

Spesifikasi Tugas Besar Milestone 1

IF3110 Web Based Development:

LinkInPurry

Tim Asisten Lab Pemrograman 2021

Versi : 1

Tgl. Revisi Terakhir : 12 Oktober 2024

Deadline : 24 Oktober 2024, 21.00 WIB

Revisi 12 Oktober 2024: Skema basis data

Revisi 13 Oktober 2024: Aturan library dan docker, hapus spesifikasi penyimpanan di cloud

Daftar Isi

Daftar Isi	2
Deskripsi Persoalan	3
Tujuan Pembelajaran	5
Spesifikasi Sistem	6
Spesifikasi Umum	6
User Story	10
Spesifikasi Fitur	11
Spesifikasi Tiap Halaman	16
1. Halaman Login	16
2. Halaman Register	16
3. Halaman Home (Job Seeker)	17
4. Halaman Home (Company)	17
5. Halaman Tambah Lowongan (Company)	17
6. Halaman Detail Lowongan (Company)	18
7. Halaman Detail Lamaran (Company)	18
8. Halaman Edit Lowongan (Company)	18
9. Halaman Detail Lowongan (Job Seeker)	19
10. Halaman Lamaran (Job Seeker)	19
11. Halaman Riwayat (Job Seeker)	19
12. Halaman Profil (Company)	19
Spesifikasi Bonus	20
1. All Responsive Web Design	20
2. UI/UX seperti LinkedIn	20
3. Data Export	20
4. Simple Job Recommendation	21
5. Google Lighthouse	21
Keyword & Tips	23
Lain-Lain	25
Responsi	25
Daftar Pertanyaan	25
Deliverables	26
Pengumpulan Tugas	27
Pembagian Kelompok	27

Pembagian Tugas	27
Catatan Tambahan	28

Deskripsi Persoalan



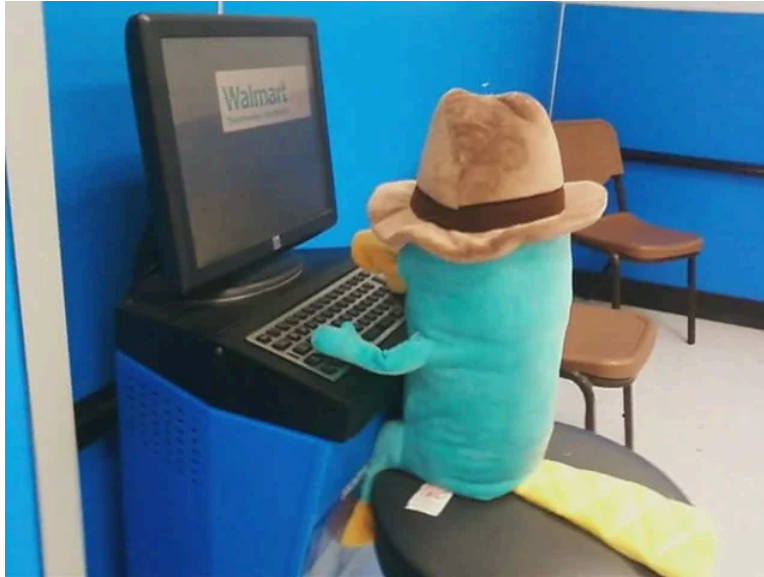
"Purry The Platypus is on a job hunt. Because even secret agents need to pay their bills!"

Di tengah liburan musim panas, Purry si Platypus, yang dikenal sebagai Agent P, menyadari bahwa dia dan banyak agen O.W.C.A. (Organisasi Warga Cool Abiez) lainnya membutuhkan pekerjaan. Meskipun para agen memiliki berbagai keahlian dan spesialisasi, mereka kesulitan menemukan informasi tentang kesempatan untuk mendapatkan pengalaman tersebut. Hal ini membuat Purry merasa bahwa ada kebutuhan mendesak untuk sebuah platform yang dapat menghubungkan para agen dengan peluang yang sesuai untuk mereka.

Menyadari tantangan ini, Purry mengambil inisiatif untuk menciptakan aplikasi yang memungkinkan para agen untuk mencari lowongan kerja. Platform ini diharapkan dapat membantu para agen menemukan pekerjaan yang sesuai dengan keahlian mereka serta memberikan kemudahan dalam mengakses informasi lowongan yang ada. Platform ini akan diberi nama **LinkinPurry**.

Untuk mewujudkan platform ini, Purry tahu bahwa dia memerlukan bantuan dari tim pengembang yang handal. Dia berharap bisa bekerja sama dengan kalian untuk merancang dan mengembangkan aplikasi web yang memiliki fitur-fitur penting yang membantu dalam proses perekrutan kerja.

Selama proses pengembangan, kalian akan menghadapi berbagai tantangan, seperti merancang antarmuka pengguna yang intuitif, memastikan keamanan data pengguna, dan mengoptimalkan kinerja aplikasi agar dapat digunakan oleh banyak agen secara bersamaan. Dengan kerja sama dan dedikasi, kalian semua dapat membantu Purry dan agen-agen O.W.C.A. lainnya menemukan peluang yang mereka butuhkan untuk melanjutkan karir mereka.

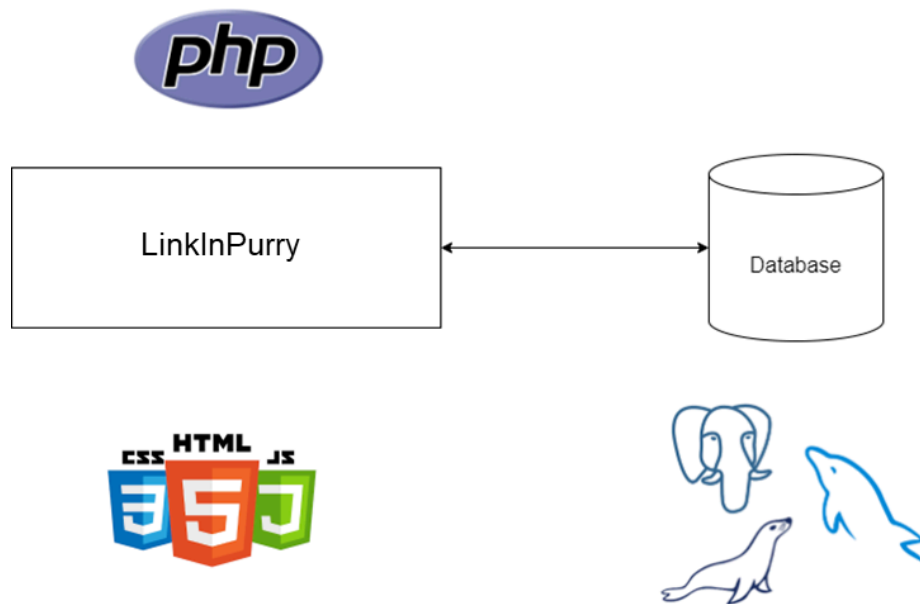


"Even Purry The Platypus is looking for a job"

Tujuan Pembelajaran

- Mahasiswa mampu membuat sebuah aplikasi web dengan menggunakan HTML, CSS, dan JS.
- Mahasiswa mampu membuat sebuah layanan web dengan menggunakan PHP.
- Mahasiswa mampu menggunakan AJAX untuk melakukan pemanggilan yang bersifat *asynchronous*.
- Mahasiswa memahami penanganan *state* dan perbedaan HTTP *method*.

Spesifikasi Sistem



Gambaran Umum untuk Milestone 1

Spesifikasi Umum

Berikut ada beberapa ketentuan umum yang harus dipenuhi oleh sistem.

1. Untuk *client-side*, wajib menggunakan **JavaScript**, **HTML**, dan **CSS** secara **murni**.

Tidak diperbolehkan menggunakan *framework* CSS atau *framework* JS (e.g. jQuery, lodash, Bootstrap, atau Tailwind). CSS harus berada di berkas yang berbeda dengan HTML (gunakan *CSS selector*). Tidak diperbolehkan **memanipulasi** CSS (mengubah isi file CSS programmatically atau dengan JS).

2. Untuk *server-side*, wajib menggunakan **PHP murni** tanpa kakas apapun (seperti Laravel, Codeigniter). Akan tetapi, kalian bisa mengambil inspirasi dari struktur *project* pada Laravel dan sebagainya. Kalian harus mengimplementasikan fitur menggunakan **HTTP method** yang tepat dan sesuai *best practices*.

3. Secara default, **tidak boleh** menggunakan library apapun selain untuk *query database* dan *rich text editor*. Jika ada library yang ingin digunakan, silahkan tanyakan melalui QnA untuk dipertimbangkan.
4. Dalam melakukan request ke *server*, kalian dapat menggunakan [basic form handling](#) ataupun AJAX, dengan **setidaknya tiga** penggunaan pada masing-masing tipe.
5. Untuk basis data, wajib menggunakan MySQL, MariaDB, atau PostgreSQL (*Relational Database*). Tidak diperbolehkan menggunakan ORM atau sejenisnya dan **tidak boleh menyimpan blob atau binary file pada basis data**. Skema yang diberikan di bawah ini dapat dijadikan panduan (**tidak wajib sama**). Kalian dapat menyesuaikan dan mengoptimalkan skema tersebut sesuai kebutuhan.

User

Nama	Tipe	Keterangan
user_id	int	
email	string	
password	string	
role	enum	jobseeker, company
nama	string	

Company Detail

Nama	Tipe	Keterangan
user_id	int	Merujuk ke user_id dari tabel User
lokasi	string	
about	string	

Lowongan

Nama	Tipe	Keterangan
lowongan_id	int	
company_id	int	Merujuk ke user_id dari tabel User
posisi	string	
deskripsi	string	
jenis_pekerjaan	string	
jenis_lokasi	enum	on-site, hybrid, remote
is_open	boolean	Default true
created_at	datetime	
updated_at	datetime	

Attachment Lowongan

Nama	Tipe	Keterangan
attachment_id	int	
lowongan_id	int	Merujuk ke lowongan_id dari tabel Lowongan
file_path	string	

Lamaran

Nama	Tipe	Keterangan
lamaran_id	int	
user_id	int	Merujuk ke user_id dari

		tabel User
lowongan_id	int	Merujuk ke lowongan_id dari tabel Lowongan
cv_path	string	
video_path	string	
status	enum	accepted, rejected, waiting
status_reason	string	
created_at	datetime	

User Story

1. Autentikasi
 - a. Pengguna dapat dibedakan menjadi **2 jenis**: *job seeker* dan *company*.
 - b. Pengguna harus melakukan autentikasi untuk dapat mengakses seluruh fitur, **kecuali** disebutkan fitur dapat diakses oleh pengguna yang tidak terautentikasi.
 - c. Seluruh pengguna yang terautentikasi juga harus dapat melakukan **logout**.
2. Pengguna dapat mendaftar sebagai *company* **atau** *job seeker*
3. *Company* dapat mengubah profil perusahaan
4. *Company* dapat melihat daftar lowongan kerja yang dibuka olehnya
5. *Company* dapat membuat lowongan kerja baru
6. *Company* dapat mengubah lowongan kerja yang sudah ada
7. *Company* dapat menghapus lowongan kerja
8. *Company* dapat menutup lowongan kerja
9. *Company* dapat melihat daftar pelamar untuk suatu lowongan kerja
10. *Company* dapat menerima atau menolak pelamar pada suatu lowongan kerja beserta alasan atau langkah selanjutnya
11. *Job seeker* dapat melihat daftar lowongan yang ada
12. *Job seeker* dapat mencari lowongan
13. *Job seeker* dapat memfilter lowongan berdasarkan jenis pekerjaan dan tipe lokasi
14. *Job seeker* dapat menyortir lowongan berdasarkan waktu *posting*
15. *Job seeker* dapat melamar pekerjaan
16. *Job seeker* dapat melihat riwayat lamaran kerjanya
17. Pengguna yang tidak terautentikasi dapat melihat daftar lowongan seperti yang dilihat *job seeker* tanpa bisa melakukan aksi apapun

Spesifikasi Fitur

Berikut ada beberapa ketentuan terkait fitur-fitur apa saja yang ada di dalam sistem.

1. Autentikasi pengguna

Dalam melakukan autentikasi, pengguna perlu melakukan aksi login terlebih dahulu. Setelah sukses melakukan login, identitas pengguna akan disimpan dalam browser. Bentuk penyimpanan identitas pengguna dibebaskan kepada mahasiswa.

- Untuk mengetahui pengguna mana yang sedang login, identitas ini dapat dicek di basis data.
- Identitas ini tidak boleh disimpan sebagai parameter HTTP GET.
- Jika identitas ini tidak ada, pengguna dianggap belum login dan diarahkan kembali ke halaman login dari halaman yang sedang dibukanya.
- Ketentuan sebelumnya tidak berlaku untuk halaman yang dapat diakses tanpa autentikasi, yaitu [Halaman Home \(Job Seeker\)](#) dan [Halaman Detail Lowongan \(Job Seeker\)](#).
- Masa berlaku identitas pengguna dibebaskan.

2. Navigation bar

Terdapat *navigation bar* di **setiap halaman** yang dibuat (untuk halaman login atau halaman register boleh untuk tidak menggunakan *navigation bar*).

Setidaknya isi dari *navigation bar* adalah sebagai berikut:

- Pengguna yang tidak terautentikasi: [Halaman Home \(Job Seeker\)](#), [Halaman Login](#), dan [Halaman Register](#)
- *Company*: [Halaman Home \(Company\)](#) dan [Halaman Profile \(Company\)](#)
- *Job seeker*: [Halaman Home \(Job Seeker\)](#) dan [Halaman Riwayat \(Job Seeker\)](#)

3. Pengelolaan lowongan pekerjaan

- *Company* dapat:

- **Membuat lowongan kerja** yang setidaknya memuat informasi berikut ini:
 1. **Nama Posisi** (Contohnya, Software Engineer, Product Manager dsb)
 2. **Deskripsi**. Deskripsi berupa **rich text** dalam bentuk HTML yang merupakan informasi umum tentang posisi yang dibuka.
 3. **Jenis Pekerjaan**. Terdapat 3 jenis posisi: Internship, Part-time, Full-time
 4. **Jenis Lokasi**. Terdapat 3 jenis lokasi: Remote, Hybrid, On-site.
 5. **Attachment(s)**. Attachments dalam bentuk file image yang berkaitan dengan lowongan.
 6. **Created At**. Kapan lowongan dibuat.
- **Membuka atau menutup lowongan**. Secara *default*, lowongan akan “terbuka” ketika dibuat. Ketika lowongan ditutup, lowongan tetap dapat dilihat oleh *job seeker*. Namun, *job seeker* tidak dapat melamar ke lowongan yang ditutup.
- **Melakukan update informasi lowongan** yang telah dibuat.
- **Melihat daftar lowongan** yang dibuat.

4. Pencarian lowongan pekerjaan

- *Job seeker* dan pengguna yang belum terautentikasi dapat:
 - **Mencari** (*substring, case insensitive*) lowongan berdasarkan nama posisi.
 - **Memfilter** lowongan berdasarkan jenis pekerjaan (*full time, part time, atau internship*) dan tipe lokasi (*on-site, hybrid, atau remote*).
 - **Menyortir** lowongan berdasarkan waktu *postingan* secara *ascending* ataupun *descending*. Pencarian dan sortir wajib **dapat dilakukan secara bersamaan**.
- Wajib melakukan implementasi **debounce** pada pencarian untuk menghindari pemanggilan yang berlebihan.

5. Melamar pekerjaan

- *Job seeker* dapat:
 - Melamar pekerjaan. Saat melamar pekerjaan, *job seeker* WAJIB meng-*upload* CV dalam bentuk PDF. *Job seeker* juga dapat meng-*upload* video pengenalan (.mp4), tetapi *field* ini **tidak** wajib.
 - Melihat lamaran yang pernah dikirim dan status lamaran.
- *Company* dapat:
 - **Melihat daftar lamaran** yang masuk.
 - **Melakukan *approve/reject* lamaran** yang masuk. *Company* perlu mengisi alasan/tindak lanjut *approve/reject* berupa Rich Text dalam bentuk HTML. *Approve/reject* hanya bisa dilakukan pada lamaran yang berstatus waiting.

6. Penyimpanan File Binary

Tempat penyimpanan file binary (*image attachments* lowongan, CV pelamar dan video pengenalan pelamar) **pada local file storage**, tidak menyimpan binary file di database. Database hanya menyimpan path menuju lokasi file.

7. *Pagination*

Pagination



- Di [Halaman Home \(Company\)](#) dan [Halaman Home \(Job Seeker\)](#) wajib terdapat *pagination*, **tampilkan sejumlah lowongan** yang sesuai dengan batasan tertentu per halaman.
- Tampilkan navigasi (halaman berikutnya dan sebelumnya) atau tautan angka halaman untuk membantu pengguna pindah ke halaman lain.
- *Pagination* **wajib dilakukan secara server-side**.
- Selain dengan membagi *page*, kalian boleh mengimplementasikannya dengan *infinite scroll* seperti pada LinkedIn.

8. Rich Text Editor

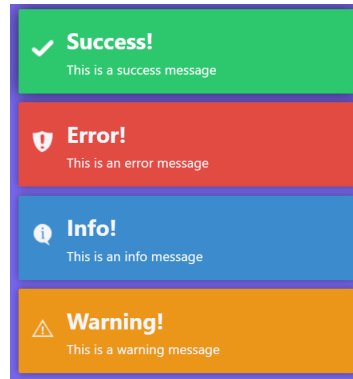
- Untuk mengimplementasikan *rich text editor*, gunakan kakas [quill.js](#).
- Apabila menggunakan *basic form handling event*, kalian dapat mengikuti [guide berikut](#)..
- **Pastikan keamanan** dalam mengimplementasikan *rich text editor* terutama ketika menampilkan hasil teks.

9. Keamanan

- *Client-side* wajib melakukan **validasi masukan** sederhana dari pengguna (misalnya *required field*, regex untuk nomor *handphone*, etc).
- *Server-side* wajib melakukan **validasi data** dari pengguna untuk memastikan integritas data (data unik, keberadaan data, logika, etc).
- *Password* pengguna wajib di-**hash** pada basis data, algoritma yang digunakan dibebaskan.

10. Error handling

Implementasikan **error handling** untuk memberikan umpan balik yang informatif kepada pengguna. Error handling mencakup *server-side* dan juga *client-side*, misalnya error untuk validasi input dan kegagalan request.



11. Responsivitas

Implementasikan minimal [Halaman Home \(Company\)](#) dan [Halaman Home \(Job Seeker\)](#) (termasuk *navigation bar*) yang **responsif** (minimal untuk ukuran 1280 x 768 dan 400 x 800). Artinya, tampilan mungkin berubah menyesuaikan ukuran layar.

12. Docker

Membuat Dockerfile dari aplikasi kalian, serta membuat file docker-compose.yml dari aplikasi kalian yang berisi aplikasi kalian serta database yang digunakan.

Pada *template* dari *repository* yang ada di Github Classroom, sudah terdapat Dockerfile dan docker-compose.yml yang **belum dapat dijalankan**. Silahkan diimplementasikan sehingga docker dapat dijalankan pada saat **demo**. Pastikan aplikasi dapat berjalan tanpa kakas apapun selain docker. Kalian juga boleh menggunakan docker untuk *development*, tetapi tidak diwajibkan dan bukan bagian dari penilaian.

Spesifikasi Tiap Halaman

1. Halaman Login

Halaman yang ditampilkan jika pengguna belum login dan ingin mengakses fitur-fitur yang hanya dapat dilakukan oleh pengguna yang terautentikasi. Pengguna dapat melakukan login dan terautentikasi sebagai *job seeker* atau *company* sesuai kredensial yang diberikan. Login hanya membandingkan email dan password saja. Tidak perlu tambahan proteksi apapun. Mekanisme autentikasi seperti yang dijelaskan sebelumnya.

2. Halaman Register

Pengguna dapat mendaftarkan akun baru jika belum login atau sudah logout. Pada halaman ini, akan terdapat pilihan untuk mendaftar sebagai *job seeker* atau sebagai *company*.

- Jika memilih untuk mendaftar sebagai *job seeker*, pengguna akan diminta untuk mengisi *form* dengan *field* nama pengguna, email pengguna, *password*, dan *confirm password*.
- Jika memilih untuk mendaftar sebagai *company*, pengguna akan diminta untuk mengisi *form* dengan *field* nama *company*, email *company*, *password*, *confirm password*. Pengguna juga akan diminta untuk memasukkan detail *company*, yaitu lokasi dan *about*. Kalian bebas berkreasi untuk mengimplementasikan fitur ini (entah menaruh semua langsung dalam 1 *form*, menggunakan *multi-step form*, membuat 2 halaman berbeda, dll.).

Baik untuk *job seeker* maupun *company*, email yang digunakan harus unik. *Email* juga harus memiliki format *email* standar seperti "example@example.com".

Setelah semua nilai *field* sudah diisi dan *valid*, pengguna dapat mendaftarkan akun barunya. Jika akun berhasil didaftarkan, pengguna langsung diarahkan ke [Halaman Home \(Job Seeker\)](#)/[Halaman Home \(Company\)](#) tergantung pengguna mendaftar sebagai apa.

3. Halaman Home (*Job Seeker*)

Halaman ini menampilkan daftar lowongan yang ada dari setiap perusahaan yang telah dipaginasi. Hal-hal yang perlu diperhatikan pada halaman ini:

1. Pengguna dapat melakukan pencarian, sortir, dan filter lowongan serta menampilkannya.
2. Pengguna dapat mengklik pilihan lowongannya dan melihat isi detail lowongan pada [Halaman Detail Lowongan \(Job Seeker\)](#).

Silakan berkreasi sesuai dengan imajinasi, asal memenuhi spesifikasi lainnya.

4. Halaman Home (*Company*)

Halaman ini menampilkan beberapa hal yang berisi aksi-aksi yang dapat sebuah *company* lakukan. Halaman utama ini menampilkan daftar lowongan kerja yang dibuka oleh *company* tersebut. Halaman ini juga berfungsi sebagai halaman manajemen lowongan kerja.

Jadi, selain menampilkan (termasuk mencari, *sort*, *filter*, dan pagination), pengguna juga harus dapat:

1. Menambah (navigasi/tombol yang merujuk ke [Halaman Tambah Lowongan \(Company\)](#)),
2. Mengupdate (menutup lowongan kerja, update deskripsi, dll yang merujuk ke [Halaman Edit Lowongan \(Company\)](#)), dan
3. menghapus lowongan kerja pada tampilan ini.

Kemudian, pengguna juga dapat mengklik lowongan tersebut untuk melihat pendaftar lowongan tersebut pada [Halaman Detail Lowongan \(Company\)](#).

Silakan berkreasi sesuai dengan imajinasi, asal memenuhi spesifikasi lainnya.

5. Halaman Tambah Lowongan (*Company*)

Halaman ini digunakan oleh *company* untuk membuat suatu lowongan baru.

Tidak ada spesifikasi khusus untuk halaman ini. Silakan berkreasi sesuai dengan imajinasi kalian, asalkan memenuhi spesifikasi yang lain salah satunya menerima form yang berisi *field-field* yang dibutuhkan untuk membuat suatu lowongan.

6. Halaman Detail Lowongan (*Company*)

Halaman ini digunakan *company* untuk:

- **Melihat detail lowongan** yang ia buka.
- **Melihat daftar lamaran** dari lowongan tersebut. Karena pelamar dari suatu lowongan bisa jadi sangat banyak, tampilkan nama pelamar serta statusnya saja.
- **Menghapus** lowongan tersebut.
- **Menutup** lowongan tersebut.

Sediakan navigasi agar *company* dapat melihat detail lamaran (ke [Halaman Detail Lamaran \(Company\)](#)) serta mengubah lamaran (ke [Halaman Edit Lowongan \(Company\)](#)).

7. Halaman Detail Lamaran (*Company*)

Pada halaman ini, tampilkan:

- Data *job seeker* yang melamar: nama dan email
- *Attachment* lamaran:
 - CV. Tampilkan dalam bentuk PDF *embed*, terserah dengan metode apa saja.
 - Video pengenalan (jika ada). Tampilkan dalam bentuk video player.
- Status lamaran, beserta alasan/tindak lanjutnya (jika ada)

Pada halaman ini juga, *company* dapat melakukan *approve/reject* serta memberikan alasan/tindak lanjutnya **hanya untuk lamaran yang masih berstatus *waiting***. Alasan/tindak lanjut ini dalam bentuk *rich text* HTML.

8. Halaman Edit Lowongan (*Company*)

Halaman ini mirip dengan halaman tambah lowongan, tetapi digunakan untuk mengubah detail lowongan yang sudah ada.

9. Halaman Detail Lowongan (*Job Seeker*)

Pada halaman ini, tampilkan profil *company* dan deskripsi dari lowongan tersebut. Jika *job seeker* belum melamar posisi tersebut, sediakan navigasi untuk ke Halaman Lamaran. Jika sudah, tampilkan detail lamaran yang ia lakukan, berupa *hyperlink* ke *attachment* (CV dan video pengenalan jika ada) serta statusnya. Jika status tersebut mempunyai alasan/tindak lanjut, tampilkan juga. Perhatikan bahwa ***job seeker* tidak dapat melamar ulang untuk lowongan yang sama.**

10. Halaman Lamaran (*Job Seeker*)

Halaman ini adalah *form* untuk mengirim lamaran ke suatu lowongan. Sediakan *field* untuk mengunggah data dan *attachment* yang diperlukan. Lamaran yang baru dikirim ini langsung berstatus *waiting*.

11. Halaman Riwayat (*Job Seeker*)

Halaman ini digunakan oleh *job seeker* untuk melihat daftar lamaran yang pernah dikirim beserta statusnya. Pastikan riwayat ini diurutkan dari lamaran yang paling terakhir dilakukan. Jika salah satu lamaran diklik, *job seeker* akan diarahkan ke [Halaman Detail Lowongan \(*Job Seeker*\)](#).

12. Halaman Profil (*Company*)

Halaman ini digunakan oleh *company* untuk melakukan perubahan profil seperti nama, lokasi, *about*, dll.

Spesifikasi Bonus

Catatan: Kerjakan dahulu spesifikasi wajib sebelum mengerjakan bonus.

1. **All Responsive Web Design**

Semua tampilan dibuat **responsif** (minimal untuk ukuran 1280 x 768 dan 400 x 800). Artinya, tampilan mungkin berubah menyesuaikan ukuran layar. Hint: gunakan CSS @media rule, lebih lanjut: https://www.w3schools.com/css/css_rwd_mediaqueries.asp.

2. **UI/UX seperti LinkedIn**

Membuat UI/UX yang serupa dengan tampilan website LinkedIn, dengan desain yang konsisten dan responsif untuk setiap halaman. Siapa tahu nanti kalian bisa dapat pekerjaan di LinkedIn. Kalian dapat menyesuaikan dengan fitur yang ada pada spesifikasi ini.

3. **Data Export**

Fitur ini memungkinkan *company* untuk mengunggah daftar pelamar ke dalam format **Excel** atau **CSV**. Ini berguna untuk keperluan internal seperti analisis, pengarsipan, atau pelaporan. Fungsionalitas ini bisa ditambahkan dengan menggunakan tombol “Export” di halaman daftar pelamar. Isi dari file ini minimal terdapat:

1. Nama
2. Pekerjaan yang dilamar
3. Tanggal melamar
4. URL file CV dan *attachment* lainnya
5. Status lamaran

4. Simple Job Recommendation

Fitur ini memberikan rekomendasi lowongan kerja kepada pelamar berdasarkan riwayat pelamaran mereka sebelumnya. Dengan cara ini, pelamar dapat menemukan lowongan yang relevan dan sesuai dengan minat serta pengalaman mereka tanpa harus mencarinya secara manual. Fitur bonus ini tidak ketat, kalian boleh berkreasi sendiri. Namun, tetap harus ada algoritma/logika yang sesuai. Contoh implementasi:

- **Contoh 1:** Pengguna telah melamar ke 3 pekerjaan yang berbeda, 2 pekerjaan di antaranya memiliki bidang/kategori yang sama yakni "Teknologi", maka lowongan rekomendasinya adalah pekerjaan yang memiliki bidang "Teknologi" juga. Jika belum ada lamaran, silahkan pikirkan sendiri implementasinya.
- **Contoh 2:** Rekomendasi pekerjaan ditampilkan dengan sistem "trend". Lowongan kerja yang baru dibuat (*fresh*) oleh perusahaan dan ternyata memiliki *engagement* yang "tiba-tiba" banyak bisa dijadikan salah satu penerapannya. *Engagement* di sini bisa dari banyaknya jumlah pelamar dalam suatu rentang waktu tertentu.

Hint: *created_at, updated_at, count, sort*

Tidak ada keketatan dalam bonus ini, silahkan berkreasi sendiri. Masing-masing implementasi tentu memiliki nilai yang berbeda-beda.

5. Google Lighthouse

Google Lighthouse adalah alat otomatis *open-source* untuk meningkatkan kualitas halaman web. Lighthouse dipakai sebagai alat pengukuran dan audit untuk performance kualitas website, aksesibilitas, aplikasi web progresif, dan banyak lagi. Tugas kalian adalah melakukan pengecekan skor di Lighthouse untuk seluruh halaman dan pastikan bahwa skor untuk **best practices**, **aksesibilitas** yang didapatkan memiliki **nilai di atas 90** dan **performance** yang didapatkan memiliki **nilai di atas 80**. Jika pada saat pertama kali pengukuran skor masih kurang, tetap *capture* dan jelaskan perubahan apa

yang kalian lakukan untuk mencapai nilai yang lebih baik. Lampirkan bukti tangkapan layar pada README.

Keyword & Tips

Untuk meringankan beban tugas ini, ada beberapa *keyword* yang bisa kalian cari untuk menyelesaikan tugas ini.

- **HTTP methods:** GET, POST, PUT, PATCH, DELETE, OPTIONS.
- **CSS:** margin, padding, font-size, text-align, flex, grid, border, color, div, span, anchor tag, box-shadow.
- **Javascript:** XMLHttpRequest, addEventListener.
- **PHP:** mysqli_connect, mysqli_query, pg_connect, pg_query, PDO, \$_GET, \$_POST, \$_COOKIE, var_dump, print_r, echo, require.
- **SQL:** SELECT, INSERT, UPDATE, DELETE, WHERE, LIKE, ORDER BY.
- **DOCKER:** Dockerfile, Docker Compose, Docker Container, Docker Image, Docker Volume.

Berikut adalah *tips* dari asisten tercinta.

- Usahakan untuk membuat code yang maintainable (*reusable, loosely coupled, dll*). Sebagai contoh, dalam membuat *navbar*, jangan meng-copy navbar pada setiap berkas php namun manfaatkan **PHP templating**, lebih lanjut: <https://css-tricks.com/php-templating-in-just-php/>.
- Gunakan [autoload](#) pada PHP untuk mempermudah modularitas kode.
- Gunakan design pattern untuk mempermudah arsitektur program, misalnya MVC (Model View Controller). Kalian bisa mengimplementasikan *routing* sendiri alih-alih *routing* berdasarkan direktori file php.
- Dalam menggunakan Docker, pahami dulu istilah-istilah yang dipakai pada Docker, jangan langsung mencoba, karena akan sulit apabila tidak berjalan atau menemukan *error*. Misalnya, pahami dulu perbedaan pembuatan *docker image* dan *docker container*. Lalu, ketahui cara kerja *docker container* mulai dari *port forwarding*, *docker volume*, dan *docker network*.
- Jangan berantakan saat membuat kode PHP. Ini saran agar keberlangsungan milestone 2 kalian lancar dan tidak terhambat oleh milestone 1 ini. Apabila

tugas besar ini dipakai untuk matkul lain, alur program dapat dibaca dan diingat dengan cepat.

Lain-Lain

Responsi

Diadakan responsi per kelompok bersama asisten menjelang pengumpulan tugas.

- Responsi bersifat **opsional**. Artinya **tidak wajib**.
- Kelompok yang ingin mendaftarkan diri dapat mengisi pada [pranala](#) berikut.
- Deadline pengisian jadwal responsi adalah **9 Oktober 2024 pukul 23.59 WIB**.
- Perlu diperhatikan bahwa ketersediaan asisten **terbatas**.
Siapkan terlebih dahulu hal-hal yang ingin ditanyakan / dikonsultasikan dan lakukan setup project di komputer kalian **sebelum** sesi responsi dimulai.
- Pastikan sudah riset dan **mencoba** mempelajari sebelum bertanya ke asisten. Agar bisa bertanya lebih banyak pertanyaan dan pertanyaan lebih dekat ke pengerjaan bukan *understanding*.
- Silahkan **membuat undangan** di Google Calendar sesuai jadwal yang telah diisi menggunakan email STD asisten.

Daftar Pertanyaan

Jika masih ada pertanyaan mengenai spesifikasi tugas, dapat dilakukan melalui [QnA](#) berikut.

Pastikan **sudah membaca spesifikasi sepenuhnya hingga akhir** dan tidak ada pertanyaan yang berulang.

Deliverables

Berikut adalah hal yang harus diperhatikan untuk pengumpulan tugas ini:

1. Buatlah tim pada github classroom [berikut](#) dengan format nama tim **<kelas>-<nomor kelompok>**, contoh: **K01-02** oleh salah satu anggota kelompok. Anggota lainnya bisa *join* ke tim yang sudah dibuat.
2. Hal-hal yang harus diperhatikan.
 - a. Silakan *commit* pada *repository* kalian yang dibuat otomatis..
 - b. Lakukan beberapa *commit* dengan pesan yang bermakna, contoh: "add register form", "fix logout bug", jangan seperti "final", "benerin dikit", "fix bug".
 - c. Disarankan untuk tidak melakukan *commit* dengan perubahan yang besar karena akan mempengaruhi penilaian (contoh: hanya melakukan satu *commit* kemudian dikumpulkan).
 - d. *Commit* dari setiap anggota tim akan mempengaruhi penilaian.
Jadi, setiap anggota tim harus melakukan commit yang berpengaruh terhadap proses pembuatan aplikasi.
 - e. Sebagai panduan, kalian bisa mengikuti [semantic commit](#).
3. Buatlah berkas **README** yang berisi:
 - a. Deskripsi aplikasi web
 - b. Daftar *requirement*
 - c. Cara instalasi
 - d. Cara menjalankan *server*
 - e. Tangkapan layar tampilan aplikasi (tidak perlu semua kasus, minimal 1 per halaman), dan
 - f. Penjelasan mengenai pembagian tugas masing-masing anggota (lihat formatnya pada bagian pembagian tugas).

Pengumpulan Tugas

- Tenggat waktu tugas adalah pada hari **24 Oktober 2024 pukul 21.00 WIB**.
- Tahap pengumpulan adalah sebagai berikut.

- a. Buat *release* dengan format **v1.x** pada Gitlab Informatika dan judul bebas tetapi bermakna.
X adalah nomor yang dimulai dari 1.
- b. Apabila kalian ingin merevisi tugas, buat *release* dengan penambahan angka sebelumnya.
Contoh, kelompok kalian sudah merilis v1.1, tetapi ternyata ada revisi sedikit, silakan buat rilis v1.2. **Tetap gunakan major version 1 yaitu v1.x untuk milestone 1 ini.**
Tugas yang dinilai adalah *release* versi terakhir.
- c. Waktu pengumpulan tugas yang dilihat adalah **waktu *release version* terakhir ke server Github** terakhir.
Akan ada pengurangan terhadap *release version* yang melebihi waktu tenggat waktu tugas.

Pembagian Kelompok

Pembagian kelompok ada di [pranala](#) berikut.

Bagi yang belum mendapatkan kelompok dapat mengontak asisten melalui Teams.

Pembagian Tugas

Setiap anggota kelompok **diwajibkan** untuk mengerjakan bagian *server-side* (PHP) dan *client-side* (HTML, CSS, JS), dengan harapan kalian bisa mencoba mempelajari semua bagian dan cara kerjanya.

Server-side

Login : 13521xxx, 13521xxx

Register : 13521xxx

(Lanjutkan ...)

Client-side

Login : 13521xxx, 13521xxx

Register : 13521xxx

(Lanjutkan...)

Catatan Tambahan

Milestone ini akan berhubungan dengan *milestone* selanjutnya; artinya tugas yang kalian buat di sini **akan digunakan kembali pada *milestone* berikutnya**. Usahakan untuk menyelesaikan spesifikasi utama agar tidak terhambat pada *milestone* berikutnya.

Tambahan



Goodluck guys - kordas



Semangat^_^~chi

*Kaluna pasti bisa nabung 300jt karena
ga perlu ngumpul di kafe buat nubes </3
- rachel*