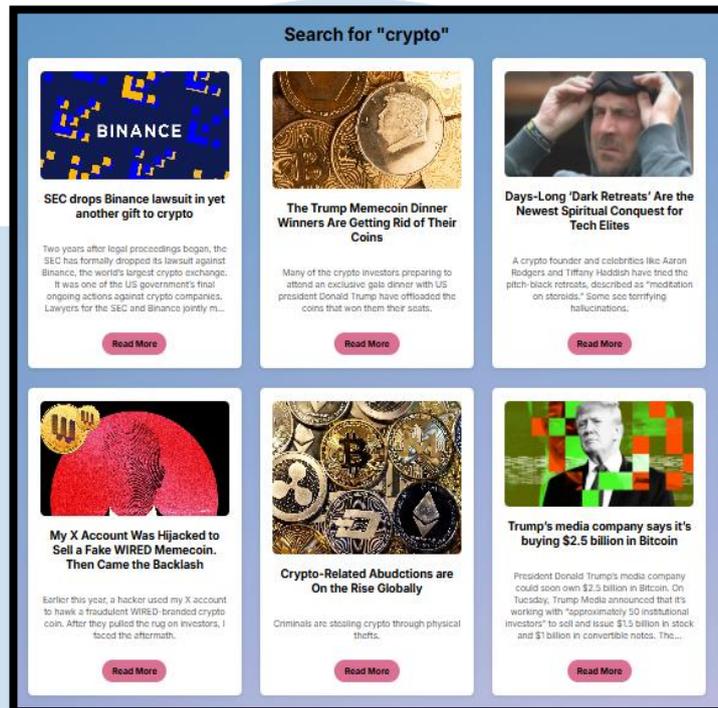
Kode 3.4 merupakan struktur HTML untuk bagian *header* sebuah *website* portal berita yang dirancang agar pengguna dapat dengan mudah mengenali identitas *website* dan menavigasi halaman. Elemen `<div class="header">` berfungsi sebagai wadah utama yang membungkus semua komponen di

dalam *header*. Di bagian kiri terdapat logo *website* yang ditampilkan melalui tag `` dalam `<div class="homelogo">`, lalu di tengah terdapat fitur pencarian berupa `<form>` yang memungkinkan pengguna memasukkan kata kunci melalui `<input>` dan mengirimkannya dengan tombol `<button>`. Selanjutnya, elemen `<nav>` digunakan sebagai navigasi utama yang memuat daftar menu dalam tag `` dengan beberapa item ``, masing-masing berisi tag `<a>` yang mengarah ke halaman seperti *Home*, *Categories*, *Contact*, dan *About*.

K. Halaman Pencarian

Halaman pencarian (*search page*) memfasilitasi pengguna untuk menemukan konten berdasarkan kata kunci yang diinput pada kolom pencarian. Setelah pengguna mengetikkan kata kunci dan memulai pencarian, sistem akan berkomunikasi dengan *server* melalui *NewsAPI* menggunakan metode GET untuk mengambil berita yang relevan dengan kata kunci yang dimasukkan. Hasil yang diterima akan ditampilkan dalam bentuk daftar berisi judul berita, ringkasan singkat, dan tautan menuju artikel lengkap. Dengan tampilan yang responsif dan mudah digunakan, fitur ini memungkinkan pengguna memperoleh informasi dengan efisien, cepat, dan praktis. Tampilan *search page* terdapat pada Gambar 3.21.

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



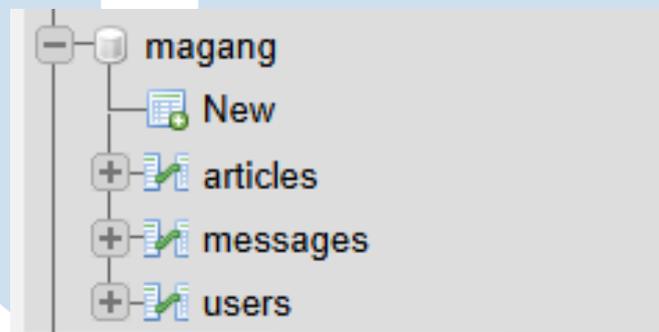
Gambar 3.21. Tampilan Search Page

Pada Gambar 3.21 memperlihatkan elemen *heading* (h1) yang menampilkan konten secara dinamis mengikuti input pencarian pengguna. Sebagai ilustrasi, jika pengguna mengetikkan kata "crypto", maka teks yang muncul akan berupa "Search for 'crypto'" untuk memastikan bahwa hasil yang ditampilkan sesuai dengan kata kunci tersebut. Fitur pencarian ini dibangun dengan HTML sebagai kerangka utama, sementara desain visual didukung oleh CSS dan untuk menciptakan interaksi pencarian yang dinamis, digunakan *Javascript*. Kemudian, ketika pengguna melakukan pencarian, sistem akan menginisiasi permintaan data ke API eksternal melalui metode *fetch*, memastikan proses pengambilan informasi berita berjalan optimal.

L. Halaman Backend

Pada sistem ini, halaman *backend* dikembangkan menggunakan bahasa PHP yang berperan sebagai *server-side scripting*^[9]. PHP dipilih karena kemampuannya untuk memproses data secara dinamis, mengelola

session, dan berinteraksi langsung dengan *database MySQL*. *Backend* bertanggung jawab dalam mengatur seluruh logika, termasuk autentikasi *admin*, manajemen data artikel, serta kontrol terhadap fitur-fitur portal berita. Setiap permintaan dari *frontend* yang berkaitan dengan manipulasi data, seperti penambahan, pengeditan, atau penghapusan berita, diproses oleh PHP yang kemudian berkomunikasi dengan *database* untuk menyimpan atau mengambil data sesuai kebutuhan. Adapun struktur data pada *MySQL* yang ditampilkan pada Gambar 3.22.



Gambar 3.22. Struktur data MySQL

Gambar 3.22 diatas merupakan *database* magang yang dikelola menggunakan *tool phpMyAdmin*, yaitu aplikasi berbasis *website* yang mempermudah pengelolaan *database MySQL* melalui antarmuka grafis tanpa perlu mengetikkan perintah SQL secara manual^[10]. Di dalam *database* ini terdapat tiga tabel utama, yaitu *articles*, *messages*, dan *users*.

- Tabel *articles* berfungsi untuk menyimpan data artikel atau berita, yang biasanya mencakup kolom seperti *id*, *title*, *content*, *author_id*, dan *created_at* untuk mendukung sistem manajemen konten pada *website*.
- Tabel *messages* digunakan untuk menampung pesan atau masukan dari pengguna yang dikirim melalui *form* kontak, dengan struktur data seperti *id*, *name*, *email*, *message*, dan *sent_at*.
- Tabel *users* menyimpan informasi akun *admin* atau *user* yang memiliki akses ke halaman *backend*, dengan kolom penting seperti

id, username, dan password_hash untuk keperluan autentikasi dan keamanan *login*.

3.6 Kendala dan Solusi

Selama proses pengerjaan *website* portal berita ini, terdapat berbagai kendala yang terjadi beserta solusi yang dapat diterapkan.

1. Kendala

- Keterbatasan komunikasi karena bekerja dari rumah, membuat komunikasi dengan *supervisor* menjadi terbatas. Tanpa tatap muka, beberapa arahan atau klarifikasi tugas sering kali tidak tersampaikan dengan jelas, sehingga berpotensi menimbulkan miskomunikasi terhadap tugas yang diberikan.
- Keterbatasan dalam penguasaan bahasa pemrograman, khususnya PHP untuk membangun pengembangan *backend* serta integrasi dengan database MySQL.
- Gangguan koneksi atau kecepatan internet yang tidak stabil sering menghambat proses, terutama saat harus mengunggah file besar, mengakses *server*, dan sebagainya.

2. Solusi

- Diupayakan adanya jadwal rutin pertemuan daring menggunakan *platform* seperti *Google Meet* atau *Zoom* agar dapat berkomunikasi secara langsung dan lebih efektif dalam membahas *progress*, kendala, pemahaman peserta, serta mendapatkan masukan yang jelas dari *supervisor*.
- Melakukan pencarian referensi tutorial *online*, serta forum diskusi seperti *Stack Overflow*. Selain itu, pendekatan *trial and error* juga dilakukan untuk menguji dan memahami bagaimana setiap baris kode bekerja.
- Menyiapkan alternatif koneksi seperti *tethering* dari ponsel agar pekerjaan tetap dapat dilanjutkan meski koneksi utama mengalami gangguan.