

BAB III

PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi

Selama pelaksanaan kegiatan magang di PT. Winnicode Garuda Teknologi, berperan sebagai *Fullstack Developer* pada Departemen *Development*. Koordinasi dilakukan secara daring melalui *Google Classroom*, dengan kewajiban pengunggahan laporan progres pekerjaan secara mingguan. Laporan tersebut mencakup aktivitas perancangan, fitur yang telah dikerjakan, serta lampiran, dan digunakan sebagai dasar pemantauan pelaksanaan magang oleh Bapak Muhammad Widyatoro selaku *Supervisor* Divisi *Fullstack Developer*.

Posisi ini berfokus pada perancangan dan pembangunan aplikasi berbasis *website* rekrutmen karyawan yang bertujuan mendukung proses seleksi internal perusahaan. Pengembangan dilakukan menggunakan PHP dan teknologi pendukung lainnya, mencakup implementasi fitur pendaftaran serta pengelolaan data pelamar, hingga integrasi antarmuka yang mudah digunakan. Selain pengembangan fitur utama, juga dilakukan penyesuaian teknis untuk memastikan alur rekrutmen berjalan efisien dan sesuai kebutuhan perusahaan. Seluruh tahapan perancangan dikonsultasikan secara berkala dengan *supervisor* guna menjaga stabilitas sistem serta kesesuaian dengan standar yang diterapkan.

3.2 Tugas yang dilakukan

Selama menjalani kegiatan magang dengan posisi sebagai *Fullstack Developer*, tugas utama yang dilakukan berfokus pada perancangan aplikasi rekrutmen karyawan berbasis *website*, yang merupakan bagian dari upaya pengembangan fitur *website* rekrutmen karyawan di PT. Winnicode Garuda Teknologi.

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

1. Perancangan (UI/UX) untuk *website* rekrutmen karyawan dengan menggunakan *Figma*.
2. Implementasi *frontend* dilakukan dengan menerjemahkan rancangan antarmuka dari *Figma* ke dalam struktur kode program berbasis PHP, CSS, dan *JavaScript*.
3. Pengembangan *backend* dan *database* menggunakan PHP untuk mengelola data pelamar, data lowongan, dan proses seleksi.
4. Integrasi fitur rekrutmen dan pengujian sistem meliputi manajemen lowongan, pengelolaan lamaran, serta status seleksi.

3.3 Uraian pelaksanaan magang

Proyek perancangan aplikasi rekrutmen karyawan berbasis *website* ini dilaksanakan secara mandiri dengan pemantauan dari *supervisor* melalui laporan mingguan yang dikumpulkan di *Google Classroom*. Meskipun penyusunan jadwal kerja bersifat fleksibel, proses perancangan dan pembangunan tetap menjadi tanggung jawab yang harus diselesaikan secara bertahap sesuai standar dari perusahaan. Seluruh aktivitas magang dicatat dan dilaporkan setiap minggu sebagaimana ditampilkan pada Tabel 3.1.

Tabel 3.1. Dokumentasi kegiatan setiap minggu

WAKTU	KEGIATAN
Minggu 1	Penyusunan proposal proyek
Minggu 2	Studi literatur dan analisis kebutuhan sistem

Tabel 3.1. Dokumentasi kegiatan setiap minggu (lanjutan)

WAKTU	KEGIATAN
Minggu 3	Penyusunan spesifikasi sistem dan rancangan alur proses
Minggu 4	Pembuatan desain <i>database</i> awal
Minggu 5	Perancangan desain UI/UX
Minggu 6	Perancangan prototipe antarmuka menggunakan HTML, PHP, CSS, <i>JavaScript</i>
Minggu 7	Integrasi awal frontend dengan <i>CodeIgniter</i>
Minggu 8	Finalisasi <i>database</i>
Minggu 9-10	Implementasi fitur <i>login</i> dan <i>signup</i>
Minggu 11-12	Implementasi fitur lamaran kerja dan <i>upload CV</i>
Minggu 13-14	Implementasi fitur manajemen lowongan kerja
Minggu 14-15	Implementasi fitur <i>dashboard admin, manager, supervisor</i>
Minggu 16-17	Unit <i>testing</i> dan <i>debugging</i>

Tabel 3.1. Dokumentasi kegiatan magang setiap minggu (lanjutan)

WAKTU	KEGIATAN
Minggu 18-19	Dokumentasi sistem dan persiapan <i>hosting</i>
Minggu 20	Launching <i>website</i> dan evaluasi akhir proyek

3.4 Tools yang digunakan

Selama pelaksanaan magang pada PT. Winnicode Garuda Teknologi berperan sebagai *Fullstack Developer*, fokus perancangan diarahkan pada pemahaman teknologi yang digunakan dalam membangun aplikasi rekrutmen karyawan berbasis *website*. Pada bagian *frontend*, teknologi yang diterapkan meliputi HTML, CSS, dan *JavaScript* sebagai dasar pembentukan antarmuka. HTML digunakan untuk menyusun struktur halaman, mencakup *form* pendaftaran, daftar lowongan, hingga detail informasi pelamar. CSS berfungsi mengatur tampilan visual, mulai dari warna, tata letak, hingga elemen desain agar *website* tampil menarik dan profesional. *JavaScript* digunakan untuk memberikan interaksi dinamis, seperti validasi *form* dan respons interaktif yang dapat meningkatkan kenyamanan pengguna dalam menggunakan *website*.

Pada bagian *backend*, pengembangan sistem dilakukan menggunakan PHP yang dipadukan dengan *framework CodeIgniter*. PHP bertugas menjalankan logika *server*, seperti pemrosesan data lamaran, autentikasi *user*, *admin*, *manager*, *supervisor* dan pengelolaan alur rekrutmen. *Framework CodeIgniter* membantu menata struktur *website* agar lebih terorganisir, efisien, serta mudah dikembangkan. Basis data *MySQL* digunakan untuk menyimpan dan mengelola berbagai data penting, termasuk informasi pelamar, data lowongan pekerjaan, hingga riwayat proses seleksi secara terstruktur.

A. Perangkat Lunak

Dalam proses perancangan dan pembangunan aplikasi rekrutmen karyawan berbasis *website*, perangkat lunak yang digunakan meliputi:

1. *Visual Studio Code* sebagai editor utama untuk menulis dan mengelola program.
2. *Github* digunakan sebagai repositori penyimpanan *source code*.
3. *Figma* berfungsi untuk merancang antarmuka pengguna (UI/UX) dan menyusun struktur tampilan *website*.
4. XAMPP digunakan sebagai server lokal untuk menjalankan PHP, *CodeIgniter*, dan *database* selama tahap perancangan dan pembangunan.
5. *phpMyAdmin* digunakan untuk mengelola *database MySQL* melalui antarmuka berbasis *browser*.
6. *Google Classroom* digunakan sebagai media pelaporan perkembangan proyek serta menerima arahan dari *supervisor*.
7. *WhatsApp* digunakan sebagai sarana komunikasi utama dengan *supervisor* dan tim PT. Winnicode Garuda Teknologi.
8. *CodeIgniter* sebagai *framework* PHP untuk mempercepat pembuatan struktur *backend* dan mengatur alur logika sistem.

B. Perangkat Keras

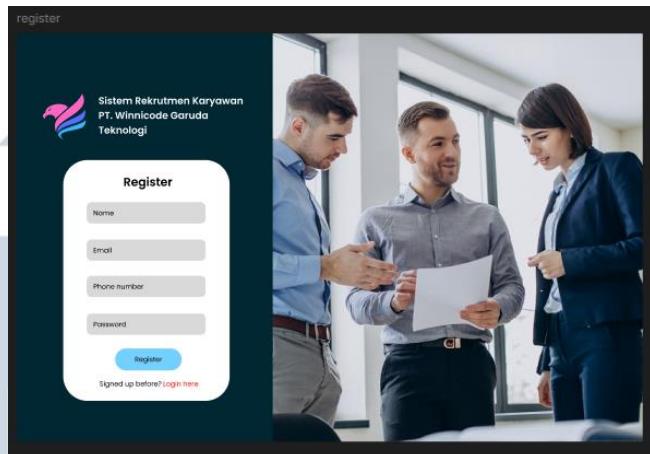
Dalam proses perancangan dan pembangunan *website* rekrutmen karyawan, adapun perangkat keras yang digunakan antara lain:

1. Lenovo Ideapad Slim 3
 - *Processor*: Intel i3 10110U
 - *Memory*: 8GB
 - *OS*: Windows 11
 - *Storage*: 512GB
2. Jaringan Internet untuk mencari referensi dan *testing website*.

3.5 Project Utama

Proyek utama yang dirancang adalah sebuah aplikasi rekrutmen karyawan berbasis *website* yang memiliki empat jenis akses, yaitu pelamar, *admin*, *manager*, dan *supervisor*. Pelamar dapat menggunakan *website* untuk melihat lowongan pekerjaan, membaca detail posisi yang tersedia, serta mengirimkan lamaran. *Admin* berperan menambahkan lowongan, mengedit informasi posisi, hingga memeriksa data pelamar yang masuk. *Manager* berperan dalam mengelola proses rekrutmen secara keseluruhan, termasuk meninjau data pelamar yang telah diseleksi oleh *admin*. *Supervisor* memiliki akses untuk memantau status lowongan pekerjaan dan data pelamar yang masuk ke dalam sistem. *Supervisor* dapat melihat informasi pelamar serta perkembangan proses rekrutmen, namun tidak memiliki kewenangan untuk melakukan persetujuan atau seleksi akhir terhadap pelamar.

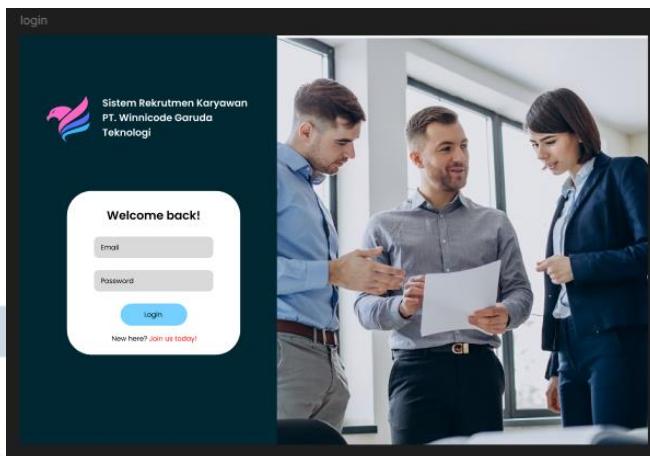
Pada tahap awal perancangan dan pembangunan *website* rekrutmen karyawan, difokuskan pada pembuatan antarmuka pengguna (*frontend*). *Frontend* ini dirancang untuk menampilkan berbagai informasi yang nantinya diambil dari basis data, seperti data lowongan, informasi status lamaran, serta akun pelamar. Pada tahap ini juga dibuat *wireframe* sebagai gambaran awal tampilan *website* rekrutmen karyawan. *Wireframe* tersebut mencakup beberapa halaman utama, antara lain *Sign Up*, *Login*, *Dashboard*, Lowongan Pekerjaan, dan Profil Pengguna. Salah satu contoh tampilan *wireframe*, yaitu halaman *Register*, ditampilkan pada Gambar 3.1.



Gambar 3.1. Wireframe Register

Gambar 3.1 menunjukkan rancangan antarmuka halaman *register* yang berisi kolom *name*, *email*, *phone number*, *password*, serta tombol *register*. Halaman ini dirancang untuk memungkinkan calon pelamar membuat akun secara mandiri sebelum menggunakan fitur-fitur yang tersedia di dalam *website*.

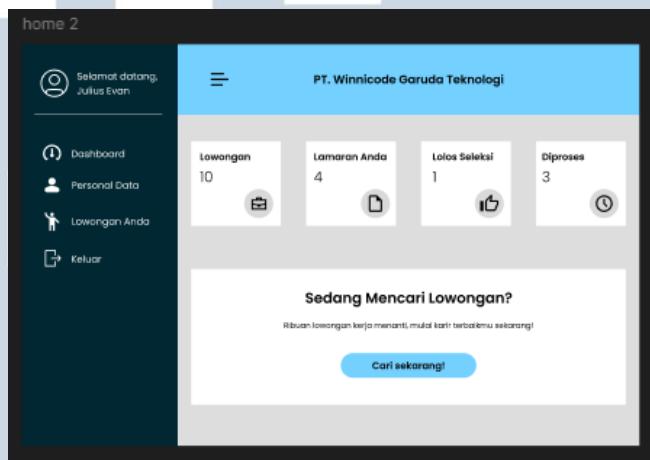
Untuk proses autentikasi selanjutnya, terdapat pula tampilan *wireframe* halaman *Login* yang dapat dilihat pada Gambar 3.2.



Gambar 3.2. Wireframe Login

Gambar 3.2 menampilkan *wireframe* halaman *login* sebagai pintu masuk calon pelamar ke dalam *website*. Pada tampilan ini tersedia kolom

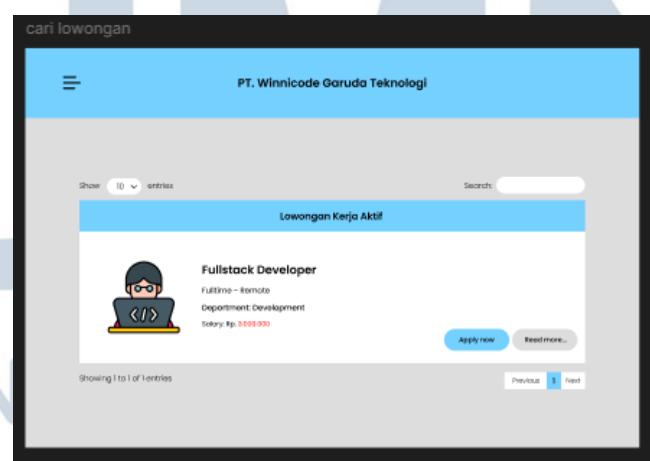
untuk *email*, *password*, dan tombol *login* untuk proses autentikasi calon pelamar. Selanjutnya, rancangan *wireframe dashboard* dapat dilihat pada Gambar 3.3.



Gambar 3.3 Wireframe Dashboard

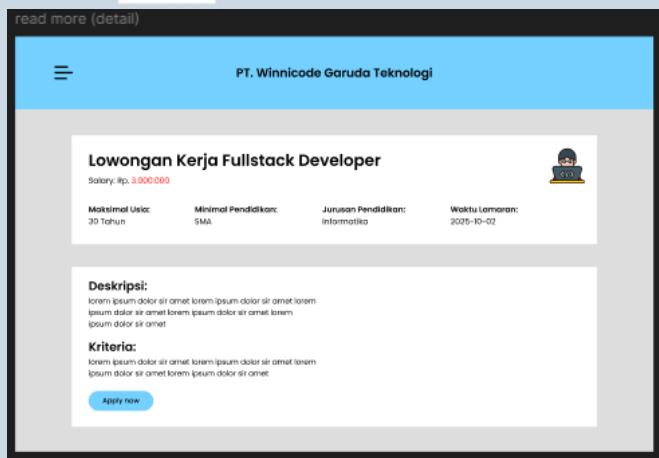
Gambar 3.3 menampilkan *dashboard* yang berfungsi sebagai pusat informasi bagi calon pelamar untuk menampilkan jumlah lowongan yang dilamar, status proses lamaran, dan lainnya.

Berikutnya, *wireframe* untuk fitur pencarian lowongan dapat dilihat pada Gambar 3.4.



Gambar 3.4. Wireframe Pencarian Lowongan

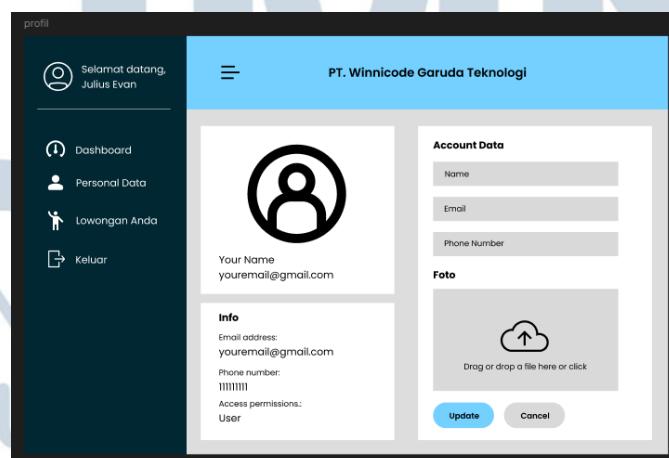
Gambar 3.4 diatas menunjukkan *wireframe* pencarian lowongan untuk menelusuri lowongan pekerjaan yang tersedia, sehingga mempermudah dalam mencari lowongan yang sesuai dengan kriteria. Selanjutnya, tampilan *wireframe* detail lowongan dapat dilihat pada Gambar 3.5.



Gambar 3.5. Wireframe Detail Lowongan

Gambar 3.5 menampilkan *wireframe* detail lowongan pekerjaan yang memuat detail seperti nama posisi, deskripsi pekerjaan, kualifikasi, serta tombol untuk mengajukan lamaran.

Selanjutnya, *wireframe* profil pelamar dapat dilihat pada Gambar 3.6.

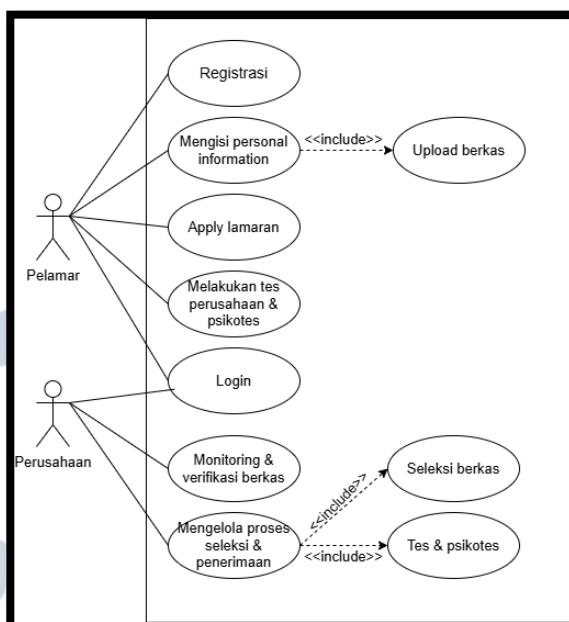


Gambar 3.6. Wireframe Profil Pelamar

Gambar 3.6 menampilkan *wireframe* profil pelamar yang berisi informasi data diri pelamar seperti nama, *email*, nomor telepon, dan fitur untuk mengubah *profile picture* pelamar.

3.5.1 Desain Sistem

Pada tahap ini dilakukan pemodelan sistem menggunakan UML untuk menggambarkan kebutuhan fungsional dan alur proses sistem. Pemodelan meliputi *Use Case Diagram* yang menunjukkan interaksi antara aktor dan fungsi-fungsi yang tersedia, serta *Activity Diagram* yang memvisualisasikan alur aktivitas sistem dari awal hingga akhir. Selain itu, *Class Diagram* digunakan untuk menggambarkan struktur sistem secara statis, mencakup kelas, atribut, metode, serta hubungan antar kelas sebagai dasar implementasi program. Setelah pemodelan UML, *flowchart* digunakan untuk memperjelas alur logika dan proses kerja sistem secara lebih detail dan terstruktur, sehingga memudahkan pemahaman tahapan proses serta implementasi fitur pada *website* rekrutmen karyawan. Pemodelan *Use Case Diagram* dapat dilihat pada Gambar 3.7.

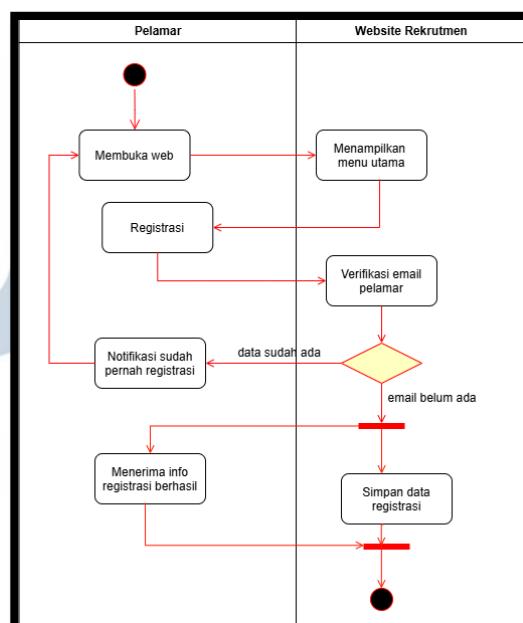


Gambar 3.7. Use Case Diagram

Gambar 3.7 menggambarkan interaksi antara dua aktor utama, yaitu pelamar dan pihak perusahaan, dalam sistem rekrutmen *online*. Pelamar memiliki akses untuk melakukan beberapa fungsi meliputi registrasi, *login*, mengisi *personal information*, mengunggah berkas, melamar pekerjaan, serta mengikuti tes perusahaan dan psikotes. Aktivitas upload berkas menjadi bagian dari proses pengisian data pelamar yang ditunjukkan melalui hubungan *<<include>>*, karena fitur ini wajib dilakukan.

Sementara itu, perusahaan dapat melakukan *login*, *monitoring* dan verifikasi berkas lamaran, serta mengelola proses seleksi dan penerimaan. Dalam proses seleksi tersebut terdapat aktivitas wajib berupa seleksi berkas serta tes dan psikotes, yang juga dihubungkan dengan relasi *<<include>>* karena menjadi bagian dari tahapan seleksi pelamar. Untuk mencegah terjadinya duplikasi data, sistem menerapkan bahwa pelamar wajib melakukan registrasi guna mendapatkan akun dengan *email* sebagai *username*.

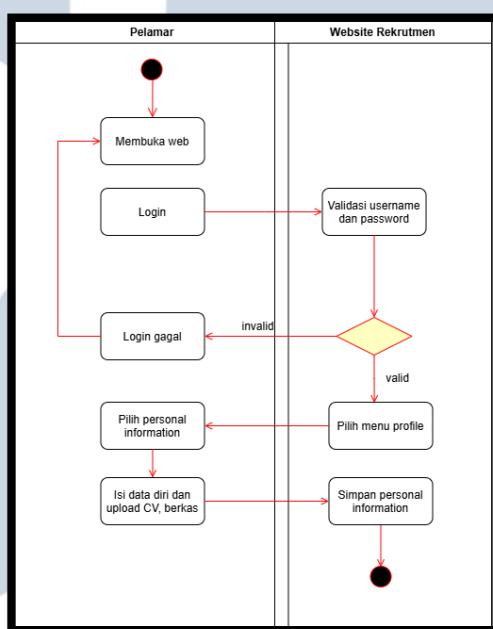
Selanjutnya, untuk pemodelan *Activity Diagram* pada bagian registrasi dapat dilihat pada Gambar 3.8.



Gambar 3.8. Activity Diagram Registrasi

Gambar 3.8 menunjukkan alur proses pendaftaran akun pelamar yang dimulai dari ketika pelamar membuka *website* dan memilih menu registrasi, kemudian sistem melakukan verifikasi terhadap *email* yang dimasukkan untuk memastikan *email* tersebut belum terdaftar. Apabila *email* sudah terdaftar, sistem akan menampilkan notifikasi bahwa pelamar telah melakukan registrasi sebelumnya, sedangkan jika *email* belum terdaftar, sistem akan menyimpan data registrasi ke dalam *database* dan memberikan informasi bahwa proses registrasi berhasil. Diagram ini menunjukkan mekanisme pengendalian alur registrasi untuk mencegah duplikasi data serta memastikan proses pendaftaran berjalan secara terstruktur dan sistematis.

Selanjutnya, untuk tampilan *activity diagram profile* dapat dilihat pada Gambar 3.9.

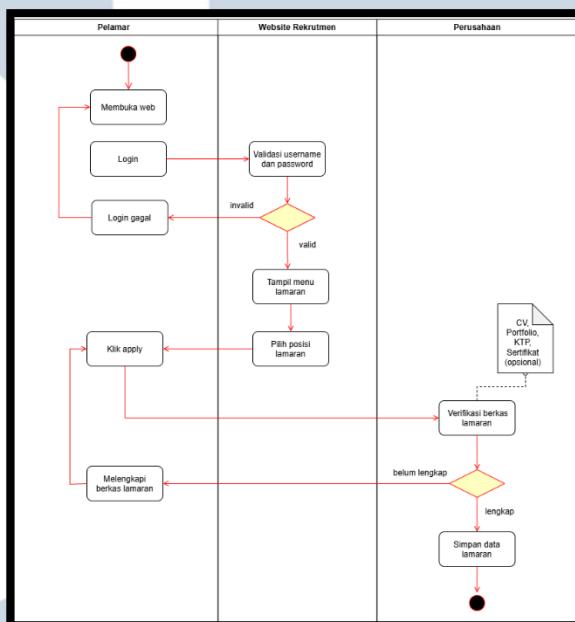


Gambar 3.9. Activity Diagram Profile

Gambar 3.9 menggambarkan alur proses pengelolaan data profil pelamar pada *website* rekrutmen karyawan. Proses dimulai ketika pelamar membuka *website* dan melakukan *login*, kemudian sistem melakukan

validasi *username* dan *password*. Apabila data tidak valid, sistem akan menampilkan informasi *login* gagal, sedangkan jika valid, pelamar dapat mengakses menu *profile*. Selanjutnya, pelamar mengisi data diri serta mengunggah CV dan berkas pendukung, kemudian sistem menyimpan seluruh *personal information* ke dalam *database*. Diagram ini menunjukkan bahwa pengisian profil hanya dapat dilakukan setelah proses autentikasi berhasil, sehingga menjamin keamanan data serta keteraturan alur pengelolaan informasi pelamar dalam sistem.

Karena data pelamar telah masuk ke dalam *database*, maka pada prosedur melamar pekerjaan cukup dengan memilih posisi pekerjaan yang diinginkan untuk selanjutnya melakukan *apply* lamaran seperti yang dijelaskan pada *Activity Diagram Apply Lamaran* pada Gambar 3.10.

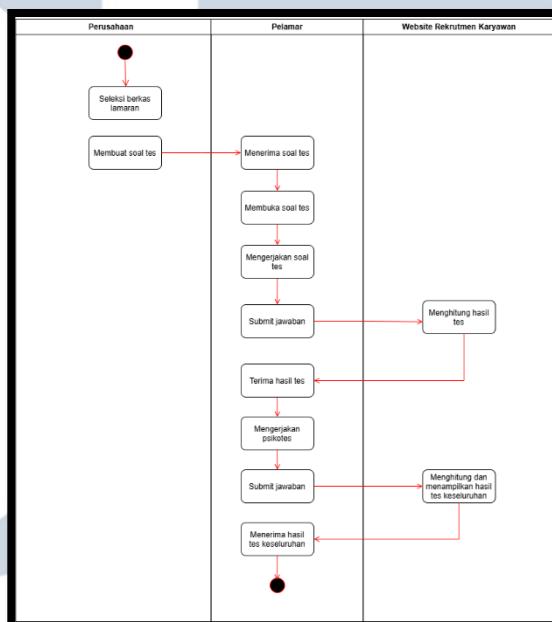


Gambar 3.10. Activity Diagram Apply Lamaran

Gambar 3.10 menunjukkan alur proses *apply* lamaran melalui *website* rekrutmen karyawan yang melibatkan interaksi antara pelamar, sistem, dan pihak perusahaan. Proses dimulai ketika pelamar membuka *website* dan melakukan *login*, kemudian sistem melakukan validasi

username dan *password*. Apabila data tidak valid, pelamar akan menerima informasi *login* gagal, sedangkan jika valid, sistem menampilkan menu lamaran. Selanjutnya, pelamar memilih posisi pekerjaan yang diinginkan dan menekan tombol *apply* untuk mengajukan lamaran. Setelah itu, pelamar melengkapi berkas lamaran yang diperlukan, seperti CV, portofolio, KTP, dan sertifikat pendukung. Berkas yang dikirimkan kemudian diverifikasi oleh sistem dan pihak perusahaan untuk memastikan kelengkapan data. Apabila berkas belum lengkap, pelamar diminta untuk melengkapi kembali dokumen lamaran, sedangkan jika berkas dinyatakan lengkap, sistem akan menyimpan data lamaran ke dalam *database* dan proses pengajuan lamaran dinyatakan selesai.

Setelah pelamar menyelesaikan proses pengajuan lamaran, sistem melanjutkan ke tahap tes sebagaimana ditunjukkan pada *Activity Diagram* Tes pada Gambar 3.11.

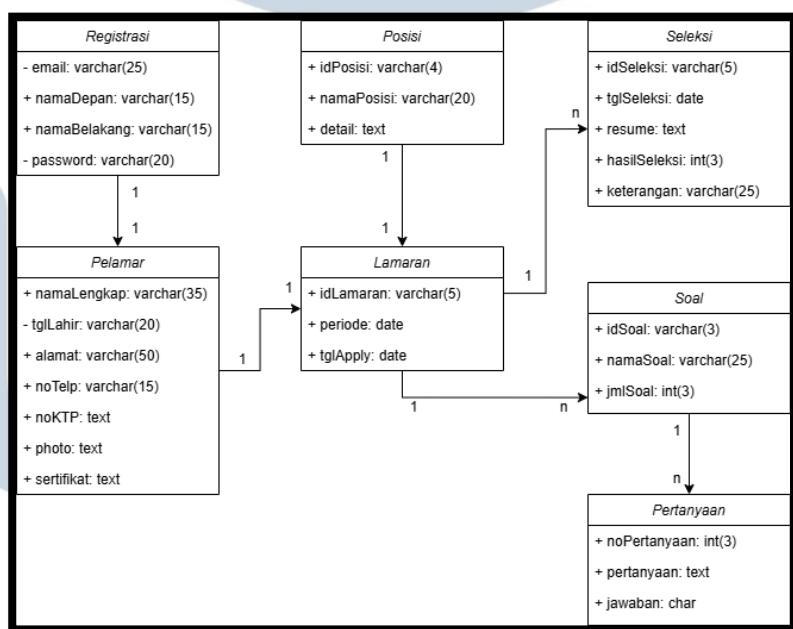


Gambar 3.11. Activity Diagram Tes

Gambar 3.11 menggambarkan proses seleksi data lamaran beserta berkas pelamar yang telah tersimpan di dalam *database*, yang selanjutnya akan dievaluasi oleh pihak perusahaan berdasarkan kelayakan dan

kesesuaian kualifikasi. Setelah dinyatakan memenuhi kriteria, perusahaan menyiapkan dan mengirimkan soal tes kepada pelamar melalui *website* rekrutmen. Pelamar kemudian menerima soal tes, membuka soal, dan mengerjakan tes sesuai ketentuan, lalu mengirimkan jawaban melalui sistem. *Website* rekrutmen secara otomatis memproses dan menghitung hasil tes tersebut, kemudian hasilnya dikirim kembali kepada pelamar. Selanjutnya, pelamar melanjutkan ke tahap psikotes dengan mengerjakan soal yang tersedia dan kembali mengirimkan jawaban melalui sistem. *Website* rekrutmen akan menghitung serta menampilkan hasil tes secara keseluruhan, dan hasil akhir tersebut diterima oleh pelamar sebagai penutup dari rangkaian proses seleksi.

Selain itu, perancangan sistem dilakukan untuk memberikan gambaran yang jelas mengenai alur data yang akan diimplementasikan ke dalam *database*, di mana struktur tabel-tabel *database* divisualisasikan dalam bentuk *class diagram* sebagaimana ditunjukkan pada Gambar 3.12.



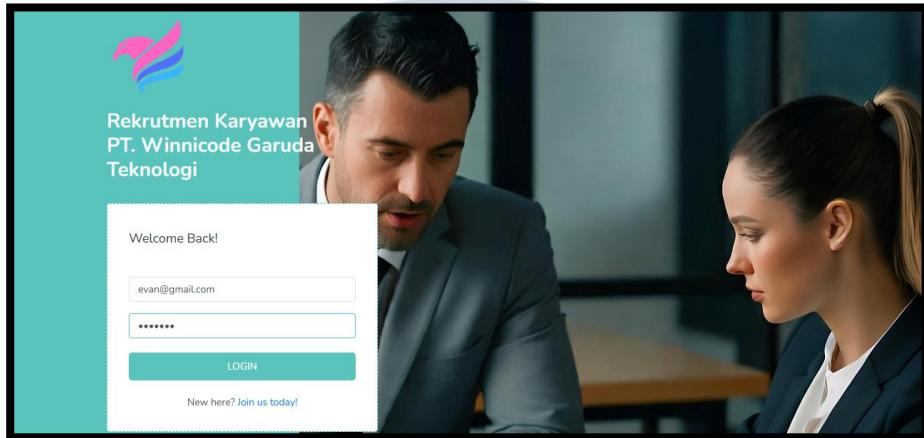
Gambar 3.12. Class Diagram Website Rekrutmen Karyawan

Gambar 3.12 menggambarkan struktur data dan hubungan antar kelas pada sistem rekrutmen karyawan. Kelas Registrasi menyimpan data akun awal pelamar yang kemudian terhubung satu-ke-satu dengan kelas Pelamar, yang memuat data pribadi lengkap seperti identitas, kontak, dan berkas pendukung. Pelamar dapat mengajukan Lamaran untuk suatu Posisi tertentu, di mana setiap lamaran terkait dengan satu posisi dan dicatat berdasarkan periode serta tanggal pendaftaran. Selanjutnya, setiap lamaran akan melalui proses Seleksi, yang berisi informasi tanggal seleksi, hasil penilaian, dan keterangan status seleksi.

Dalam proses seleksi, sistem menggunakan Soal yang terdiri dari beberapa Pertanyaan. Satu soal dapat memiliki banyak pertanyaan, sedangkan setiap pertanyaan menyimpan teks pertanyaan dan jawaban yang benar. Secara keseluruhan, *class diagram* ini menunjukkan alur rekrutmen mulai dari registrasi pelamar, pengajuan lamaran, hingga proses seleksi berbasis soal dan pertanyaan, serta hubungan antar data yang mendukung proses tersebut secara terstruktur dan terintegrasi.

A. Halaman Login

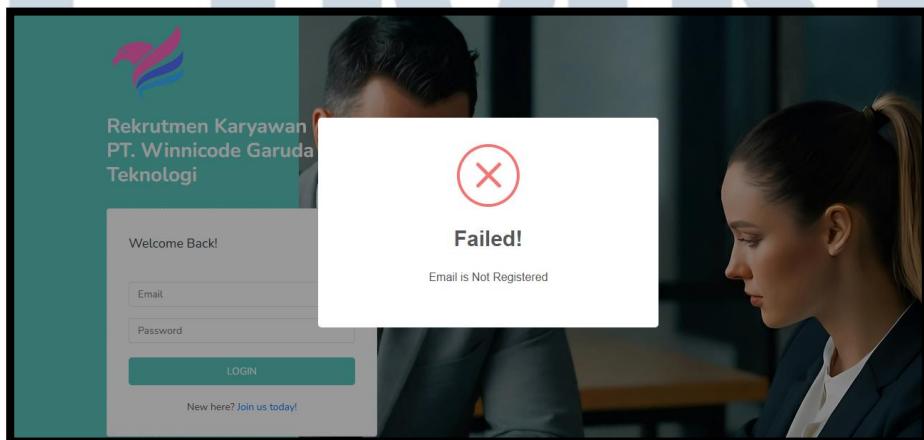
Halaman *login* merupakan tampilan awal yang diakses oleh pelamar, *admin*, *manager*, ataupun *supervisor* saat memasuki *website* rekrutmen karyawan. Halaman ini berfungsi untuk melakukan proses autentikasi sesuai *role* masing-masing melalui input *email* dan *password* sebelum dapat mengakses fitur-fitur yang tersedia. Selain sebagai gerbang utama, halaman ini juga berperan penting dalam keamanan data serta memastikan hanya pelamar yang terdaftar yang dapat mengakses layanan rekrutmen yang disediakan. Tampilan halaman *login* dapat dilihat pada Gambar 3.13.



Gambar 3.13. Tampilan Halaman Login

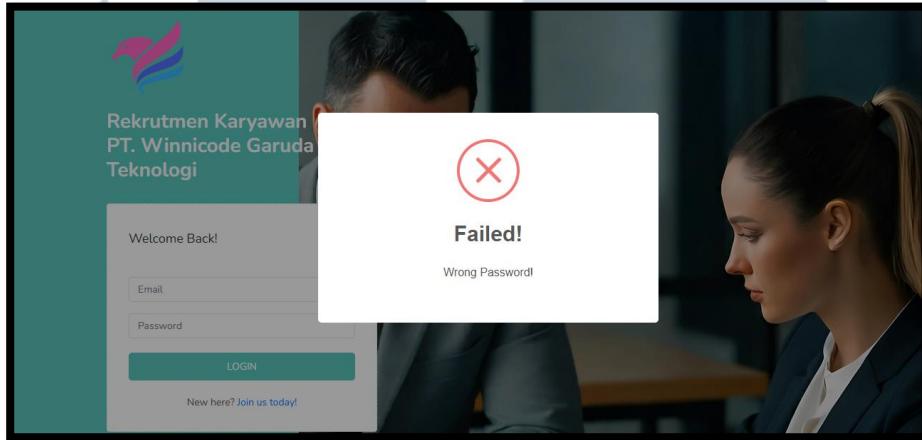
Gambar 3.13 menampilkan halaman *login* pada website rekrutmen karyawan. Halaman ini digunakan sebagai pintu masuk bagi pengguna untuk mengakses sistem sesuai dengan hak akses masing-masing. Pengguna diharuskan memasukkan *email/username* dan kata sandi yang telah terdaftar. Sistem akan melakukan proses autentikasi untuk memastikan kesesuaian data sebelum memberikan akses ke halaman berikutnya.

Apabila *email* yang dimasukkan tidak terdaftar dalam sistem, maka akan muncul tampilan pesan *error* sebagaimana ditunjukkan pada Gambar 3.14.



Gambar 3.14. Pesan Error Email Tidak Terdaftar

Apabila *password* yang dimasukkan juga tidak sesuai, sistem akan menampilkan pesan *error* sebagaimana ditunjukkan pada Gambar 3.15.



Gambar 3.15. Pesan Error Password Salah

Selanjutnya, sistem diimplementasikan menggunakan potongan kode program yang bertugas melakukan pengecekan kesesuaian *email* dan *password* serta menampilkan pesan kesalahan apabila terjadi ketidaksesuaian data. Potongan kode dapat dilihat pada Kode 3.1.

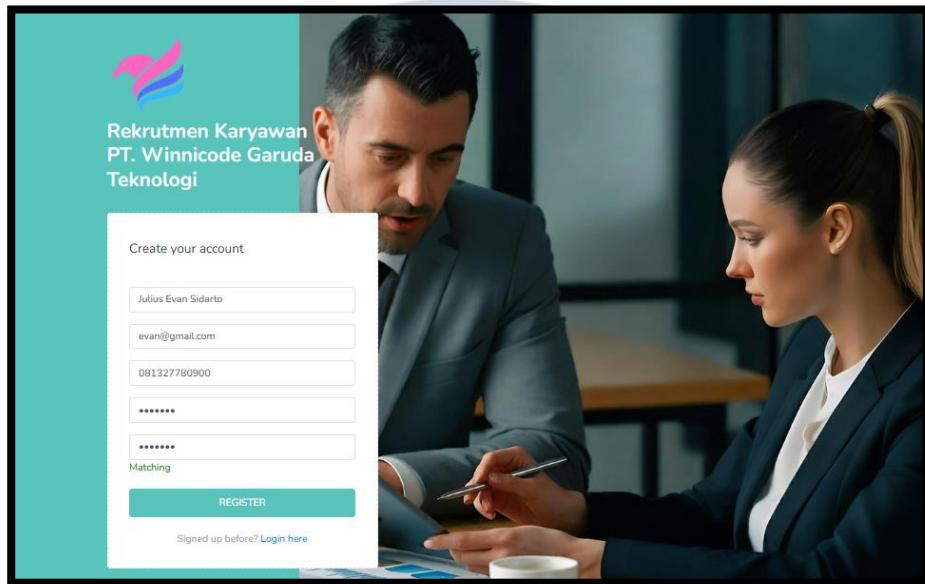
```
20  <body data-theme="light" class="font-nunito">
21  <!-- WAPPER -->
22  <div class="flash-data-success" data-flashdata=<?>= $this->session->flashdata('success'); ?>"></div>
23  <div class="flash-data-info" data-flashdata=<?>= $this->session->flashdata('info'); ?>"></div>
24  <div class="flash-data-error" data-flashdata=<?>= $this->session->flashdata('error'); ?>"></div>
25  <div id="wrapper" class="theme-cyan">
26  <div class="vertical-align-wrap auth-main">
27  <div class="auth-box">
28  <div class="top">
29  <img src=<?>= base_url(); ?>assets/logo.png" alt="Winnicode" style="width: 100%; height: 100%;"/>
30  <h3 class="text-light" style="font-weight:bold; padding-top: 3%;>Rekrutmen Karyawan <br>PT. Winnicode Garuda <br>Teknologi</h3>
31  </div>
32  <div class="card">
33  <div class="flash-data-success" data-flashdata=<?>= $this->session->flashdata('success'); ?>"></div>
34  <div class="flash-data-info" data-flashdata=<?>= $this->session->flashdata('info'); ?>"></div>
35  <div class="flash-data-error" data-flashdata=<?>= $this->session->flashdata('error'); ?>"></div>
36  <div class="header">
37  <p class="lead"><strong>Welcome Back!</strong></p>
38  </div>
39  <div class="body">
40  <form class="form-auth-small" action=<?>= base_url(); ?>Auth/prosesLogin" method="POST">
41  <div class="form-group">
42  <label for="signin-email" class="control-label sr-only">Email</label>
43  <input type="email" class="form-control" id="signin-email" name="email" placeholder="Email">
44  </div>
45  <div class="form-group">
46  <label for="signin-password" class="control-label sr-only">Password</label>
47  <input type="password" class="form-control" id="signin-password" name="password" placeholder="Password">
48  </div>
49  </form>
50  </div>
```

Kode 3.1. Login.php

Kode 3.1 merupakan implementasi halaman *login* pada *website* rekrutmen karyawan yang menggunakan HTML dan PHP. Halaman diawali dengan pengaturan tema tampilan dan jenis *font* pada elemen <body>, kemudian dilengkapi dengan beberapa elemen notifikasi yang memanfaatkan *session flashdata* untuk menampilkan pesan sukses, informasi, maupun kesalahan sesuai hasil proses *login*. Seluruh komponen halaman *login* dibungkus dalam sebuah *container* yang berfungsi mengatur posisi tampilan agar berada di tengah layar. Pada bagian atas halaman ditampilkan logo dan judul sistem sebagai identitas *website*. Bagian utama halaman berupa *form login* yang menggunakan metode POST dan diarahkan ke proses autentikasi pada *server*, yang di dalamnya terdapat input *email* dan *password*. Input *email* digunakan untuk memverifikasi alamat *email* pelamar, sedangkan input *password* disamarkan demi menjaga keamanan data. Setiap *field* dilengkapi label tersembunyi dan *placeholder* untuk meningkatkan aksesibilitas dan kemudahan penggunaan.

B. Halaman Register

Halaman *register* merupakan halaman yang digunakan oleh calon pelamar untuk membuat akun baru pada *website* rekrutmen karyawan PT. Winnicode Garuda Teknologi. Pada halaman ini, pelamar diminta untuk mengisi beberapa data, yaitu nama lengkap, alamat *email*, nomor telepon, *password*, dan konfirmasi *password*. Setelah seluruh data terisi dengan benar, pelamar dapat menekan tombol *Register* untuk menyelesaikan proses pendaftaran akun. *Website* akan melakukan validasi terhadap data yang dimasukkan, termasuk kecocokan *password* dan konfirmasi *password*, sebelum akun berhasil dibuat. Selain itu, halaman *register* juga menyediakan tautan menuju halaman *login* bagi pelamar yang telah memiliki akun. Tampilan halaman *register* dapat dilihat pada Gambar 3.16.



Gambar 3.16. Tampilan Halaman Register

Gambar 3.16 merupakan halaman registrasi akun pada *website* rekrutmen karyawan yang digunakan oleh calon pelamar untuk membuat akun baru. Pengguna diminta mengisi data diri seperti nama, *email*, nomor telepon, dan *password*. Setelah data divalidasi oleh sistem, akun dapat didaftarkan dan digunakan untuk *login* ke dalam sistem rekrutmen.

Jika ada perbedaan antara input di kolom *password* dan konfirmasi *password*, sistem akan menampilkan pesan *error* sebagaimana ditunjukkan pada Gambar 3.17.



Gambar 3.17. Tampilan Password Not Matching

Gambar 3.17 menampilkan pesan kesalahan yang muncul pada halaman pendaftaran ketika *password* dan konfirmasi *password* yang dimasukkan oleh pelamar tidak sama. *Website* secara otomatis melakukan validasi terhadap kedua *field* tersebut untuk memastikan konsistensi data sebelum akun dapat dibuat. Apabila terjadi perbedaan antara *password* dan konfirmasi *password*, *website* akan menampilkan notifikasi “*Not Matching*” sebagai pemberitahuan kepada pelamar agar melakukan pengecekan dan pengisian ulang *password* dengan benar. Mekanisme ini bertujuan untuk meminimalkan kesalahan *input* serta meningkatkan keamanan dan keakuratan data pelamar.

C. Halaman Pengisian Data User

Halaman pengisian data pelamar muncul setelah pelamar berhasil melakukan *register* dan *login* ke dalam *website* rekrutmen. Halaman ini bersifat wajib, artinya pelamar harus menyelesaikan pengisian seluruh data yang diminta sebelum dapat mengakses *dashboard* utama dan melamar lowongan pekerjaan. Pada halaman ini pelamar diminta untuk mengisi informasi pribadi seperti nomor KTP, nama lengkap, status perkawinan, agama, kewarganegaraan, nomor NPWP, jenis dan nomor SIM, serta alamat sesuai KTP. Pengisian data disusun secara bertahap melalui beberapa *tab*, yaitu Data Diri, Pendidikan, Pengalaman, dan Keahlian, sehingga memudahkan pelamar dalam melengkapi informasi secara sistematis, sebagaimana terlihat pada Gambar 3.18, Gambar 3.19, Gambar 3.20, dan Gambar 3.21.

Gambar 3.18. Tampilan Tab Data Diri

Gambar 3.18 menunjukkan tampilan menampilkan form pengisian identitas pelamar yang digunakan sebagai dasar data pada proses seleksi karyawan. Setelah selesai melengkapi data diri, proses pengisian dilanjutkan ke Gambar 3.19.

Gambar 3.19. Tampilan Tab Pendidikan

Gambar 3.19 menggambarkan tampilan tab pendidikan yang berisi *form* informasi riwayat pendidikan pelamar, seperti jenjang pendidikan dan institusi, yang berfungsi sebagai dasar penilaian kualifikasi akademik.

Setelah data pendidikan dilengkapi, pelamar melanjutkan ke tab pengalaman sebagaimana ditunjukkan pada Gambar 3.20.

1. Data Diri	2. Pendidikan	3. Pengalaman	4. Keahlian
<p>Nama Perusahaan Sebelumnya PT. Sejahtera</p> <p>Posisi Pengalaman Kerja Terakhir Fullstack Developer</p> <p>Posisi Jabatan Terakhir Staff</p> <p>Mulai Bekerja di perusahaan sebelumnya February 2025</p> <p>Berakhir Bekerja di perusahaan sebelumnya October 2025</p> <p>Penghasilan Pokok Terakhir (Gaji Perbulan) 10.000.000</p> <p>Penghasilan Tambahan Terakhir (Insentif Perbulan) 2.000.000</p> <p>Fasilitas Terakhir yang didapatkan di perusahaan sebelumnya Makan, Uang Transport</p> <p>Alasan Berhenti dari pekerjaan sebelumnya Sudah habis kontrak</p>			
<p>Previous</p> <p>Next</p>			

Gambar 3.20. Tampilan Tab Pengalaman

Gambar 3.20 menunjukkan tab pengalaman untuk mencatat pengalaman kerja atau pengalaman pendukung lainnya yang relevan dengan posisi yang dilamar. Selanjutnya, pelamar diarahkan ke tab keahlian sebagaimana digambarkan pada Gambar 3.21.

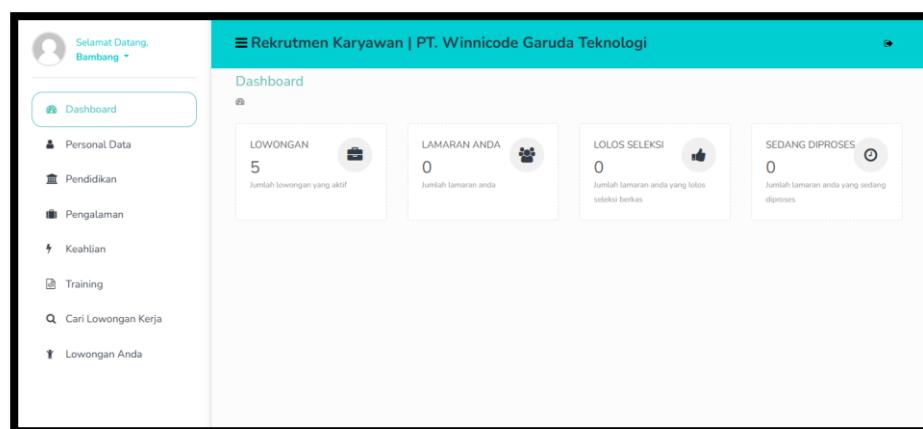
1. Data Diri	2. Pendidikan	3. Pengalaman	4. Keahlian
<p>Keahlian</p> <p>Keahlian Tingkat Keahlian</p> <p>HTML CSS JS PHP Sangat Baik</p> <p>Training</p> <p>Training yang pernah diikuti Sertifikasi yang pernah diikuti Penyelenggara Pelatihan Tahun Pelatihan</p> <p>Bootcamp Fullstack Algorithm Data Analysis Winnicode January 2025</p> <p>Bahasa</p> <p>Bahasa Asing yang dikusai Tingkat Keahlian Bahasa asing</p> <p>Bahasa Inggris Sangat Baik</p>			
<p>Previous</p> <p>Finish</p>			

Gambar 3.21. Tampilan Tab Keahlian

Gambar 3.21 digunakan untuk mencantumkan kemampuan atau keterampilan yang dimiliki sebagai penunjang kompetensi pelamar dalam proses seleksi.

D. Halaman Dashboard User

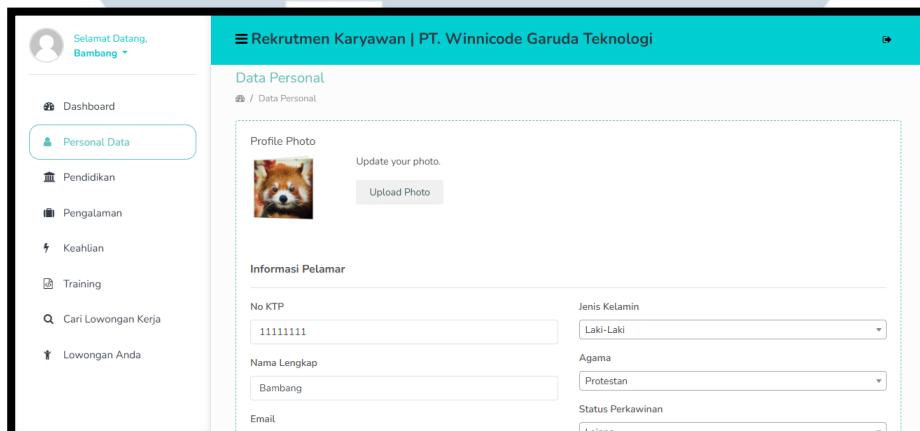
Halaman *dashboard user* merupakan halaman utama yang ditampilkan setelah pelamar berhasil melengkapi semua data yang diperlukan pada *website* rekrutmen karyawan PT. Winnicode Garuda Teknologi. Halaman ini berfungsi sebagai pusat informasi yang memberikan gambaran umum mengenai aktivitas dan fitur untuk pelamar. Pada bagian atas halaman terdapat *header* yang menampilkan nama perusahaan serta informasi akun pelamar yang *sedang login*, sedangkan pada sisi kiri halaman tersedia menu navigasi yang digunakan untuk mengakses berbagai fitur seperti pengelolaan data pribadi, riwayat pendidikan, pengalaman kerja, keahlian, *training*, pencarian lowongan kerja, serta daftar lowongan yang telah dilamar. Selain itu, ketika pelamar berhasil melakukan *login*, *website* akan menampilkan notifikasi di bagian kanan bawah halaman dengan pesan “Anda Berhasil Login!” sebagai penanda bahwa proses *login* telah berhasil dan pelamar telah masuk ke dalam *website*. Tampilan halaman *dashboard user* dapat dilihat pada Gambar 3.22.



Gambar 3.22. Tampilan Halaman Dashboard User

E. Halaman Personal Data

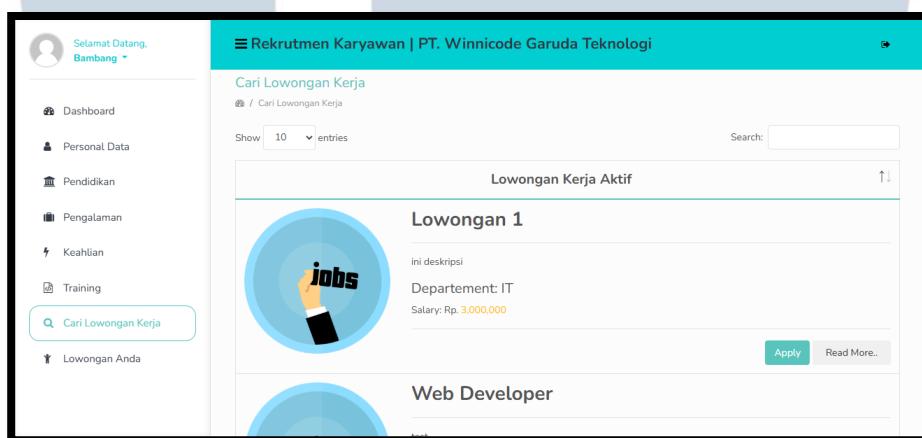
Halaman personal data merupakan halaman yang digunakan oleh pelamar untuk melihat dan mengelola data pribadi yang telah diinput ke dalam sistem rekrutmen. Pada halaman ini ditampilkan identitas pelamar secara lengkap, mulai dari nomor KTP, nama lengkap, jenis kelamin, serta data pribadi lainnya yang berkaitan dengan profil pelamar. Halaman ini juga menyediakan fitur untuk melakukan pembaruan data. Pelamar dapat mengunggah atau memperbarui foto profil, sehingga identitas akun menjadi lebih jelas. Selain itu, pelamar juga dapat memperbarui dokumen CV, serta melakukan perubahan data pribadi lainnya apabila terjadi kesalahan atau pembaruan data. Tampilan halaman personal data dapat dilihat pada Gambar 3.23.



Gambar 3.23. Tampilan Halaman Personal Data

F. Halaman Pencarian Lowongan Kerja

Halaman pencarian merupakan halaman yang digunakan oleh pelamar untuk melihat daftar lowongan kerja yang tersedia pada *website* rekrutmen. Pada halaman ini ditampilkan berbagai informasi lowongan yang sedang aktif, seperti nama posisi, deskripsi singkat, departemen, dan gaji. Untuk melihat informasi lowongan secara lengkap, pelamar dapat menggunakan fitur “*Read More*” yang berfungsi menampilkan detail deskripsi lowongan secara menyeluruh. Tampilan halaman pencarian lowongan pekerjaan dapat dilihat pada Gambar 3.24.



Gambar 3.24. Tampilan Halaman Pencarian Lowongan Kerja

Selain itu, pada setiap lowongan pekerjaan tersedia tombol “*Apply*” yang digunakan oleh pelamar untuk mengajukan lamaran pekerjaan sesuai dengan posisi yang dipilih. Setelah tombol tersebut ditekan, sistem akan memproses pengajuan lamaran dan secara otomatis menyimpan data lowongan yang dilamar ke dalam daftar lowongan pelamar.

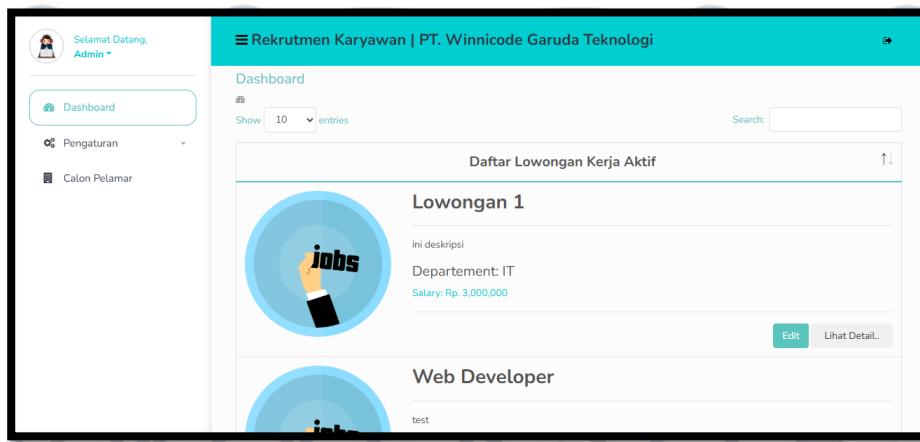
Daftar lowongan ini berfungsi sebagai informasi riwayat lamaran yang telah diajukan dan ditampilkan pada halaman daftar lowongan, sebagaimana ditunjukkan pada Gambar 3.25.



Gambar 3.25. Tampilan Halaman Daftar Lowongan

G. Halaman Dashboard Admin

Halaman *dashboard admin* merupakan halaman utama yang digunakan oleh *admin* perusahaan dalam mengelola dan memantau lowongan pekerjaan yang sedang dibuka. Pada halaman ini, *admin* dapat melihat daftar lowongan yang sedang aktif, dilengkapi dengan fitur pencarian dan pengaturan jumlah data yang ditampilkan guna memudahkan *admin* dalam menemukan lowongan tertentu. Tampilan halaman *dashboard admin* ditampilkan pada Gambar 3.26.



Gambar 3.26. Tampilan Halaman Dashboard Admin

Setiap lowongan dilengkapi dengan ikon visual serta tombol *action* seperti “Edit” untuk mengubah data lowongan dan “Lihat Detail” untuk

melihat informasi lowongan lebih lengkap. Selain itu, melalui menu navigasi di sisi kiri, *admin* dapat mengakses halaman lain seperti pengaturan data pelamar, departemen, jabatan, serta lowongan.

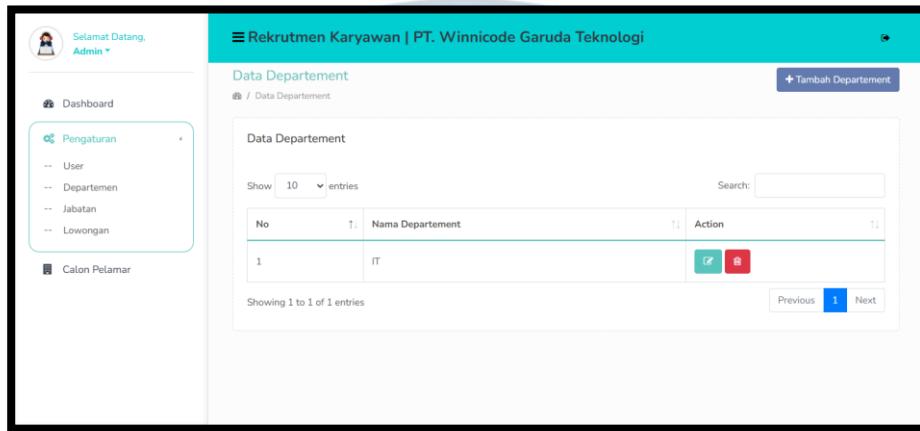
H. Halaman Pengaturan Admin

Halaman pengaturan *admin* merupakan halaman yang digunakan untuk mengelola data dalam *website* rekrutmen karyawan, diantaranya pengaturan untuk *user*, departemen, jabatan, dan lowongan. Pada halaman ini, *admin* memiliki hak akses penuh untuk menambah, mengubah, dan menghapus data yang berkaitan dengan kebutuhan rekrutmen. Tampilan halaman pengaturan *admin* yang meliputi *user*, departemen, jabatan, dan lowongan ditunjukkan pada Gambar 3.27, Gambar 3.28, Gambar 3.29, dan Gambar 3.30.

No	Foto	Nama	Hak Akses	Email	No Telp	Action
1		p	User	p@gmail.com	1234	
2		Admin	Admin	admin@gmail.com	123456	
3		Manager	Manager	manager@gmail.com	12345	
4		Supervisor	Supervisor	supervisor@gmail.com	12345	

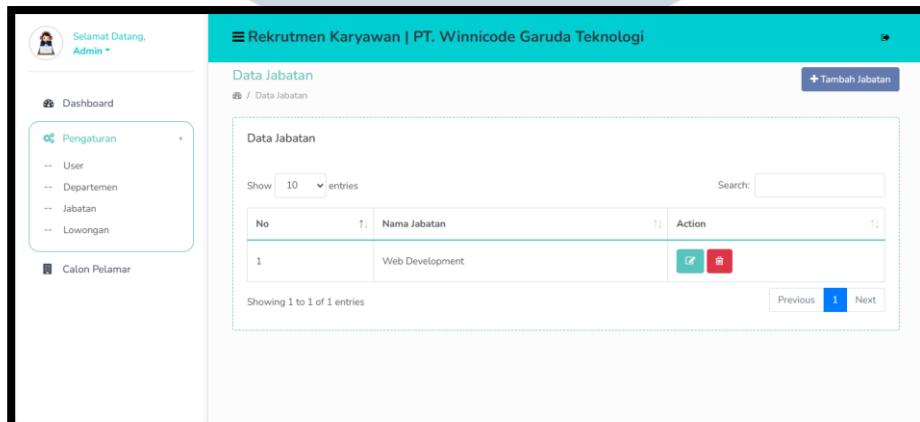
Gambar 3.27. Tampilan Pengaturan Admin (User)

Gambar 3.27 digunakan oleh *admin* untuk mengelola data pengguna sistem. Pada halaman ini, *admin* dapat melihat daftar *user* yang terdaftar beserta informasi seperti nama, foto, hak akses, *email*, dan nomor telepon. Selain itu, *admin* juga dapat melakukan *action* seperti penambahan, pengubahan, dan penghapusan data user, sehingga pengelolaan hak akses sistem dapat dilakukan secara terstruktur.



Gambar 3.28. Tampilan Pengaturan Admin (Departemen)

Gambar 3.28 digunakan *admin* untuk mengelola data departemen yang ada di perusahaan. Data departemen ini berfungsi sebagai pengelompokan unit kerja dan menjadi acuan dalam penentuan jabatan serta lowongan pekerjaan.



Gambar 3.29. Tampilan Pengaturan Admin (Jabatan)

Gambar 3.29 digunakan *admin* untuk mengelola data jabatan yang tersedia pada setiap departemen. Informasi jabatan ini digunakan dalam proses pembuatan lowongan kerja serta sebagai referensi dalam penempatan dan penilaian pelamar.

Jabatan	Maks Usia	Minimal Pendidikan	Tanggal Close	Status	Soal	Psikotes	Action
eb Development	24	D4/S1	31/12/2027	Aktif	Lihat Soal	Lihat Psikotes	Edit Hapus
eb Development	23	D3	31/07/2022	Tidak Aktif	Lihat Soal	Lihat Psikotes	Edit Hapus
eb Development	22	SMA	11/12/2026	Aktif	Lihat Soal	Lihat Psikotes	Edit Hapus

Gambar 3.30. Tampilan Pengaturan Admin (Lowongan)

Gambar 3.30 digunakan oleh *admin* untuk mengelola data lowongan pekerjaan yang akan dibuka. Pada halaman ini, *admin* dapat menambahkan, mengubah, dan menghapus informasi lowongan. Selain itu, *admin* dapat menambah ataupun mengubah soal perusahaan dan juga soal psikotes.

I. Halaman Psikotes

Halaman psikotes merupakan halaman yang digunakan oleh *admin* untuk mengelola soal-soal psikotes yang akan diberikan kepada calon pelamar dalam proses rekrutmen. Pada halaman ini, *admin* dapat menambahkan soal psikotes baru dengan menginput pertanyaan beserta beberapa pilihan jawaban, seperti opsi A, B, C, D, serta menentukan kunci jawaban yang benar. Selain menambahkan soal, halaman ini juga menampilkan daftar soal psikotes yang telah dibuat sebelumnya lengkap dengan detail pertanyaan, pilihan jawaban, dan kunci jawaban, sebagaimana terlihat pada Gambar 3.31 dan Gambar 3.32.

Selamat Datang, Admin

Tambah Soal Psikotes

Tambah Data Soal Psikotes

No	SOAL TEST
Pertanyaan :	ini pertanyaan
A.	ini a
B.	ini b
C.	ini c
D.	ini d

Gambar 3.31. Tampilan Fitur Tambah Psikotes

Gambar 3.31 merupakan fitur tambah soal psikotes yang digunakan oleh *admin* untuk menambahkan data psikotes ke dalam sistem. Pada halaman ini, *admin* dapat memasukkan informasi psikotes seperti pertanyaan, kunci jawaban, serta mengatur banyaknya jumlah pertanyaan.

Selamat Datang, Admin

Details Soal Psikotes

Anda dapat menambah, mengedit, atau menghapus soal psikotes *

No	SOAL TEST
Pertanyaan : test pertanyaan 131	ini pilihan a13
1	ini pilihan b13
	ini pilihan c13
	ini pilihan d13

Kunci JawabanA

Gambar 3.32. Tampilan Fitur Lihat Psikotes

Gambar 3.32 menunjukkan daftar soal psikotes yang telah dibuat. Melalui halaman ini, *admin* dapat melihat informasi psikotes yang tersedia serta melakukan perbaruan data. Psikotes yang tersedia pada halaman ini akan digunakan dalam proses seleksi dan secara otomatis dapat diakses oleh pelamar, sehingga soal-soal psikotes yang ditampilkan akan dikerjakan oleh pelamar sesuai dengan tahapan seleksi yang ditentukan.

J. Halaman Data Calon Pelamar

Halaman Data Calon Pelamar merupakan halaman yang digunakan oleh *admin* untuk melihat dan mengelola data pelamar yang telah mendaftar pada lowongan kerja. Pada halaman ini ditampilkan daftar calon pelamar dalam bentuk tabel yang memuat informasi penting, seperti nama lowongan yang dilamar, nama pelamar, alamat *email*, nomor telepon, status lamaran, serta tanggal pengajuan lamaran. Sistem juga memberikan penanda warna biru pada data pelamar yang belum diberikan respons oleh *admin*, sehingga memudahkan dalam proses *monitoring* dan tindak lanjut. Selain itu, tersedia fitur pencarian dan pengaturan jumlah data yang ditampilkan untuk membantu *admin* menemukan data pelamar secara cepat. *Admin* dapat melihat informasi lebih lengkap mengenai pelamar melalui tombol detail, serta melakukan *approve* atau *reject* terhadap calon pelamar berdasarkan hasil seleksi, apakah pelamar memenuhi kriteria yang ditentukan atau tidak, sebagaimana ditunjukkan pada Gambar 3.33.

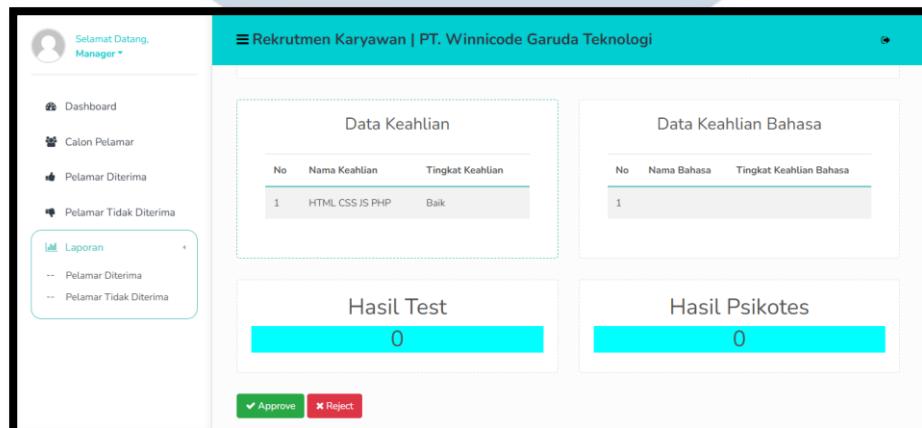
No	Nama Lowongan	Nama Pelamar	Email	Hp	Status Lamaran	Tanggal Apply	Action
1	Web Developer	Julius Evan Sidarto	evan@gmail.com	1111111	Diterima	20, Dec 2025	<button>Detail</button>
2	Lowongan 1	Julius Evan Sidarto	evan@gmail.com	1111111	Sedang Di Proses	24, Dec 2025	<button>Detail</button>
3	Lowongan 1	Bambang	bambang@gmail.com	9999999	Menunggu	26, Dec 2025	<button>Detail</button>

Gambar 3.33. Tampilan Data Calon Pelamar

UNIVERSITAS
MULTIMEDIA
NUSANTARA

K. Halaman Seleksi Pelamar

Halaman ini merupakan halaman yang digunakan oleh *manager* untuk meninjau hasil seleksi calon pelamar yang sebelumnya telah diproses oleh *admin*. Setelah *admin* melakukan seleksi awal, termasuk pemeriksaan data dan hasil tes serta psikotes, data pelamar akan diteruskan ke *manager* untuk dilakukan penilaian lanjutan. Pada halaman ini, *manager* dapat melihat informasi pelamar secara lengkap beserta status lamaran dan hasil seleksi yang telah dilakukan, kemudian mengambil keputusan akhir dengan menerima atau menolak calon pelamar berdasarkan kesesuaian hasil tes dan psikotes dengan kriteria yang dibutuhkan perusahaan. Halaman ini berperan penting dalam pengambilan keputusan akhir pada proses rekrutmen agar pelamar yang diterima benar-benar sesuai dengan kebutuhan perusahaan. Tampilan halaman seleksi pelamar oleh *manager* dapat dilihat pada Gambar 3.34.



Gambar 3.34. Tampilan Halaman Seleksi Pelamar

UNIVERSITAS
MULTIMEDIA
NUSANTARA

L. Halaman Laporan Pelamar

Halaman Laporan Pelamar merupakan halaman yang hanya dapat diakses oleh *manager* dan berfungsi untuk melihat laporan hasil proses rekrutmen, baik pelamar yang diterima maupun pelamar yang tidak diterima. Pada halaman ini, manager dapat melakukan penyaringan data pelamar berdasarkan rentang tanggal tertentu, sehingga memudahkan dalam pencarian dan peninjauan data, terutama ketika jumlah pelamar yang masuk cukup banyak. Fitur pencarian berdasarkan tanggal memungkinkan manager untuk memperoleh laporan secara lebih cepat, akurat, dan terstruktur. Tampilan halaman laporan pelamar dapat dilihat pada Gambar 3.35 dan Gambar 3.36.



Gambar 3.35. Tampilan Laporan Pelamar Diterima

Gambar 3.35 menampilkan daftar pelamar yang telah dinyatakan lulus seluruh tahapan seleksi dan daftar ini untuk memudahkan *admin* dalam meninjau dan memantau data pelamar yang diterima, terutama ketika jumlah pelamar yang masuk cukup banyak.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

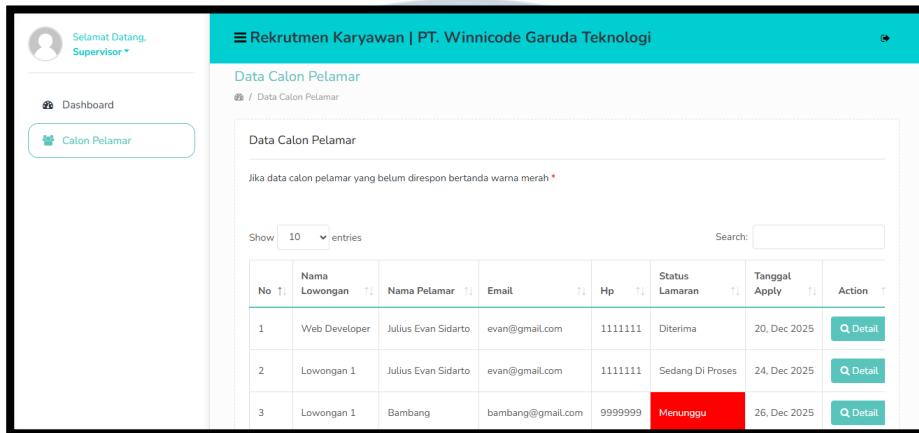


Gambar 3.36. Tampilan Laporan Data Tidak Diterima

Gambar 3.36 digunakan untuk menampilkan daftar pelamar yang tidak lolos dalam proses rekrutmen. Dengan adanya pemisahan data ini, *admin* dapat dengan mudah melakukan peninjauan dan evaluasi terhadap data pelamar yang tidak diterima secara lebih cepat dan efisien, khususnya pada kondisi jumlah data pelamar yang besar.

M. Halaman Supervisor

Halaman *supervisor* merupakan halaman yang digunakan oleh *supervisor* untuk melakukan pemantauan terhadap proses rekrutmen yang sedang berjalan tanpa memiliki hak akses untuk melakukan perubahan data. Pada halaman ini, *supervisor* dapat melihat informasi terkait data calon pelamar, serta status lamaran yang sedang diproses sebagai bahan *monitoring* dan evaluasi. Seluruh informasi yang ditampilkan bersifat *read-only*, sehingga *supervisor* hanya berperan dalam mengawasi dan memastikan proses rekrutmen berjalan sesuai dengan prosedur yang telah ditetapkan. Tampilan halaman *supervisor* dapat dilihat pada Gambar 3.37.



Gambar 3.37. Tampilan Halaman Supervisor

Gambar 3.37 menampilkan *list* data pelamar yang digunakan untuk memantau proses rekrutmen. Halaman ini bertujuan untuk memudahkan *supervisor* dalam melakukan pemantauan dan pengawasan terhadap data pelamar, terutama ketika jumlah pelamar yang masuk cukup banyak.

Berdasarkan tampilan halaman *supervisor* yang telah dijelaskan sebelumnya, berikut ditunjukkan potongan kode program yang digunakan dalam implementasinya pada Kode 3.2.

```
class Supervisor extends CI_Controller
{
    function __construct()
    {
        parent::__construct();
        $this->load->model('AuthModel');
        $this->load->model('AdminModel');
        $this->AuthModel->cekLoginSupervisor();
    }

    public function index()
    {
        $where_calon['status_hasil'] = 'Sedang Di Proses';
        $where_diterima['status_hasil'] = 'Diterima';
        $where_tidak['status_hasil'] = 'Tidak Diterima';
        $data = [
            'lowongan' => $this->AdminModel->total_lowongan('lowongan'),
            'calon_pelamar' => $this->AdminModel->total_pelamar('apply_job', $where_calon),
            'pelamar_diterima' => $this->AdminModel->total_pelamar('apply_job', $where_diterima),
            'pelamar_tidak_terima' => $this->AdminModel->total_pelamar('apply_job', $where_tidak),
        ];
        $this->template->load('layout/layout', 'supervisor/dashboard', $data);
    }

    public function Calon_Pelamar()
    {
        $database = 'apply_job';
        $database1 = 'pelamar';
        $database2 = 'lowongan';
    }
}
```

Kode 3.2. Supervisor.php

Kode 3.2 merupakan *controller* *Supervisor* pada *framework* *CodeIgniter* yang berfungsi mengatur logika halaman *supervisor* dalam

sistem rekrutmen karyawan. *Class Supervisor* mewarisi *CI_Controller*, sehingga mengikuti pola arsitektur MVC. Pada fungsi *__construct()*, sistem memanggil konstruktor induk, memuat *AuthModel* untuk keperluan autentikasi serta *AdminModel* untuk pengolahan data rekrutmen, kemudian melakukan pengecekan akses melalui fungsi *cekLoginSupervisor()* agar hanya pengguna dengan peran *supervisor* yang dapat mengakses halaman ini.

Fungsi *index()* berperan sebagai halaman *dashboard supervisor* yang menampilkan data *monitoring* rekrutmen. Pada bagian ini, sistem menetapkan kondisi berdasarkan status hasil pelamar, yaitu *Sedang Di Proses*, *Diterima*, dan *Tidak Diterima*. Selanjutnya, sistem mengambil data lowongan pekerjaan serta menghitung jumlah pelamar pada setiap status menggunakan fungsi *total_pelamar()* dari *AdminModel*. Seluruh data tersebut dikirimkan ke tampilan *supervisor/dashboard* melalui template utama, sehingga *supervisor* dapat memantau perkembangan rekrutmen tanpa memiliki kewenangan untuk melakukan seleksi atau persetujuan pelamar.

N. Halaman Backend

Halaman *backend* merupakan bagian dari sistem yang digunakan untuk mengelola seluruh data dan proses pada *website* rekrutmen karyawan. *Backend* berperan sebagai penghubung antara antarmuka pengguna dengan basis data, sehingga seluruh aktivitas dapat berjalan dengan baik. Pada sistem ini, *backend* dibangun menggunakan bahasa pemrograman PHP dengan *framework* *CodeIgniter* serta didukung oleh *database* MySQL sebagai media penyimpanan data. Struktur *database* dapat dilihat pada Gambar 3.38.

Table	Action	Rows	Type	Collation	Size	Overhead
apply_job		5	InnoDB	latin1_swedish_ci	16.0 Kib	-
bahasa		5	InnoDB	latin1_swedish_ci	16.0 Kib	-
departemen		1	InnoDB	latin1_swedish_ci	16.0 Kib	-
jabatan		1	InnoDB	latin1_swedish_ci	16.0 Kib	-
keahlian		8	InnoDB	latin1_swedish_ci	16.0 Kib	-
lowongan		5	InnoDB	latin1_swedish_ci	16.0 Kib	-
pelamar		3	InnoDB	latin1_swedish_ci	16.0 Kib	-
pendidikan		3	InnoDB	latin1_swedish_ci	16.0 Kib	-
pengalaman		3	InnoDB	latin1_swedish_ci	16.0 Kib	-
psikotes		1	InnoDB	latin1_swedish_ci	16.0 Kib	-
soal		8	InnoDB	latin1_swedish_ci	16.0 Kib	-
training		3	InnoDB	latin1_swedish_ci	16.0 Kib	-
user		6	InnoDB	latin1_swedish_ci	16.0 Kib	-
13 tables	Sum	52	InnoDB	utf8mb4_general_ci	208.0 Kib	0 B

Gambar 3.38. Struktur Database Rekrutmen Karyawan

Gambar 3.38 memiliki fungsi dan peran masing-masing dalam menyimpan serta mengelola data yang diperlukan.

1. Tabel *user* digunakan untuk menyimpan data akun pengguna yang mengakses sistem, termasuk informasi *login* dan peran pengguna seperti *admin, supervisor, manager*, maupun pelamar.
2. Tabel pelamar berisi data detail calon pelamar, seperti identitas pribadi yang akan digunakan selama proses rekrutmen.
3. Tabel *apply_job* berfungsi untuk mencatat data lamaran pekerjaan yang diajukan oleh pelamar, termasuk status hasil seleksi yang sedang berjalan.
4. Tabel *lowongan* menyimpan informasi mengenai lowongan pekerjaan yang tersedia.
5. Tabel *jabatan* dan *departemen* digunakan untuk mengelompokkan posisi pekerjaan berdasarkan jabatan dan departemen tertentu.
6. Tabel *keahlian* berfungsi untuk menyimpan data keterampilan yang dimiliki oleh pelamar sebagai pendukung kualifikasi.
7. Tabel *pendidikan* dan *pengalaman* digunakan untuk menyimpan riwayat pendidikan serta pengalaman kerja pelamar.
8. Tabel *bahasa* mencatat kemampuan bahasa yang dimiliki pelamar.
9. Tabel *training* menyimpan informasi mengenai pelatihan atau sertifikasi yang pernah diikuti oleh pelamar.

10. Tabel psikotes digunakan untuk menyimpan data hasil psikotes pelamar.
11. Tabel soal berisi kumpulan soal yang digunakan dalam pelaksanaan tes psikotes.

Adapun konfigurasi koneksi *database* yang digunakan pada *website* rekrutmen karyawan ditunjukkan melalui potongan kode pada file database.php, sebagaimana ditunjukkan pada Kode 3.3.

```
$active_group = 'default';
$query_builder = TRUE;

✓ $db['default'] = array(
    'dsn' => '',
    'hostname' => 'localhost',
    'username' => 'root',
    'password' => '',
    'database' => 'rekrutmen',
    'dbdriver' => 'mysqli',
    'dbprefix' => '',
    'pconnect' => FALSE,
    'db_debug' => (ENVIRONMENT !== 'production'),
    'cache_on' => FALSE,
    'cachedir' => '',
    'char_set' => 'utf8',
    'dbcollat' => 'utf8_general_ci',
    'swap_pre' => '',
    'encrypt' => FALSE,
    'compress' => FALSE,
    'stricton' => FALSE,
    'failover' => array(),
    'save_queries' => TRUE
);
```

Kode 3.3. Database.php

Kode 3.3 merupakan database.php, yang dimana konfigurasi koneksi menggunakan *hostname* *localhost* dengan *username* *root* tanpa *password* dan nama *database* *rekrutmen*. *Driver* *mysqli* digunakan untuk memastikan koneksi *database* berjalan stabil. Selain itu, pengaturan *db_debug* disesuaikan dengan kondisi *environment* agar pesan kesalahan hanya ditampilkan pada tahap pengembangan. Sistem juga menggunakan karakter set *utf8* dan *collation* *utf8_general_ci* untuk mendukung penyimpanan data dengan berbagai karakter. Beberapa fitur tambahan

seperti *cache*, enkripsi, dan kompresi dinonaktifkan karena tidak diperlukan, sementara opsi *save_queries* diaktifkan untuk membantu proses *debugging* selama pengembangan *website*.

3.6 Kendala dan Solusi

Selama proses penggerjaan *website* rekrutmen karyawan, terdapat beberapa kendala yang terjadi beserta solusi yang dapat dilakukan.

1. Kendala

1. Kurang optimalnya komunikasi dengan *supervisor*, khususnya ketika menyampaikan kebutuhan atau permintaan dari pihak kampus terkait pelaksanaan magang. *Supervisor* sering kali memberikan respons yang lambat dan sulit dihubungi dalam waktu tertentu, sehingga proses penyampaian informasi, pertanyaan terkait fitur rekrutmen, dan konfirmasi kebutuhan menjadi terhambat. Kondisi ini berdampak pada keterlambatan pengambilan keputusan dan memengaruhi kelancaran proses penggerjaan *website* secara keseluruhan.
2. Tidak adanya pertemuan rutin atau sesi diskusi secara daring menggunakan *platform* seperti *Zoom* atau *Google Meet*. Tanpa adanya pertemuan tatap muka secara virtual, pembahasan terkait progres perancangan dan pembangunan, evaluasi hasil kerja, serta diskusi solusi atas kendala teknis menjadi kurang efektif. Akibatnya, proses konfirmasi dan pengambilan keputusan terkait pengembangan *website* rekrutmen karyawan memerlukan waktu yang lebih lama.
3. Penggunaan *framework* baru, yaitu *CodeIgniter*, yang sebelumnya belum pernah dipelajari secara mendalam. Kurangnya pengalaman dan pemahaman awal terhadap struktur *framework*, alur kerja *backend*, serta pola yang digunakan menyebabkan proses adaptasi dalam penggerjaan *website* menjadi lebih lama. Kondisi ini berdampak pada waktu penggerjaan,

khususnya saat mengimplementasikan fitur pengelolaan data pelamar, lowongan pekerjaan, serta proses seleksi pada *website* rekrutmen karyawan.

2. Solusi

1. Dilakukan upaya komunikasi yang lebih terstruktur dan proaktif dengan menyampaikan kebutuhan dari pihak kampus serta pertanyaan terkait pengembangan *website* rekrutmen karyawan secara ringkas dan jelas melalui media komunikasi yang tersedia. Selain itu, pengiriman pengingat secara berkala dilakukan dengan tetap menjaga etika komunikasi agar proses penyampaian informasi dan konfirmasi kebutuhan dapat berjalan lebih efektif meskipun respons tidak selalu diperoleh secara cepat.
2. Mengompensasi keterbatasan tersebut melalui dokumentasi progres kerja secara mandiri, seperti mencatat hasil pengembangan, kendala teknis, serta solusi yang telah diterapkan. Dokumentasi ini kemudian digunakan sebagai bahan diskusi ketika komunikasi dengan *supervisor* dapat dilakukan, sehingga proses konfirmasi dan pengambilan keputusan dapat berlangsung lebih terarah dan efisien.
3. Dilakukan pembelajaran secara mandiri melalui dokumentasi resmi, tutorial daring, serta forum diskusi pemrograman. Selain itu, penerapan metode *trial and error* juga dilakukan untuk memahami struktur *framework*, alur kerja *backend*, dan pola pengembangan yang digunakan.

UNIVERSITAS
MULTIMEDIA
NUSANTARA