

# **BAB I**

## **PENDAHULUAN**

### **A. Latar belakang**

Era digital saat ini kebutuhan akan *Curriculum Vitae* (CV) yang menarik dan informatif sangatlah penting. Ini karena CV adalah dokumen penting yang mengungkapkan informasi tentang latar belakang pendidikan, pengalaman kerja, keterampilan, dan prestasi seseorang. sebuah CV yang dibuat dengan baik dapat membuka jalan menuju prospek karir yang lebih menjanjikan atau meningkatkan kemungkinan diterima di perusahaan atau institusi yang diinginkan (Kusumawardhani & Febriyanti, 2018). Selain itu, di pasar kerja yang semakin kompetitif, CV yang menonjol dapat membedakan kandidat dari yang lain dan menarik perhatian pemberi kerja.

Proses pembuatan *Curriculum Vitae* (CV) secara manual seringkali membosankan dan memakan waktu. Proses ini membutuhkan pengumpulan semua informasi yang relevan, pemformatan dokumen sesuai dengan standar yang diinginkan, dan memastikan tidak adanya kesalahan atau ketidakkonsistenan. Proses ini dapat menjadi lebih kompleks bagi mereka yang memiliki pengalaman kerja atau pendidikan yang beragam, atau bagi siswa yang baru lulus yang tidak terbiasa dengan struktur dan konten CV profesional (Patel & Desai, 2020). Dengan pesatnya perkembangan teknologi, ada tuntutan untuk membuat CV yang lebih menarik secara visual dan mudah diakses di berbagai perangkat, baik desktop

maupun *mobile* (Setyawan & Sari, 2019). Di era kontemporer, perusahaan sering menggunakan Sistem Pelacakan Pelamar (*Applicant Tracking Systems/ATS*) untuk menyaring CV. Oleh karena itu, CV harus dioptimalkan agar dapat melewati sistem ini (Johnson & Gueutal, 2021). Selain itu, CV yang responsif dan mudah dibaca di perangkat seluler sangat penting, karena banyak profesional sumber daya manusia yang memeriksa lamaran melalui ponsel mereka.

Diperlukan sebuah sistem yang dapat membantu pembuatan *Curriculum Vitae* (CV) secara otomatis dan efisien. Sistem tersebut harus dapat mengumpulkan informasi yang diperlukan dari pengguna, memformat data ke dalam tampilan yang menarik dan konsisten, dan menghasilkan CV yang siap dicetak atau dibagikan secara digital. Selain itu, sistem harus cerdas dalam mengorganisir informasi sesuai dengan jenis pekerjaan atau industri yang ditargetkan, memastikan penggunaan kata kunci yang tepat untuk melewati (*Applicant Tracking Systems/ATS*), dan menyediakan *template* yang sesuai dengan tren desain terkini (Smith & Brown, 2022).

Salah satu teknologi yang sangat relevan dengan kebutuhan ini adalah aplikasi web progresif (*progressive web apps/PWA*). Teknologi ini merupakan konsep rancang bangun aplikasi web yang menggabungkan fitur-fitur terbaik dari aplikasi web dan aplikasi *mobile* (Supriadi & Budiman, 2020). Dengan menggunakan PWA, sistem pembuatan CV otomatis dapat diakses melalui browser di perangkat apapun, baik desktop maupun *mobile*, tanpa perlu mengunduh aplikasi khusus. Teknologi PWA juga menawarkan keunggulan seperti kemampuan untuk bekerja secara offline, loading yang cepat, dan tampilan yang menyerupai aplikasi

*mobile native*. Alhasil, pengguna dapat mengakses dan membuat CV dari lokasi mana pun dan kapan pun, tanpa terkendala konektivitas internet maupun jenis perangkat yang digunakan (Kurniawan & Saputra, 2021).

Dalam konteks ini, diharapkan pengguna dapat lebih berkonsentrasi pada konten dan kualitas informasi yang mereka masukkan, bukan pada aspek teknis seperti format atau desain. Pendekatan ini tidak hanya akan menghemat waktu dan tenaga, tetapi juga meningkatkan kualitas dan efektivitas CV secara keseluruhan, sehingga meningkatkan peluang mereka untuk bersaing di pasar kerja yang dinamis. Oleh karena itu, dirancanglah "***Rancang bangun Sistem Pembuatan CV Otomatis Berbasis Progressive Web Apps***".

## **B. Rumusan Masalah**

Dengan latar belakang yang telah dijelaskan, maka yang menjadi rumusan masalah pada penelitian ini adalah Bagaimana merancang dan mengembangkan sebuah sistem pembuatan (*Curriculum Vitae*) CV otomatis Berbasis *Progressive Web Apps* yang efisien untuk memudahkan pelamar kerja dalam membuat CV secara cepat dan efisien?

## **C. Tujuan Penelitian**

Tujuan dari penelitian ini adalah merancang dan mengembangkan CV otomatis menggunakan teknologi *Progressive Web Apps* untuk memberikan kontribusi untuk memudahkan pelamar kerja dalam membuat CV secara cepat dan efisien.

### **D. Batasan Masalah**

Berdasarkan rumusan masalah di atas, maka batasan masalah dalam penelitian ini adalah sebagai berikut:

1. Sistem akan mengumpulkan informasi yang diperlukan untuk membuat CV dari pengguna, seperti data pribadi, riwayat pendidikan, pengalaman kerja, keterampilan, dan prestasi. Informasi yang tidak termasuk dalam kategori ini mungkin tidak dapat dimasukkan ke dalam CV yang dihasilkan oleh sistem.
2. Sistem ini hanya akan fokus pada pembuatan CV atau daftar riwayat hidup secara otomatis. Fitur-fitur lain seperti pencarian lowongan pekerjaan atau jaringan profesional tidak akan dimasukkan dalam lingkup rancang bangun sistem ini.
3. Pengguna tidak dapat membuat atau mengubah template secara bebas dalam sistem ini.
4. Sistem akan dikembangkan dengan menggunakan konsep *Progressive Web Apps* (PWA) yang dapat diakses melalui browser web tanpa perlu mengunduh aplikasi khusus. Sistem ini tidak akan dikembangkan sebagai aplikasi seluler native.

### **E. Manfaat Penelitian**

Adapun manfaat yang dapat diberikan dari penelitian ini adalah sebagai berikut:

1. Efisiensi waktu dan upaya

Sistem ini memungkinkan pengguna untuk membuat CV secara otomatis

dengan mengisi informasi yang diperlukan, menghilangkan kebutuhan untuk memformat secara manual. Hal ini dapat menghemat banyak waktu dan upaya yang biasanya dibutuhkan untuk membuat CV secara tradisional.

## 2. Konsistensi dan kualitas CV

Dengan menggunakan template dan format yang seragam, sistem ini dapat memastikan konsistensi dalam penampilan dan struktur CV yang dihasilkan. Ini dapat meningkatkan kualitas dan daya tarik CV bagi calon pemberi kerja.

## 3. Aksesibilitas yang lebih baik

Dengan menggunakan konsep *Progressive Web Apps* (PWA), sistem ini dapat diakses dari berbagai perangkat, baik desktop maupun seluler, tanpa perlu mengunduh aplikasi khusus. Ini meningkatkan aksesibilitas dan memungkinkan pengguna untuk membuat CV di mana pun dan kapan pun.

## BAB II

### TINJAUAN PUSTAKA

#### A. Kajian teori

##### 1. *Progressive web apps*

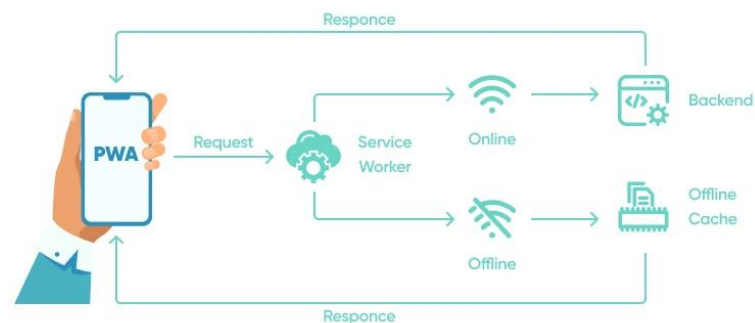


**Gambar 2. 1** *progressive web apps*

*Progressive Web Apps* adalah untuk teknologi untuk membuat aplikasi web yang memberikan tampilan dan nuansa aplikasi web konvensional pada aplikasi asli. Pengembang dapat menyempurnakan aplikasi web tradisional dengan semua komponen dan *fungsi* aplikasi seluler dengan menggunakan aplikasi web *progressive*. Ini memperluas pengalaman pengguna (UX) yang luar biasa dan meningkatkan kegunaan. Rancang bangun aplikasi ini menciptakan pengalaman web dengan *fungsi* aplikasi. PWA adalah aplikasi web yang memanfaatkan kemampuan khusus browser untuk berperilaku lebih seperti aplikasi native atau *mobile app* saat digunakan dengan browser yang compatible (Wargo, 2020). Sedangkan menurut (Ater, 2019) PWA adalah aplikasi web generasi baru yang menggabungkan fitur dan sumber daya aplikasi native dengan aplikasi web standar. PWA adalah situs

web standar yang dibuat dengan teknologi HTML, CSS dan *Javascript* yang kita kenal dan sukai sebagai web developer, tetapi PWA selangkah lebih maju dan memberi pengguna pengalaman yang lebih baik (Alan Hume, 2021). Sebuah PWA harus dapat berjalan dengan baik di berbagai perangkat dan ukuran layar, memberikan pengalaman yang mirip dengan aplikasi native, selalu mendapatkan pembaruan terbaru, mudah ditemukan oleh pengguna, dapat diinstal di perangkat pengguna, dan dapat dibagikan dengan pengguna lain.

PWA dapat meningkatkan *fungsi* web *app* yang sudah ada, baik itu dibangun menggunakan *framework* atau murni *Javascript* (*Vanilla JS*). PWA dapat diterapkan pada website yang dibangun menggunakan *React*, *Angular*, atau *Vue*. Anda bahkan dapat memasukkan fitur ini ke dalam website yang sudah aktif tanpa banyak mengubah kode.

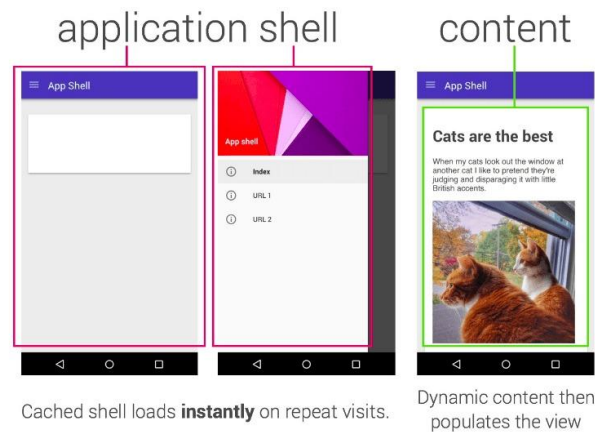


**Gambar 2. 2 Arsitektur teknologi PWA**

Ada beberapa komponen yang harus digunakan saat membangun PWA. Beberapa komponen harus digunakan setiap kali membuat PWA, sedangkan yang lain adalah opsional. Komponen wajib digunakan setiap kali membuat PWA, sedangkan yang opsional adalah komponen yang tidak mempengaruhi

kinerja PWA tetAPI dapat digunakan untuk meningkatkan fiturnya. Berikut komponen wajib yang dimaksud:

a. *Application shell*



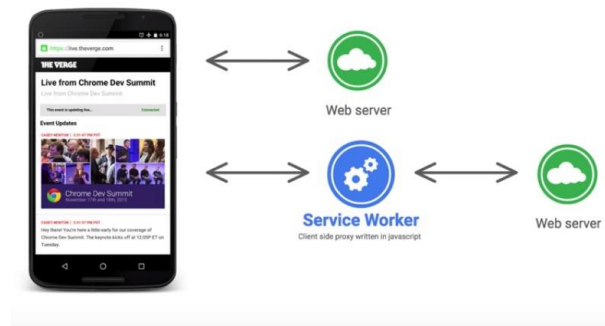
**Gambar 2. 3** Application Shell

*Application Shell*, juga disebut sebagai "*shell aplikasi*", adalah kerangka antarmuka aplikasi yang terdiri dari berbagai komponen halaman dan aset lainnya. Mereka dapat muncul secara instan saat aplikasi dibuka karena telah disimpan lebih dahulu di dalam *cache*.

b. *Web app manifest*

*File JSON* yang disebut *Web App Manifest* yang mengatur bagaimana bagaimana aplikasi ditampilkan dan dijalankan di sisi pengguna. Dalam kebanyakan kasus, *file* ini disebut *manifest.json*. *Browser* akan membaca *file manifest*, mengunduh sumber daya, dan menampilkan konten saat program pertama kali diluncurkan di *browser*.

c. *Service worker*



**Gambar 2. 4 Service worker**

*Service worker* adalah *file* yang ditulis dalam bahasa *Javascript* yang berjalan di latar belakang *browser* dan terpisah dari *script* lain pada halaman *web* tersebut. Dengan adanya *service worker*, *browser* dapat menampilkan konten bahkan dalam mode jaringan *offline*.

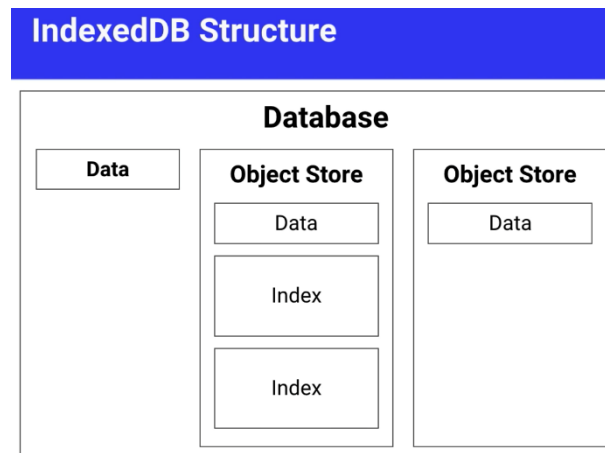
d. *Cache application programming interface*



**Gambar 2. 5 Cache**

*Cache Application Programming Interface (API)* adalah ruang penyimpanan lokal yang digunakan *browser* untuk menyimpan sumber daya sehingga dapat digunakan kembali selama panggilan berikutnya tanpa membuat permintaan jaringan tambahan.

e. *IndexedDB*

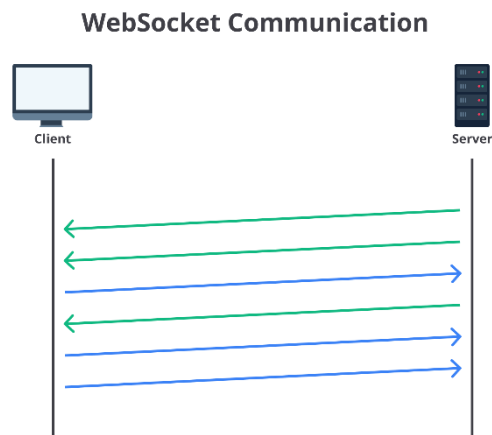


**Gambar 2. 6 indexedDB**

Data apa pun dapat disimpan sebagai objek *Javascript* di sistem penyimpanan lokal untuk *browser* yang disebut *IndexedDB*, yang berbasis NoSQL. Selain berperan sebagai *cache*, *IndexedDB* juga dapat digunakan untuk menyimpan *item* tertentu.

f. *Web Socket*

Standar baru untuk komunikasi *real-time* di *web* atau aplikasi seluler disebut *Web Socket* yang banyak digunakan saat ini untuk membuat aplikasi obrolan waktu nyata, umpan berita, dan bahkan *game* multipemain daring. *Web Socket* menjadi standar dan komponen HTML5 di dalam *browser* itu sendiri.



**Gambar 2. 7 Websocket**

*Client* dan *server* harus selalu terkoneksi agar dapat menerima sinyal komunikasi dari kedua belah pihak saat menggunakan pola komunikasi aktif dua arah (*full-duplex*) seperti ini. Kami dapat membangun komunikasi waktu nyata karena kami selalu terhubung. *Web Socket* dapat digunakan untuk komunikasi waktu nyata, *streaming* langsung, *game* multipemain, dan progres waktu nyata.

*g. Notifications*

Pesan *pop-up* yang dikenal sebagai notifikasi dapat ditampilkan di *gadget*. Situs *web* yang mendukung *PWA* mendukung tampilan notifikasi. Memberi tahu pengguna tentang informasi apa pun sangat penting. Metode efektif untuk mendorong orang kembali ke situs *web* adalah pemberitahuan.

## 2. *Curriculum vitae(cv)*

*Curriculum Vitae* (CV) adalah dokumen komprehensif yang merangkum kualifikasi profesional, pendidikan, dan pengalaman kerja seseorang. Berbeda dengan resume, CV biasanya lebih detail dan digunakan sebagai alat pemasaran diri dalam pencarian kerja atau rancang bangun karir (Smith, 2021). Komponen utama CV meliputi informasi pribadi, ringkasan profesional, riwayat pendidikan, pengalaman kerja, keterampilan, pencapaian, dan kegiatan profesional. Untuk posisi akademis, CV juga mencantumkan publikasi dan presentasi.

Dalam era digital, format CV telah berkembang. CV digital atau *e-CV* semakin populer, memungkinkan penggunaan tautan, portofolio online, dan elemen multimedia (Johnson, 2022). Perkembangan *Artificial Intelligence* (AI) dalam rekrutmen telah mendorong munculnya CV yang "*ATS-friendly*" (*Applicant Tracking System*), yang dioptimalkan untuk sistem perekrutan otomatis (Brown, 2023).

Penelitian terbaru menekankan pentingnya personalisasi CV sesuai dengan posisi dan perusahaan yang ditargetkan (Garcia, 2024). Penggunaan kata kunci yang relevan dengan industri dan deskripsi pekerjaan dapat meningkatkan peluang CV untuk lolos seleksi awal. Integrasi elemen personal branding juga menjadi tren dalam CV modern. Tautan ke profil *LinkedIn*, portofolio online, atau blog profesional sering disertakan untuk memberikan gambaran yang lebih holistik tentang kandidat (Wilson, 2024).

Meskipun teknologi membawa perubahan dalam format dan penyajian CV, prinsip-prinsip dasar seperti kejelasan, kejujuran, dan relevansi informasi tetap penting (Smith & Brown, 2022). Penggunaan bahasa yang jelas dan ringkas, penekanan pada pencapaian terukur, dan penyajian informasi yang terorganisir dengan baik masih menjadi praktik terbaik. CV terus berevolusi mengikuti perkembangan teknologi dan tuntutan pasar kerja. Adaptabilitas menjadi kunci, dengan kandidat perlu secara reguler memperbarui CV mereka.

Pemahaman tentang tren terbaru dalam penyusunan CV, dikombinasikan dengan kemampuan menyajikan informasi secara menarik dan relevan, memberikan keunggulan kompetitif dalam pasar kerja yang dinamis (Johnson, 2022). CV mungkin akan menuju format yang lebih interaktif dan terintegrasi dengan platform digital. Namun, esensinya sebagai representasi komprehensif dari kualifikasi dan pengalaman profesional akan tetap relevan dalam perjalanan karir individu. CV tetap menjadi instrumen penting dalam proses rekrutmen dan rancang bangun karir. Seiring perkembangan teknologi dan tuntutan pasar kerja, format dan konten CV terus berubah, menekankan pentingnya adaptasi dan pembaruan reguler untuk memastikan efektivitasnya dalam menyajikan kualifikasi profesional seseorang (Brown, 2023).

### 3. HTML



**Gambar 2. 8** *Hypertext Markup Language*

*HyperText Markup Language* atau HTML, adalah bahasa markup standar industri untuk membangun halaman web. Ini menawarkan kumpulan tag dan properti yang menentukan pengorganisasian halaman web, konten, dan format halaman web. Untuk merepresentasikan elemen pada halaman web, HTML menggunakan struktur hierarkis yang dikenal sebagai *Document Object Model* (DOM). Teks, foto, tautan, multimedia, dan elemen lainnya semuanya dapat dimuat dalam setiap elemen, yang dibatasi oleh tag pembuka dan penutup.

Istilah "*HyperText*" mengacu pada kemampuan HTML untuk membuat tautan yang memungkinkan pengguna dengan mudah dan cepat berpindah dari satu situs web ke situs web lainnya. *HyperText Markup Language*, juga dikenal sebagai HTML, adalah bahasa global *World Wide Web*. Seperti yang tersirat dari namanya *HyperText Markup Language*, HTML adalah bahasa markup bukan bahasa pemrograman (Hartl & Donahoe, 2019).

Dokumen yang dibuat dengan menggunakan bahasa markup seperti HTML memiliki bagian yang dirender pada hasil akhir serta bagian

yang menginstruksikan program rendering cara menginterpretasikan teks yang tersisa. Proses mengubah dokumen teks yang berisi teks HTML menjadi, misalnya, representasi visualnya di layar disebut sebagai "*rendering*" (Nematrian, 2020).

Perkembangan HTML sangat pesat. Dari permulaan HTML 1.0 yang sederhana, yang mencakup elemen dasar seperti judul dan paragraf, hingga penambahan tabel dan gambar pada HTML 2.0, dan akhirnya ke frame dan dukungan form yang disempurnakan pada HTML 3.2. Dengan CSS, konten dan presentasi sekarang dapat dipisahkan dalam HTML 4.01, yang juga meningkatkan komponennya. Dengan elemen semantiknya, dukungan multimedia, bentuk yang lebih baik, fitur aksesibilitas, dan kemampuan *mobile-friendly*, HTML5 menandai periode transformasi.

Versi HTML terbaru, yang dikenal sebagai HTML5, digunakan untuk membuat dan mengatur dokumen di internet. Ini mencakup elemen dan fitur baru yang memungkinkan pembuatan halaman web yang lebih dinamis dan menarik, termasuk dukungan untuk audio dan video, kanvas untuk grafik dan animasi, dan peningkatan kompatibilitas dengan perangkat seluler.

*Fungsionalitas* baru yang diperkenalkan oleh HTML5 memungkinkan pembuatan halaman web yang lebih dinamis dan interaktif, yaitu:

- a. Dukungan audio dan video.
- b. Kanvas visual dan animasi.
- c. Dukungan yang ditingkatkan untuk *mobile*.

- d. *Semantic Element* baru termasuk *nav*, *article*, *sections* dan *headers*.
- e. Kontrol untuk form, yaitu tanggal, waktu, *email*, pencarian, dll.
- f. Penyimpanan web untuk menyimpan informasi secara lokal.
- g. Fitur drag and drop.

#### 4. CSS



**Gambar 2. 9** *cascading style sheets*

CSS adalah bahasa yang terus berkembang untuk menggambarkan presentasi konten web di layar, printer, sintesis ucapan, pembaca layar, dan jendela obrolan. CSS digunakan oleh semua browser pada semua ukuran layar di semua jenis perangkat IoT, termasuk ponsel, komputer, video game, televisi, jam tangan, dan konsol otomatis (Meyer & Weyl, 2020). Sedangkan menurut (Duckett, 2022) CSS adalah bahasa pemrograman web yang digunakan untuk mengendalikan tampilan halaman web. (Castro, 2022) menjelaskan bahwa CSS memiliki berbagai properti yang dapat digunakan untuk mengatur tampilan halaman web. CSS terdiri dari dua bagian utama, yaitu deklarasi dan selektor. Deklarasi adalah bagian yang menentukan properti dari suatu halaman web. Selektor adalah bagian yang menentukan elemen web mana yang akan

diatur oleh deklarasi. CSS juga memiliki berbagai properti yang dapat digunakan untuk mengatur tampilan halaman web. Beberapa properti CSS yang umum digunakan antara lain:

- a. Ukuran: properti ini digunakan untuk mengatur ukuran halaman web, seperti ukuran font, ukuran gambar, dan ukuran jarak antar elemen.
- b. Warna: properti ini digunakan untuk mengatur warna halaman web, seperti warna font, warna latar belakang, dan warna border.
- c. Font: properti ini digunakan untuk mengatur font elemen web, seperti jenis font, ukuran font, dan gaya font.
- d. Tata letak: properti ini digunakan untuk mengatur tata letak elemen web, seperti posisi elemen, ukuran elemen, dan jarak antar elemen.

## 5. PHP



**Gambar 2. 10** *Hypertext Preprocessor*

PHP adalah bahasa *server-side* yang secara khusus digunakan untuk aplikasi web. Bahasa PHP berjalan di sisi *server*, artinya ia berjalan pada *server* dan bisa dimasukkan di antara bahasa HTML (*HyperText Markup Language*). Oleh karena itu, kode PHP tidak akan terlihat lagi dan hasil akhir dalam bentuk HTML akan dikirimkan ke browser (Sutarman, 2020). Karena PHP dijalankan di komputer *server*, itu dikenal sebagai bahasa pemrograman sisi *server*. Ini berbeda dengan bahasa pemrograman sisi klien, seperti *Javascript*, yang

beroperasi pada klien, yaitu web browser.

Pada awalnya, PHP adalah singkatan dari *Personal Home Page*. PHP digunakan untuk mengembangkan situs web pribadi, sesuai dengan namanya. Seiring berjalannya waktu, PHP berkembang menjadi bahasa pemrograman web yang sangat kuat, tidak hanya digunakan untuk membuat halaman web dasar tetapi juga untuk situs-situs populer seperti Wikipedia, WordPress, Joomla, dan lainnya yang dikunjungi oleh jutaan orang.

## 6. **JAVASCRIPT**



**Gambar 2. 11 JAVASCRIPT**

*Javascript* adalah bahasa pemrograman *open-source* untuk teknologi web yang dapat digunakan untuk memodifikasi HTML dan CSS. Biasanya disingkat menjadi JS saja. Sebagian besar browser web, termasuk Internet Explorer, Google Chrome, dan Mozilla Firefox, menggunakan *Javascript* untuk operasi yang dilakukan di antarmuka pengguna. Karena kemampuannya untuk menginstruksikan browser untuk melakukan tugas, sekarang *Javascript* menjadi *client-side scripting language* yang paling populer (Shute, 2020).

*Javascript* dapat meningkatkan kegunaan halaman web dengan

memberikan umpan balik yang cepat. Dengan bantuan bahasa pemrograman *Javascript*, Anda dapat meningkatkan HTML Anda dengan menambahkan animasi, interaksi, dan efek visual yang dinamis untuk pengguna browser. (McFarland, 2021). Selain itu, *Javascript* dapat digunakan pada platform apa pun dan merupakan alat yang berguna untuk mengembangkan program dan makro sementara. Anda dapat dengan cepat membuat perangkat lunak yang layak dengan menggunakan *text Editor* dan browser web berkat ketersediaan browser yang tersebar luas (Patrick, 2019).

Saat pertama kali keluar, *Javascript* hanya memiliki sedikit "*bagian yang bagus*". Saat ini, sebagian besar komponen sangat baik. Meskipun masih ada beberapa fitur lawas, fitur baru yang ditawarkan oleh standar ES6/7/8 membantu Anda menulis kode yang lebih aman, mudah, dan singkat (Subramaniam, 2019). Kode *Javascript* mengalami perubahan yang cukup signifikan saat ES6 atau *ECMAScript 6* diterbitkan. Sekarang Anda akan dapat dengan cepat membedakan antara kode yang ditulis menggunakan *sintaks* sebelum ES6 diterbitkan dan kode yang ditulis menggunakan *sintaks* ES6 karena perubahannya sangat signifikan. *Javascript* modern saat ini sangat menyenangkan untuk ditulis. Saya telah lama menggeluti dunia pemrograman, tetapi saya akui bahwa kadang-kadang saya merasa *sintaks* sebelumnya canggung dan sulit dijelaskan daripada *sintaks Javascript* modern ini yang lebih unggul (Morgan, 2019).

Tidak seperti ES5, ES6 menambahkan lebih dari sekadar sejumlah kecil *API* baru ke dalam bahasa. Ini mencakup banyak struktur *sintaksis* baru,

beberapa diantaranya mungkin memerlukan waktu untuk membiasakan diri. Selain itu, ada banyak organization form baru dan dukungan *API* tambahan untuk berbagai jenis data. ES6 adalah kemajuan yang signifikan untuk bahasa ini. ES6 penuh dengan fitur baru yang belum Anda ketahui (Simpson, 2020).

## 7. Framework Tailwind css



**Gambar 2. 12** *Tailwind css*

*Tailwind CSS* adalah *framework CSS utility-first* yang mendukung pengembangan antarmuka pengguna secara custom dan responsif (Azhariyah & Mukhlis, 2024). Berbeda dengan *framework CSS* tradisional seperti Bootstrap yang menyediakan komponen UI *pre-built*, *Tailwind CSS* menyediakan kelas-kelas CSS modular yang dapat dikombinasikan untuk membangun berbagai macam elemen UI. Keunggulan utama *Tailwind CSS* adalah fleksibilitasnya yang tinggi. Developer tidak terpaksa pada desain *pre-built* dan memiliki kebebasan penuh untuk menciptakan UI sesuai dengan visi mereka. Selain itu, *Tailwind CSS* mudah dipelajari dan digunakan, bahkan bagi developer pemula. *Framework* ini juga memfasilitasi desain *responsif* yang memungkinkan tampilan yang konsisten di berbagai perangkat (Rifandi et al.,

2022).

Meskipun *Tailwind CSS* memiliki banyak keunggulan, *framework* ini juga memiliki beberapa kekurangan. Salah satunya adalah kurva belajar yang relatif lebih tinggi dibandingkan CSS tradisional. Developer perlu mempelajari kelas-kelas CSS dan cara kombinasinya untuk dapat menggunakan *Tailwind CSS* secara efektif. Selain itu, penggunaan kelas-kelas CSS *Tailwind CSS* bisa menjadi lebih panjang dibandingkan CSS tradisional, dan beberapa fitur *framework* ini membutuhkan *Javascript* untuk berfungsi. Secara keseluruhan, *Tailwind CSS* merupakan *framework* CSS powerful dan fleksibel yang menawarkan banyak keuntungan bagi developer web modern. Pendekatan utility-first *framework* ini memungkinkan developer untuk membangun UI custom dengan cepat dan mudah, tanpa terpaku pada desain *pre-built*.

## 8. Framework Laravel



**Gambar 2. 13** *framework laravel*

*Laravel* adalah aplikasi *framework* PHP open-source yang menggunakan konsep *Model-View-Controller* (MVC). *Laravel* adalah kerangka kerja untuk rancang bangun web yang menampilkan *sintaks* ekspresif dan elegan. Kerangka kerja ini dirancang untuk membuat proses rancang bangun lebih menyenangkan bagi pengembang tanpa mengorbankan *fungsionalitas* aplikasi, meningkatkan kualitas aplikasi dengan mengurangi

biaya rancang bangun, merampingkan proses pengelolaan proyek, dan meningkatkan produktivitas karyawan dengan menggunakan kode program yang cepat dan terorganisir dengan baik. Salah satu penerapan *Laravel* adalah dalam pengembangan sistem pengelolaan aset laboratorium komputer yang memungkinkan proses pengembangan lebih cepat dengan metode RAD (Rapid Application Development) (Awaluddin, 2020). Hal ini didukung dengan adanya fitur seperti *Command Line Interface* (CLI) Artisan dan manajer paket *Composer* yang mempercepat dan memudahkan pengelolaan proyek (SISITI, 2016).

Beberapa fitur dari *framework Laravel* termasuk menggunakan Command Line Interface (CLI) Artisan, menggunakan manajer paket PHP Composer, dan menulis kode yang cepat, bersih, dan mudah ditafsirkan. Informasi di bawah ini tentang *framework Laravel* dapat membantu pengembang mengelola bug ketika mereka menghasilkan kode dengan memungkinkan mereka untuk menulis kode sesuai dengan standar pengkodean dan tanpa mengorbankan *fungsionalitas* aplikasi.

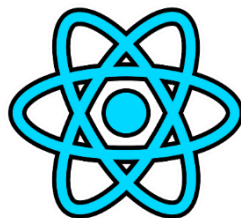
## 9. *phpmyAdmin*



Gambar 2. 14 *phpmyAdmin*

*PhpmyAdmin* adalah sebuah aplikasi open source yang berfungsi untuk memudahkan manajemen *MySQL*. Aplikasi ini memungkinkan pengguna mengelola database *MySQL* melalui antarmuka berbasis web, sehingga operasi seperti membuat *database*, tabel, memasukkan data, menghapus, dan memperbarui data dapat dilakukan dengan lebih mudah tanpa perlu menuliskan perintah SQL secara manual (Wijianto et al., 2018). Dengan menggunakan *PhpmyAdmin*, anda dapat membuat *database*, membuat tabel, meng-insert, menghapus dan meng-update data dengan GUI dan terasa lebih mudah, tanpa perlu mengetikkan perintah SQL secara manual. Karena berbasis web, maka *PhpmyAdmin* dapat dijalankan di banyak OS, selama dapat menjalankan web server dan *MySQL*.

## 10. React



**Gambar 2. 15** *React*

*React.js* adalah *Javascript* library yang digunakan untuk membuat antarmuka pengguna atau *User interface*. *React.js* dibuat oleh Facebook dan pertama kali dirilis pada tahun 2013. *React.js* bukanlah sebuah *framework* atau kerangka kerja, tetapi *API React* dapat digunakan bersama dengan *framework* lain untuk membuat lingkungan kerja yang lebih spesifik, seperti *React Native* untuk membuat aplikasi *mobile*. *React.js* lebih ke pembuatan elemen yang

reusable yaitu bisa digunakan kembali untuk menampilkan data dinamis. Tujuannya adalah untuk mengurangi cacat dan menghindari masalah dalam rancang bangun *User interface* aplikasi. *React* adalah open-source *framework* dengan ribuan komponen, modul, dan dukungan alat.

*React* adalah library yang paling banyak digunakan untuk mengembangkan antarmuka pengguna. *React* Digunakan untuk membangun aplikasi oleh perusahaan dan proyek besar. *React* populer karena faktanya *React* dibangun diatas kemampuan inti rancang bangun web artinya setiap fungsi akan mudah dipahami dan dipelajari pada percobaan pertama (Morgan, 2021).

#### **11. Unified modelling language**

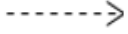
*UML* adalah serangkaian alat yang digunakan untuk menyederhanakan suatu sistem atau perangkat lunak berbasis objek. Singkatan dari Unified Modelling Language, *UML* juga berperan dalam mempermudah rancang bangun aplikasi secara berkelanjutan. Sistem atau aplikasi yang tidak didokumentasikan dengan baik seringkali menghambat proses rancang bangun karena memaksa para pengembang untuk melakukan penyelidikan dan memahami kode program yang kompleks. *UML* juga berfungsi sebagai alat untuk men-transfer pengetahuan tentang suatu sistem atau aplikasi dari satu pengembang ke pengembang lainnya. Penggunaan *UML* dalam tahapan analisis, desain, pengkodean, dan pengujian perangkat lunak membantu pengembang dalam menggambarkan kebutuhan sistem secara efektif (Setiaji & Sastra, 2021). Keberadaan *UML* tidak hanya mempermudah komunikasi

antara para pengembang, tetapi juga memungkinkan orang di luar lingkaran rancang bangun, seperti pihak bisnis atau siapapun, untuk memahami suatu sistem dengan lebih baik.

*UML* (Unified Modelling Language), sebagai standar industri, digunakan untuk menggambarkan, merencanakan, dan mendokumentasikan sistem perangkat lunak. *UML* menyediakan format standar untuk merancang model sistem. Namun, karena *UML* memanfaatkan kelas dan operasi sebagai bagian dari konsep intinya, lebih sesuai digunakan untuk rancang bangun perangkat lunak dengan bahasa berbasis objek seperti C++, Java, C#, atau VB.NET. Meskipun begitu, *UML* masih bisa digunakan untuk memodelkan aplikasi prosedural dalam bahasa seperti VB atau C.

*UML* dikembangkan oleh Object Management Group sejak Januari 1997. Konsepnya meliputi objek, kelas, abstraksi, enkapsulasi, pewarisan dan polimorfisme.

**Tabel 2. 1** *Symbol Use Case Diagram*

| NO. | GAMBAR                                                                              | NAMA                  | KETERANGAN                                                                                                                                                                                 |
|-----|-------------------------------------------------------------------------------------|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1   |    | <i>Actor</i>          | Peran pengguna saat berinteraksi dengan use case.                                                                                                                                          |
| 2   |    | <i>Dependency</i>     | Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri ( <i>independent</i> ) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri ( <i>independent</i> ). |
| 4   |    | <i>Generalization</i> | Menspesifikasikan bahwa <i>use case</i> sumber secara eksplisit.                                                                                                                           |
| 5   |    | <i>Include</i>        | Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.                                                            |
| 6   |  | <i>Extend</i>         | Apa yang menghubungkan antara objek satu dengan objek lainnya.                                                                                                                             |
| 7   |  | <i>Association</i>    | Menspesifikasikan paket yang menampilkan sistem secara terbatas.                                                                                                                           |
| 8   |  | <i>Use Case</i>       | Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor                                                                        |
| 9   |  | <i>Collaboration</i>  | Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).                                          |
| 10  |  | <i>Note</i>           | Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi                                                                                              |

**Tabel 2. 2** *Symbol Class Diagram*

| NO. | GAMBAR                                                                              | NAMA                    | KETERANGAN                                                                                                                                                         |
|-----|-------------------------------------------------------------------------------------|-------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1   |    | <i>Generalization</i>   | Hubungan dimana objek anak ( <i>descendent</i> ) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk ( <i>ancestor</i> ).                |
| 2   |    | <i>Nary Association</i> | Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.                                                                                                        |
| 3   |    | <i>Class</i>            | Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.                                                                                            |
| 4   |    | <i>Collaboration</i>    | Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu <i>actor</i>                                         |
| 5   |    | <i>Realization</i>      | Operasi yang benar-benar dilakukan oleh suatu objek.                                                                                                               |
| 6   |  | <i>Dependency</i>       | Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri ( <i>independent</i> ) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri |
| 7   |  | <i>Association</i>      | Apa yang menghubungkan antara objek satu dengan objek lainnya                                                                                                      |

**Tabel 2. 3** *Symbol Sequence Diagram*

| No | GAMBAR                                                                              | NAMA            | KETERANGAN                                                                                             |
|----|-------------------------------------------------------------------------------------|-----------------|--------------------------------------------------------------------------------------------------------|
| 1  |  | <i>LifeLine</i> | Objek <i>entity</i> , antarmuka yang saling berinteraksi.                                              |
| 2  |  | <i>Message</i>  | Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktivitas yang terjadi |
| 3  |  | <i>Message</i>  | Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktivitas yang terjadi |

**Tabel 2. 4** *Symbol State Chart Diagram*

| No | GAMBAR                                                                             | NAMA                        | KETERANGAN                                                                                                |
|----|------------------------------------------------------------------------------------|-----------------------------|-----------------------------------------------------------------------------------------------------------|
| 1  |   | <i>State</i>                | Nilai atribut dan nilai <i>Link</i> pada suatu waktu tertentu, yang dimiliki oleh suatu objek.            |
| 2  |   | <i>Initial Pseudo State</i> | Bagaimana objek dibentuk atau diawali                                                                     |
| 3  |   | <i>Final State</i>          | Bagaimana objek dibentuk dan dihancurkan                                                                  |
| 4  |   | <i>Transition</i>           | Sebuah kejadian yang memicu sebuah state objek dengan cara memperbaharui satu atau lebih nilai atributnya |
| 5  |   | <i>Association</i>          | Apa yang menghubungkan antara objek satu dengan objek lainnya.                                            |
| 6  |  | <i>Node</i>                 | Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.            |

**Tabel 2. 5** *Symbol Activity Diagram*

| No. | GAMBAR                                                                              | NAMA                       | KETERANGAN                                                                                |
|-----|-------------------------------------------------------------------------------------|----------------------------|-------------------------------------------------------------------------------------------|
| 1   |  | <i>Activity</i>            | Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain |
| 2   |  | <i>Action</i>              | State dari sistem yang mencerminkan eksekusi dari suatu aksi                              |
| 3   |  | <i>Initial Node</i>        | Bagaimana objek dibentuk atau diawali.                                                    |
| 4   |  | <i>Activity Final Node</i> | Bagaimana objek dibentuk dan dihancurkan                                                  |
| 5   |  | <i>Fork Node</i>           | Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran                      |

## B. Kajian hasil penelitian terdahulu

Angga Pratama (2021) dalam penelitiannya yang berjudul “Pengembangan Aplikasi Pembuatan CV Otomatis Berbasis Web” mengembangkan sebuah aplikasi web yang membantu pengguna dalam membuat CV secara otomatis. Aplikasi ini memanfaatkan form untuk mengumpulkan data pribadi, pendidikan, dan pengalaman kerja, yang kemudian diolah menjadi CV dalam format PDF. Penelitian ini menekankan kemudahan penggunaan dan personalisasi template CV, namun aplikasi tersebut masih memiliki keterbatasan karena tidak memanfaatkan teknologi *Progressive Web Apps* (PWA), sehingga hanya bisa digunakan saat terhubung ke internet.

Dian Septiani (2022) dalam penelitian yang berjudul “Penggunaan *Progressive Web Apps* untuk Aplikasi Berbasis Web yang Dapat Diakses Secara Offline” membahas pengembangan aplikasi berbasis PWA yang dapat diakses secara offline. Fokus penelitian ini adalah pada manajemen data mahasiswa, dengan PWA memberikan pengalaman aplikasi seperti aplikasi native serta kemampuan akses tanpa koneksi internet melalui *cache* lokal. Hasil penelitian menunjukkan bahwa penerapan PWA pada aplikasi berbasis web mampu meningkatkan performa dan pengalaman pengguna.

Ridwan Hidayat (2023) dalam penelitiannya yang berjudul “Rancang Bangun Aplikasi Pembuat CV Berbasis *Mobile* dengan Implementasi *Progressive Web Apps* (PWA)” mengembangkan sebuah aplikasi pembuat CV yang dapat diakses baik di desktop maupun perangkat *mobile*. Aplikasi ini menggunakan teknologi PWA sehingga pengguna dapat membuat CV secara offline dengan data

yang disimpan di perangkat lokal. Selain itu, aplikasi ini dilengkapi dengan antarmuka yang responsif dan fitur service worker untuk *cache* management serta notifikasi push.

Siti Fadhilah (2020) dalam penelitian yang berjudul “Pemanfaatan PWA untuk Peningkatan Pengalaman Pengguna pada Aplikasi Pembuatan CV Otomatis” mengembangkan aplikasi PWA untuk pembuatan CV yang dapat digunakan secara offline. Aplikasi ini menekankan kemudahan akses, terutama untuk pengguna perangkat *mobile*. Dengan berbagai pilihan template CV yang dapat diunduh dalam format PDF, aplikasi ini berhasil meningkatkan aksesibilitas dan pengalaman pengguna dengan teknologi PWA yang memungkinkan penggunaan tanpa koneksi internet.

Ahmad Saifullah (2024) dalam penelitiannya yang berjudul “Sistem Pembuatan CV Berbasis *Progressive Web Apps* untuk Meningkatkan Aksesibilitas Pengguna” meneliti implementasi PWA pada aplikasi pembuatan CV. Aplikasi ini dirancang untuk pengguna di daerah dengan akses internet terbatas dan memungkinkan pembuatan CV secara offline. Fitur pencadangan otomatis ke cloud saat perangkat terhubung kembali ke internet menjadi salah satu inovasi penting dalam penelitian ini, yang meningkatkan keamanan dan aksesibilitas pengguna.

Yulianto Andika (2023) dalam penelitiannya yang berjudul “Implementasi Fitur Personal Branding pada Aplikasi Pembuatan CV Berbasis PWA” mengembangkan aplikasi CV yang memanfaatkan PWA dengan penekanan pada

fitur personal branding. Pengguna dapat menambahkan portofolio, tautan media sosial, serta menyesuaikan warna dan layout CV. PWA memungkinkan aplikasi ini tetap dapat diakses secara offline dan fleksibel untuk digunakan di berbagai perangkat, memberikan solusi yang tepat bagi pengguna yang ingin memperkuat citra diri mereka melalui CV.

### C. Kerangka pikir

*Curriculum Vitae* (CV) adalah dokumen penting yang memuat informasi mengenai pendidikan, pengalaman kerja, keterampilan, dan prestasi seseorang. CV yang dibuat dengan baik dan informatif dapat meningkatkan peluang seseorang dalam dunia kerja yang kompetitif. Namun, pembuatan CV secara manual seringkali memakan waktu, membosankan, dan membutuhkan perhatian pada aspek format dan konsistensi informasi, yang dapat menjadi hambatan bagi banyak orang.

Generasi muda, terutama lulusan baru, sering mengalami kesulitan dalam menyusun CV profesional yang sesuai dengan standar perusahaan atau industri. Dengan perkembangan teknologi yang pesat, kebutuhan akan CV yang menarik secara visual dan dapat diakses di berbagai perangkat juga semakin meningkat. Selain itu, perusahaan banyak menggunakan sistem pelacakan pelamar (*Applicant Tracking System/ATS*), sehingga CV harus dioptimalkan agar dapat lolos seleksi awal.

Proses pembuatan CV perlu disederhanakan dengan menggunakan teknologi modern. Oleh karena itu, solusi yang relevan adalah rancang bangun sistem pembuatan CV otomatis berbasis *Progressive Web Apps* (PWA). PWA memungkinkan aplikasi dapat diakses dengan mudah di berbagai perangkat tanpa perlu diunduh, serta menawarkan keunggulan seperti kecepatan, akses yg praktis, dan tampilan yang menyerupai aplikasi *mobile native*.

Direncanakan Rancang bangun Sistem Pembuatan CV Otomatis  
Berbasis *Progressive Web Apps*.

## BAB III

### METODE PENELITIAN

#### A. Jenis penelitian

Metode yang digunakan dalam penelitian ini adalah **metode penelitian dan pengembangan** (*Research and Development/R&D*). Jenis penelitian ini bertujuan untuk merancang, mengembangkan, dan menguji **sistem pembuatan Curriculum Vitae (CV) otomatis Berbasis Progressive Web Apps (PWA)**. Proses penelitian R&D melibatkan tahap analisis kebutuhan, perancangan sistem, pengembangan prototipe, uji coba, evaluasi, dan perbaikan terhadap sistem yang dikembangkan hingga mencapai produk yang siap digunakan. Metode ini cocok digunakan karena fokus penelitian pada pengembangan aplikasi baru yang berfungsi untuk mempermudah pengguna dalam membuat CV secara otomatis dengan teknologi PWA.

#### B. Waktu penelitian

Penelitian ini akan berlangsung selama  $\pm 1$  (satu) bulan .

#### C. Alat dan bahan penelitian

##### 1. Perangkat keras

Laptop Macbook air M1 dengan spesifikasi hardware:

- *Processor* : M1
- *Installed RAM* : 8GB
- *SSD* : 256GB

## 2. Perangkat lunak

- MacOS
- Visual studio code
- Xampp
- PHP
- *Javascript*

### D. Rancangan sistem

#### 1. Sistem yang berjalan

Sistem yang akan dikembangkan terdiri dari beberapa komponen utama, yaitu *frontend*, *backend*, dan fitur-fitur utama. Antarmuka pengguna (*frontend*) akan dibangun menggunakan *React.js* dengan *Material Tailwind* untuk memastikan desain yang responsif dan ramah pengguna. Formulir dinamis akan disediakan untuk menyesuaikan inputan pengguna dengan format CV yang diinginkan. *Backend* akan menggunakan *framework Laravel* untuk mengelola data dan logika bisnis, dengan *MySQL* sebagai basis data untuk menyimpan informasi pengguna dan template CV.

Fitur utama sistem ini meliputi pendaftaran dan *Login* untuk mengamankan data pribadi, pembuatan CV melalui formulir input untuk data pribadi, pendidikan, pengalaman kerja, dan keterampilan, serta pilihan template CV yang beragam. Pengguna juga dapat melihat pratinjau CV mereka dan mengunduhnya dalam format PDF, serta meng*Edit* dan memperbarui CV yang telah dibuat.

Tahapan rancang bangun sistem ini meliputi analisis kebutuhan, desain sistem, implementasi, pengujian, serta penerapan dan evaluasi. Analisis kebutuhan dilakukan dengan mengumpulkan data dari hasil observasi dan wawancara untuk menentukan kebutuhan fungsional dan non-fungsional sistem. Desain sistem mencakup pembuatan wireframe dan mockup antarmuka pengguna serta desain arsitektur sistem dan *database*. Implementasi melibatkan rancang bangun *frontend* dan *backend* serta integrasi keduanya. Pengujian dilakukan untuk memastikan kualitas dan *fungsionalitas* sistem melalui pengujian unit, integrasi, dan sistem. Setelah itu, sistem akan diterapkan pada lingkungan produksi dan dievaluasi berdasarkan umpan balik dari pengguna untuk perbaikan lebih lanjut.

sistem pembuatan CV yang dikembangkan diharapkan dapat memudahkan pengguna dalam menyusun CV yang profesional dan sesuai dengan standar industri. Penelitian ini juga diharapkan memberikan kontribusi pada rancang bangun teknologi yang mendukung proses rekrutmen. Lampiran penelitian ini mencakup *diagram* alir sistem yang menampilkan alur kerja dari pendaftaran hingga pembuatan dan pengunduhan CV, serta contoh template CV yang dapat dipilih oleh pengguna. Referensi yang digunakan dalam penelitian ini disertakan dalam daftar literatur dan sumber.

## 2. Sistem yang diusulkan



Gambar 3. 1 *Struktur database*

Struktur database ini dirancang untuk mengelola data pengguna dan informasi terkait CV mereka. Tabel utama adalah *user*, yang menyimpan informasi dasar pengguna seperti nama, *email*, dan kata sandi. Setiap

pengguna dapat memiliki satu atau lebih entri di tabel biodatas, yang berisi informasi detail CV mereka, seperti nama, asal, kontak, keterampilan, pengalaman, pendidikan, sertifikat, dan foto. Hubungan antara biodatas dan user bersifat *many-to-one*, di mana setiap entri di biodatas terhubung dengan satu pengguna melalui kolom *user\_id*.

Selain itu, terdapat tabel *password\_reset\_tokens* yang menyimpan token pemulihan kata sandi, dengan hubungan ke user berdasarkan *email*. Tabel *cache* digunakan untuk menyimpan data sementara dengan sistem kunci dan nilai untuk meningkatkan kinerja aplikasi. Struktur ini memungkinkan sistem untuk mengelola informasi CV secara efisien, memastikan autentikasi pengguna, serta mendukung mekanisme pemulihan akun.

## **E. Teknik Pengumpulan Data**

### **1. Studi pustaka**

Studi pustaka merupakan strategi pengumpulan data dari berbagai macam sumber literatur seperti, buku, jurnal, e-book, youtube, .ataupun website-website yang relevan dengan masalah yang diidentifikasi.

### **2. Observasi**

Observasi merupakan teknik pengumpulan data dengan cara mengamati secara langsung maupun tidak langsung. Observais tidak langsung dilakukan dengan cara mengamati objek penelitian melalui media tertentu, seperti video atau foto.

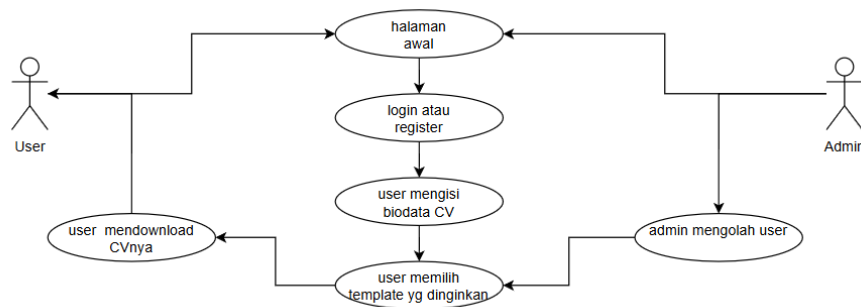
### **3. Dokumentasi**

Pada tahap ini dilakukan mengumpulkan data yang berkaitan dalam bentuk dokumen. Dokumen-dokumen tersebut dapat berupa dokumen tertulis, dokumen elektronik, atau dokumen audiovisual.

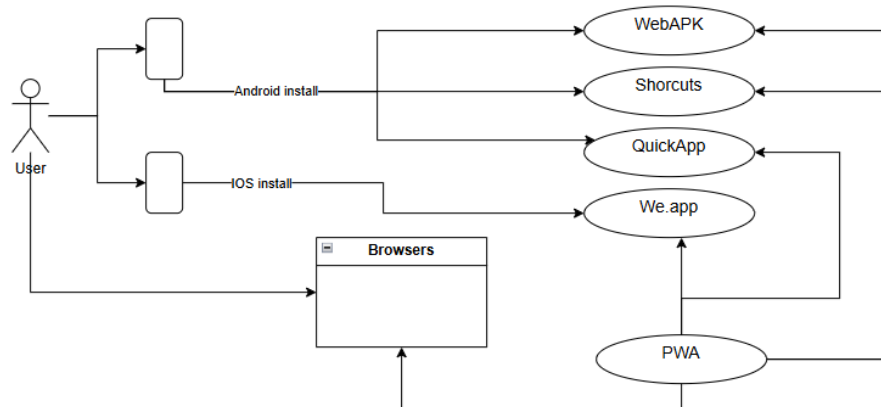
## BAB IV

### PEMBAHASAN DAN HASIL

#### A. Usecase diagram



Gambar 4. 1 Usecase Diagram Alur System



Gambar 4. 2 Usecase Diagram PWA

## B. Tabel Usecase

Tabel 4. 1 tabel penjelasan *usecase alur system*

| No. | Use Case            | Deskripsi                                                                                     | Aktor          | Alur Kerja                                                                                        |
|-----|---------------------|-----------------------------------------------------------------------------------------------|----------------|---------------------------------------------------------------------------------------------------|
| 1   | Halaman Awal        | Menampilkan halaman awal aplikasi untuk akses ke fitur login atau registrasi.                 | User,<br>Admin | - User mengakses aplikasi melalui browser.<br>- Sistem menampilkan halaman utama.                 |
| 2   | Login atau Register | User atau admin masuk ke sistem menggunakan akun atau membuat akun baru jika belum terdaftar. | User,<br>Admin | - User memasukkan kredensial (email dan password) atau mendaftar.<br>- Sistem memverifikasi akun. |
| 3   | Mengisi Biodata CV  | User mengisi informasi pribadi untuk pembuatan CV, seperti nama, email, alamat, dan keahlian. | User           | - User mengisi form biodata.<br>- Sistem menyimpan data ke database.                              |
| 4   | Memilih Template CV | User memilih template desain CV yang tersedia untuk menyesuaikan dengan preferensinya.        | User           | - User melihat daftar template CV.<br>- User memilih template.                                    |
| 5   | Mendownload CV      | User mengunduh CV yang telah dibuat dalam format file yang diinginkan (misalnya, PDF).        | User           | - User mengklik tombol unduh.<br>- Sistem menghasilkan file CV dan memberikannya untuk diunduh.   |
| 6   | Mengelola User      | Admin mengelola data user, termasuk menambah, mengedit, atau menghapus informasi user.        | Admin          | - Admin melihat daftar user.<br>- Admin dapat menambah, mengedit, atau menghapus data user.       |

Tabel 4. 2 tabel penjelasan *Usecase PWA*

| No. | Use Case  | Deskripsi                                                                                        | Aktor | Alur Kerja                                                                                          |
|-----|-----------|--------------------------------------------------------------------------------------------------|-------|-----------------------------------------------------------------------------------------------------|
| 1   | WebAPK    | Fitur yang memungkinkan pengguna untuk menginstal aplikasi PWA sebagai APK di perangkat Android. | User  | - User mengakses aplikasi melalui browser di Android.<br>- Sistem memberikan opsi instalasi WebAPK. |
| 2   | Shortcuts | Memberikan pintasan (shortcut) di perangkat pengguna untuk mengakses aplikasi dengan cepat.      | User  | - User menambahkan shortcut ke layar utama perangkat.<br>- Shortcut mengarah ke PWA.                |

| No. | Use Case | Deskripsi                                                                                       | Aktor | Alur Kerja                                                                                     |
|-----|----------|-------------------------------------------------------------------------------------------------|-------|------------------------------------------------------------------------------------------------|
| 3   | QuickApp | Fitur yang mengoptimalkan pengalaman pengguna dengan akses instan tanpa instalasi penuh.        | User  | - User mengakses aplikasi melalui QuickApp.<br>- Sistem memberikan pengalaman aplikasi instan. |
| 4   | We.app   | Fitur untuk perangkat iOS yang memungkinkan pengguna menginstal aplikasi melalui format WebApp. | User  | - User menggunakan perangkat iOS.<br>- Sistem menawarkan opsi instalasi We.app.                |
| 5   | PWA      | Progressive Web Apps yang memungkinkan akses langsung dari browser tanpa instalasi.             | User  | - User mengakses aplikasi melalui browser.<br>- Sistem memberikan pengalaman PWA standar.      |
| 6   | Browsers | Media utama untuk mengakses aplikasi sebelum instalasi atau penggunaan fitur lain.              | User  | - User membuka aplikasi menggunakan browser.<br>- Sistem menampilkan versi awal aplikasi.      |

### C. Analisis aliran data *UML*

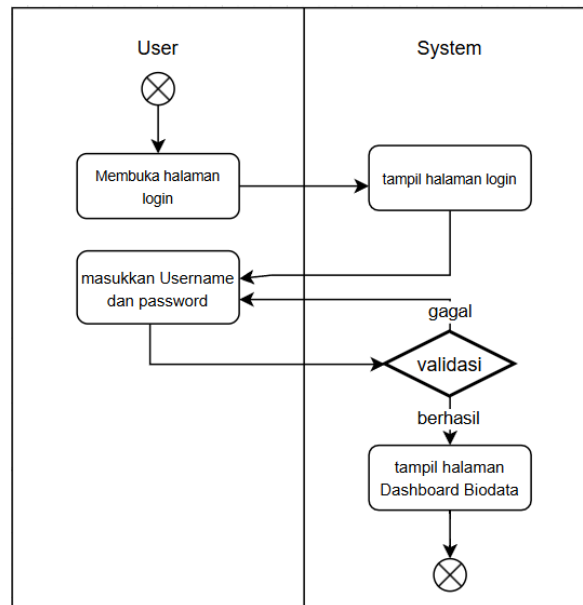
Dalam bagian ini, pembahasan akan berfokus pada dua *diagram UML* penting untuk aplikasi Pembuatan CV otomatis, yaitu *Activity diagram* dan *Sequence diagram*, yang memperlihatkan aliran data dan interaksi.

#### 1. *Activity Diagram*

*Activity diagram* berfungsi sebagai representasi visual aliran proses bisnis dan use case. *Diagram* ini menunjukkan aktivitas-aktivitas yang terjadi dalam sistem dan urutan eksekusinya.

### a. Activity Diagram Login Register

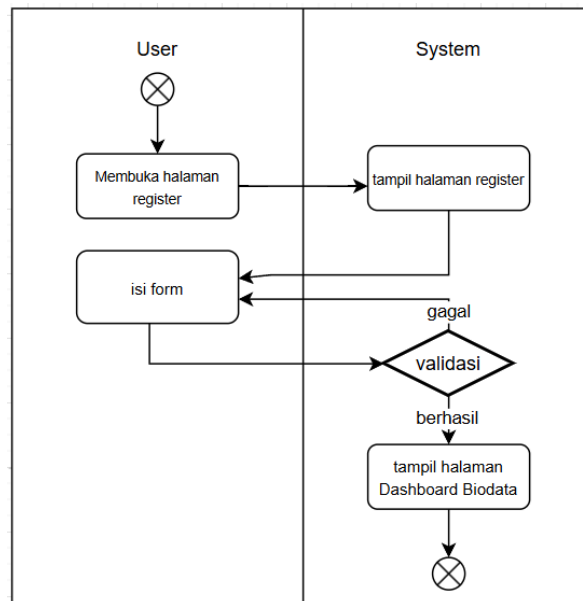
#### 1) Activity Diagram Login



Gambar 4. 3 Activity Diagram Login

Gambar 4.3 memperlihatkan *Activity Diagram* proses *Login* pengguna pada aplikasi Pembuatan CV otomatis. Proses dimulai ketika pengguna mengakses halaman *Login*, kemudian mengisi formulir dengan *Username* dan *password*. Sistem memvalidasi *kredensial* tersebut dan membandingkannya dengan data di database. Jika valid, sistem mengautentikasi pengguna dan mengalihkan ke halaman utama aplikasi. Jika tidak valid, sistem menampilkan pesan kesalahan. *Diagram* ini mengilustrasikan alur proses *Login* yang efisien dan aman, memastikan pengguna dapat mengakses aplikasi dengan mudah.

## 2) Activity Diagram Register

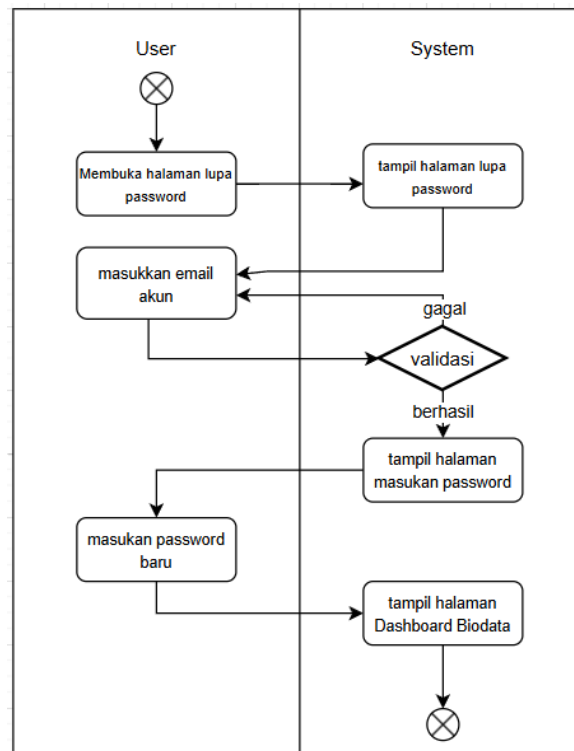


Gambar 4. 4 Activity Diagram Register

Gambar 4.4 menampilkan *Activity Diagram* proses registrasi pengguna pada aplikasi Pembuatan CV otomatis. Proses dimulai ketika pengguna mengakses halaman registrasi, kemudian mengisi formulir dengan informasi pribadi dan *kredensial*. Sistem memvalidasi input tersebut dan memeriksa ketersediaan *Username*. Jika valid, sistem menyimpan data pengguna dan mengirimkan konfirmasi melalui *email*. Setelah konfirmasi, akun pengguna aktif dan siap digunakan. *Diagram* ini menggambarkan alur proses registrasi yang sistematis dan aman

untuk memastikan integritas data pengguna.

### 3) Activity Diagram Lupa password



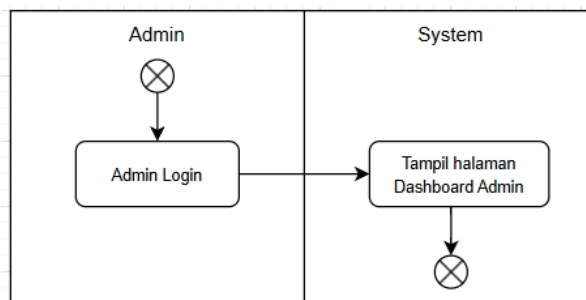
Gambar 4. 5 Activity Lupa password

Gambar 4.5 menampilkan *Activity Diagram* proses pemulihan kata sandi (*lupa password*) pada aplikasi Pembuatan CV otomatis. Proses dimulai ketika pengguna mengakses halaman *lupa password*, kemudian memasukkan alamat *email* terdaftar. Sistem memvalidasi *email* dan mengirimkan kode verifikasi melalui *email*. Pengguna memasukkan kode verifikasi dan membuat kata sandi baru. Sistem memperbarui kata sandi dan mengaktifkan akun. *Diagram* ini mengilustrasikan alur proses pemulihan kata sandi yang aman dan efisien untuk membantu

pengguna mengakses akunnya kembali.

## b. *Activity Diagram Admin*

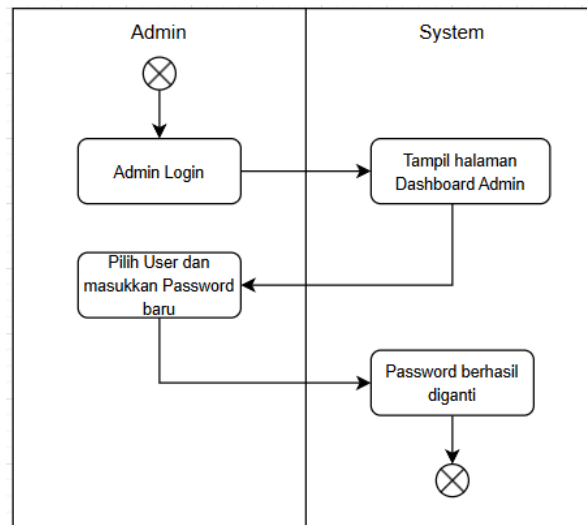
### 1) *Activity Diagram Dashboard Admin*



Gambar 4. 6 *Activity Diagram Dashboard Admin*

Gambar 4.6 memperlihatkan *Activity Diagram Dashboard Admin* pada aplikasi Pembuatan CV Otomatis. Proses dimulai ketika *Admin Login*, kemudian mengakses *Dashboard Admin*. *Admin* dapat melakukan beberapa aktivitas seperti mengatur pengguna, memantau statistik dan mengelola konfigurasi aplikasi. Sistem memvalidasi hak akses *Admin* sebelum melakukan perubahan. *Diagram* ini menggambarkan alur proses pengelolaan aplikasi yang efisien dan aman.

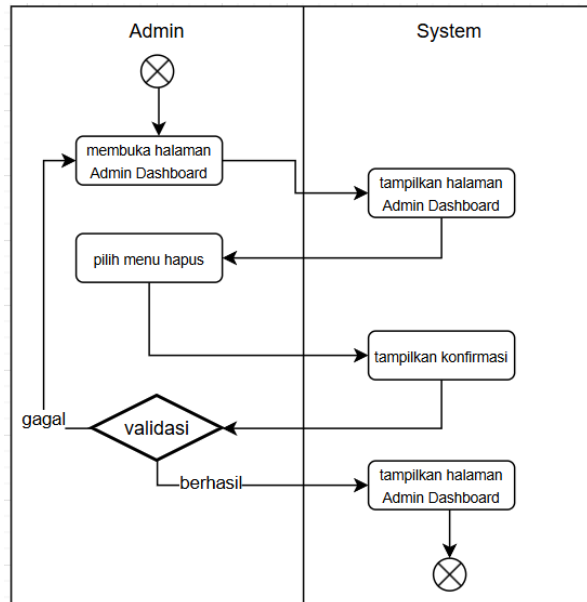
## 2) Activity Diagram Ganti password User



Gambar 4. 7 Diagram ganti password User

Gambar 4.7 menampilkan *Activity Diagram* perubahan kata sandi pengguna oleh *Admin* pada aplikasi Pembuatan CV Otomatis. Proses dimulai ketika *Admin* mengakses *Dashboard*, memilih pengguna, dan mengubah kata sandi. Sistem memvalidasi hak akses *Admin* dan memperbarui kata sandi pengguna. Kemudian, sistem mengirimkan notifikasi perubahan kata sandi kepada pengguna. *Diagram* ini mengilustrasikan alur proses perubahan kata sandi yang aman dan efisien.

### 3) Activity Diagram Hapus User

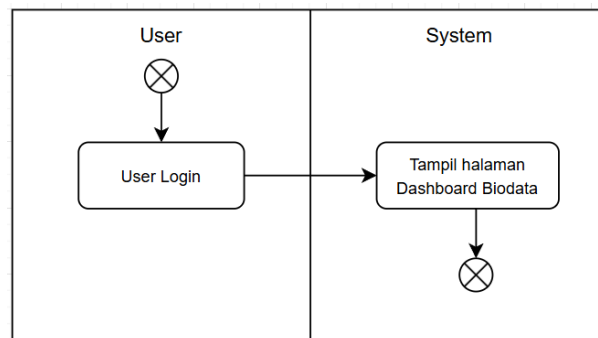


Gambar 4. 8 Activity Diagram Hapus User

Gambar 4.8 memperlihatkan Activity Diagram penghapusan pengguna oleh Admin pada aplikasi Pembuatan CV Otomatis. Proses dimulai ketika Admin mengakses Dashboard, memilih pengguna yang akan dihapus, dan mengkonfirmasi penghapusan. Sistem memvalidasi hak akses Admin, menghapus data pengguna, dan memperbarui basis data. Kemudian, sistem mencatat aktivitas penghapusan dalam log. Diagram ini menggambarkan alur proses penghapusan pengguna yang aman dan sistematis.

### c. *Activity Diagram User*

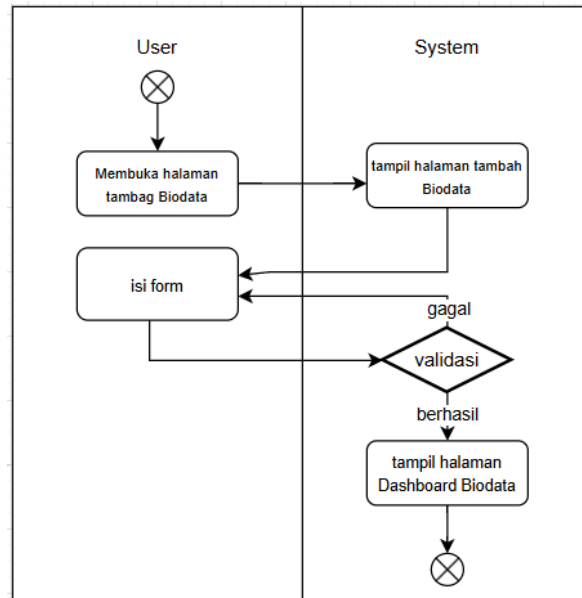
#### 1) *Activity Diagram Dashboard Biodata*



Gambar 4. 9 *Activity Diagram Dashboard Biodata*

Gambar 4.9 menampilkan *Activity Diagram Dashboard* biodata pengguna pada aplikasi *Dashboard Biodata*. Proses dimulai ketika pengguna mengakses *Dashboard* biodata, kemudian memilih opsi untuk melihat, mengubah atau menghapus data. Sistem memvalidasi hak akses pengguna dan menampilkan atau memperbarui data biodata. *Diagram* ini mengilustrasikan alur proses pengelolaan biodata yang efisien dan aman, memastikan integritas data pengguna.

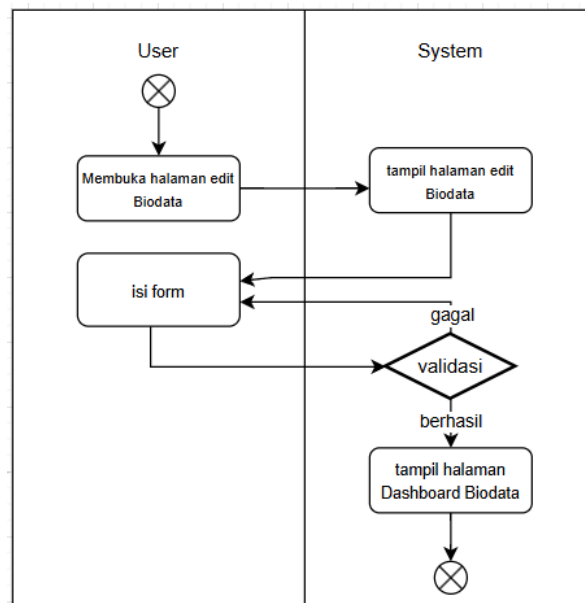
## 2) Activity Diagram Tambah Biodata



Gambar 4. 10 Activity Diagram Tambah Biodata

Gambar 4.10 memperlihatkan *Activity Diagram* penambahan biodata. Proses dimulai dengan pengguna mengakses halaman tambah biodata, mengisi formulir, dan mengunggah dokumen pendukung. Sistem memvalidasi input, memeriksa kesesuaian data, dan menyimpan biodata. Setelah validasi, sistem mengkonfirmasi penambahan biodata dan memperbarui profil pengguna. *Diagram* ini menggambarkan proses efisien dan aman untuk mengelola biodata.

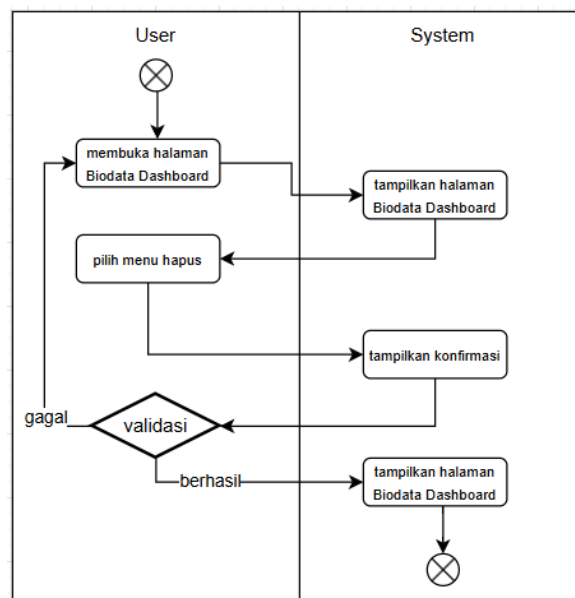
### 3) Activity Diagram Edit Biodata



Gambar 4. 11 Activity Diagram Edit Biodata

Gambar 4.11 menampilkan *Activity Diagram* pengEditan biodata. Proses dimulai dengan pengguna mengakses halaman *Edit* biodata, memperbarui informasi, dan mengunggah dokumen pendukung. Sistem memvalidasi perubahan, memeriksa kesesuaian data, dan memperbarui profil pengguna. Setelah validasi, sistem mengkonfirmasi perubahan biodata dan mencatat aktivitas dalam log. *Diagram* ini menggambarkan proses pengEditan biodata yang efisien dan aman.

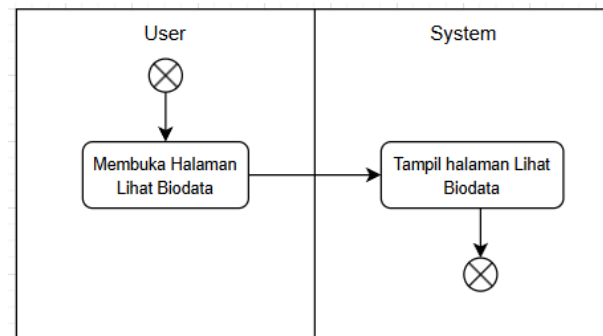
#### 4) Activity Diagram Hapus Biodata



Gambar 4. 12 Activity Diagram Hapus Biodata

Gambar 4.12 memperlihatkan *Activity Diagram* penghapusan biodata. Proses dimulai dengan pengguna mengakses halaman biodata, memilih data untuk dihapus, dan mengkonfirmasi penghapusan. Sistem memvalidasi hak akses, menghapus data, dan memperbarui profil pengguna. Selanjutnya, sistem mencatat aktivitas penghapusan dalam log untuk memastikan integritas data. *Diagram* ini menggambarkan proses penghapusan biodata yang aman dan sistematis.

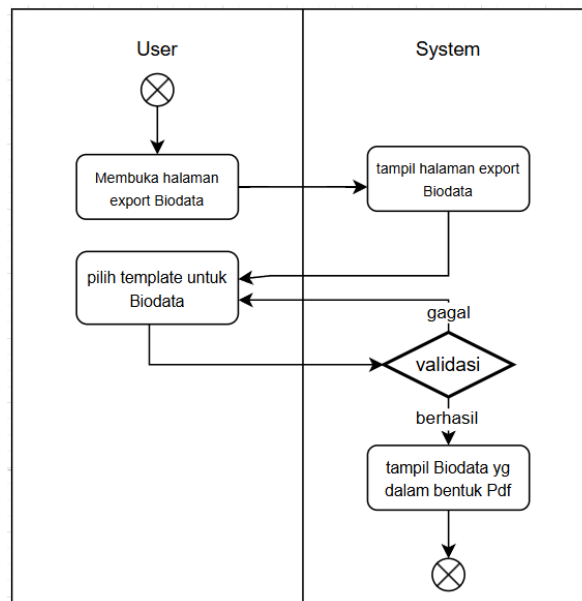
### 5) *Activity Diagram* Lihat detail Biodata



Gambar 4. 13 *Activity Diagram* Lihat detail Biodata

Gambar 4.13 menampilkan *Activity Diagram* melihat detail biodata .Proses dimulai dengan pengguna mengakses halaman biodata, memilih data untuk dilihat, dan sistem menampilkan detail biodata. Sistem memvalidasi hak akses dan memastikan integritas data. *Diagram* ini menggambarkan proses akses biodata yang efisien, aman dan terstruktur, memudahkan pengguna mengakses informasi.

## 6) Activity Diagram Export Biodata



Gambar 4. 14 Activity Diagram Export Biodata

Gambar 4.14 memperlihatkan Activity Diagram ekspor biodata. Proses dimulai dengan pengguna mengakses halaman biodata, memilih opsi ekspor, dan memilih template. Sistem kemudian memproses data dan menghasilkan file PDF. Pengguna menerima file biodata dalam format PDF, memudahkan penyimpanan dan penggunaan data. Diagram ini menggambarkan proses ekspor biodata yang efisien dan sistematis.

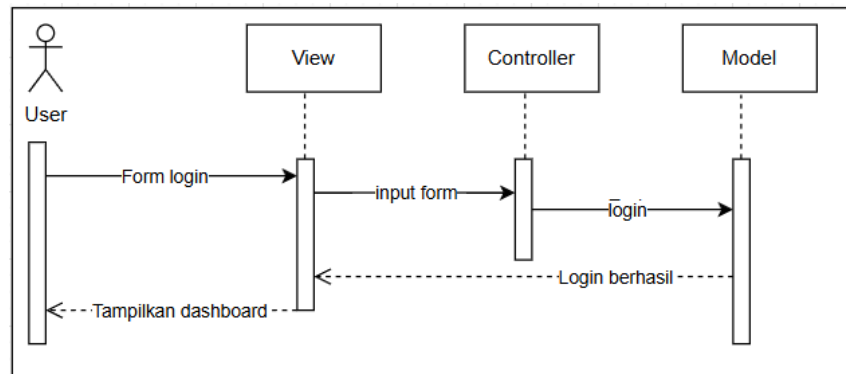
## 2. Sequence Diagram

Sequence diagram merupakan diagram yang menggambarkan interaksi antara objek-objek dalam sistem secara berurutan berdasarkan waktu. Diagram

ini menunjukkan bagaimana objek-objek saling bertukar pesan atau data untuk menyelesaikan suatu tugas atau *use case*.

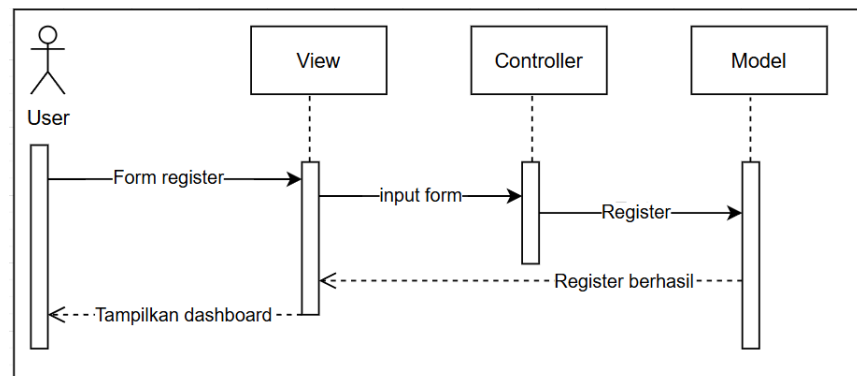
**a. Sequence Diagram Login Register**

**1) Sequence Diagram Login**



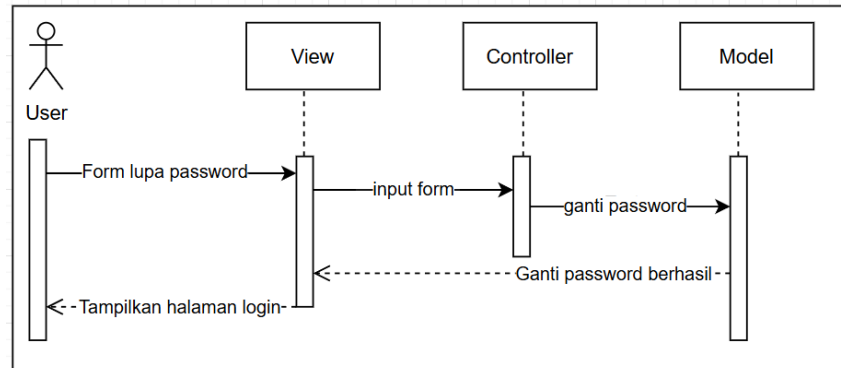
Gambar 4. 15 Sequence Diagram Login

**2) Sequence Diagram Register**



Gambar 4. 16 Sequence Diagram Register

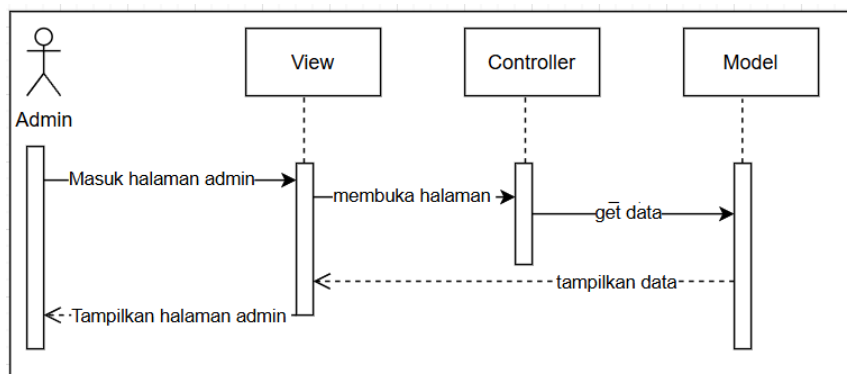
### 3) Sequence Diagram Lupa password



Gambar 4. 17 Sequence Diagram lupa password

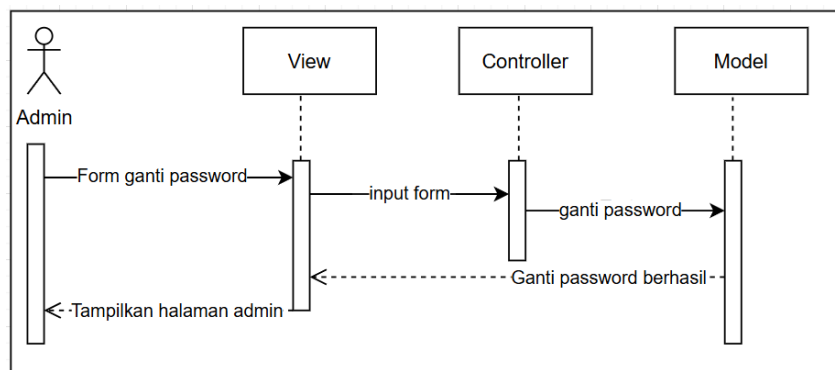
## b. Sequence Diagram Admin

### 1) Sequence Diagram Dashboard Admin



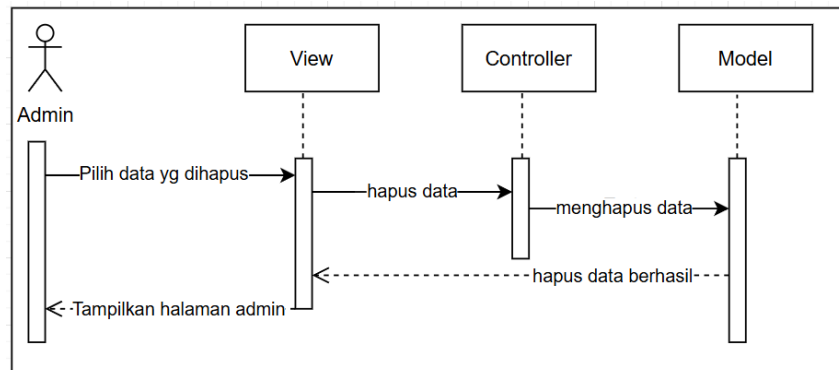
Gambar 4. 18 Sequence Diagram Dashboard Admin

### 2) Sequence Diagram Ganti password User



Gambar 4. 19 Sequence Diagram Ganti password User

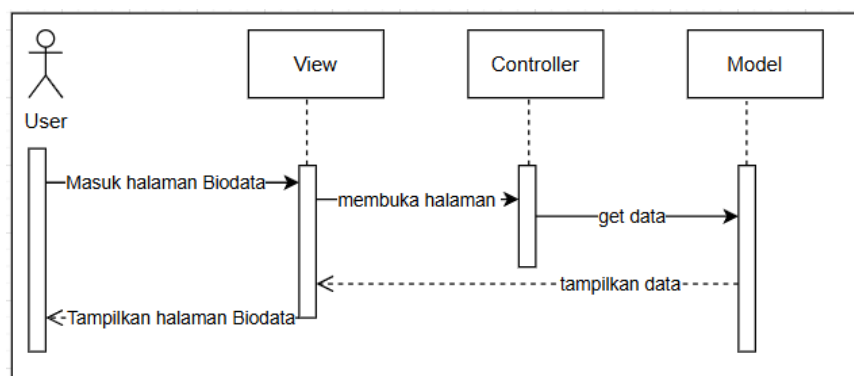
### 3) *Sequence Diagram Hapus User*



Gambar 4. 20 *Sequence Diagram Hapus User*

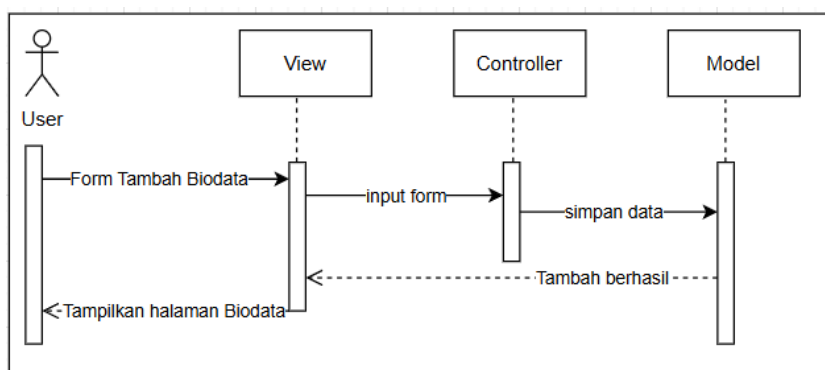
### c. *Sequence Diagram User*

#### 1) *Sequence Diagram Dashboard Biodata*



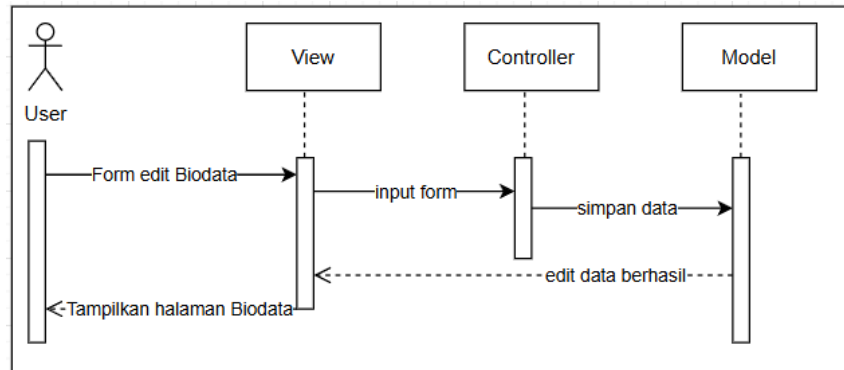
Gambar 4. 21 *Sequence Diagram Dashboard Biodata*

#### 2) *Sequence Diagram tambah Biodata*



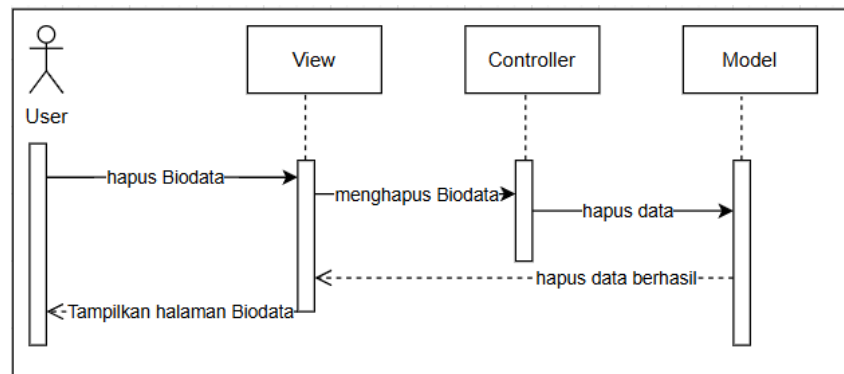
Gambar 4. 22 *Sequence Diagram Tambah Biodata*

### 3) Sequence Diagram Edit Biodata



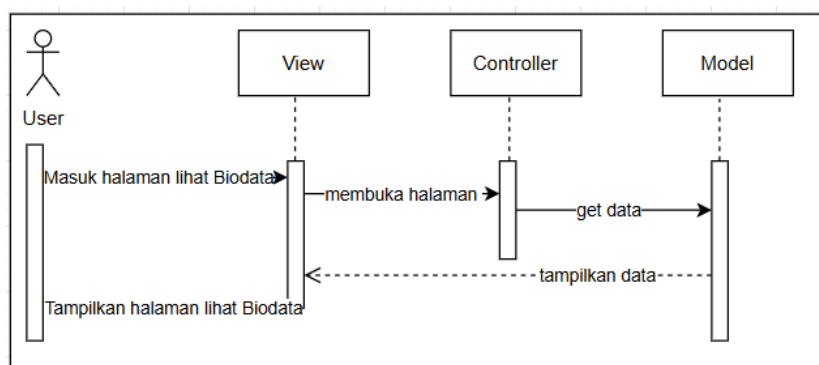
Gambar 4. 23 Sequence Diagram Edit Biodata

### 4) Sequence Diagram Hapus Biodata



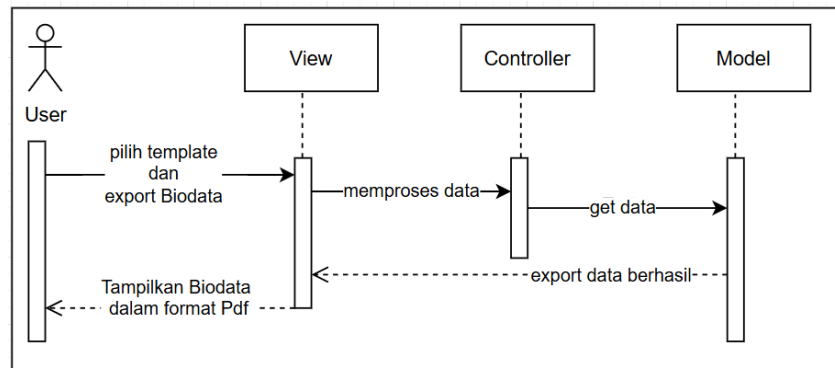
Gambar 4. 24 Sequence Diagram hapus Biodata

### 5) Sequence Diagram Lihat detail Biodata



Gambar 4. 25 Sequence Diagram Lihat detail Biodata

## 6) Sequence Diagram Export Biodata

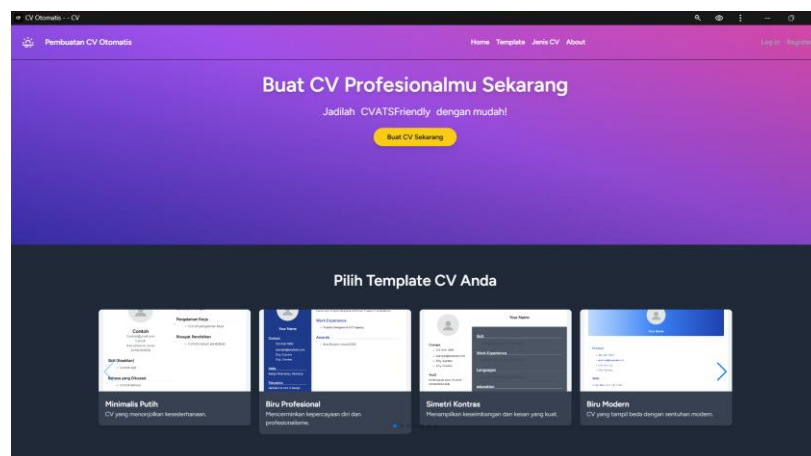


Gambar 4. 26 Sequence Diagram Export Biodata

## D. Detail sistem

### 1. Halaman Awal

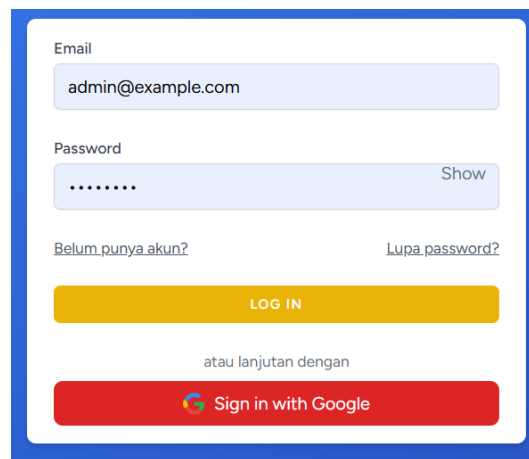
Halaman awal aplikasi Pembuatan CV Otomatis merupakan antarmuka utama yang menampilkan menu navigasi, fitur-fitur utama dan informasi penting, memudahkan pengguna mengakses dan menjelajahi aplikasi.



Gambar 4. 27 halaman Awal

## 2. Halaman *Login*

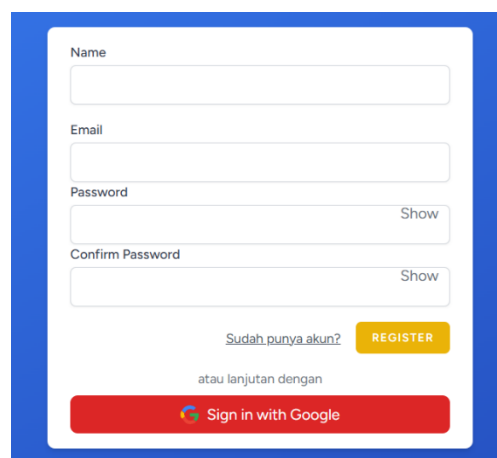
Halaman *Login* aplikasi Pembuatan CV Otomatis adalah antarmuka untuk pengguna memasukkan *kredensial* akun, seperti *Username* dan kata sandi, untuk mengakses fitur dan konten aplikasi secara aman.

The image shows a login form with a blue border. It contains an 'Email' field with the text 'admin@example.com', a 'Password' field with masked characters and a 'Show' button, a 'LOG IN' button, and a 'Sign in with Google' button. There are also links for 'Belum punya akun?' and 'Lupa password?'.

Gambar 4. 28 Halaman *Login*

## 3. Halaman *Register*

Halaman *Register* aplikasi Pembuatan CV Otomatis adalah antarmuka untuk pengguna baru mendaftarkan akun. Pengguna diminta mengisi formulir *Username*, kata sandi untuk membuat akun.

The image shows a registration form with a blue border. It contains fields for 'Name', 'Email', 'Password', and 'Confirm Password', each with a 'Show' button. There is a 'REGISTER' button and a 'Sign in with Google' button. There is also a link for 'Sudah punya akun?'.

Gambar 4. 29 Halaman *Register*

#### 4. Halaman Lupa *password*

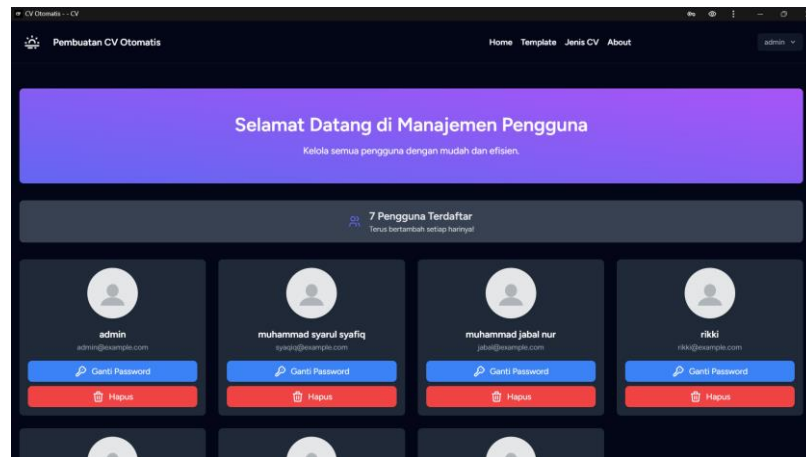
Halaman "Lupa *Password*" aplikasi Pembuatan CV Otomatis memungkinkan pengguna mengatur ulang kata sandi akun. Pengguna harus memasukkan alamat *email* atau *Username* terdaftar, kemudian sistem mengirimkan tautan reset kata sandi untuk mengatur ulang kata sandi baru dan mengakses akun.

Gambar 4. 30 halaman Lupa dan ganti *password*

#### 5. Halaman *Admin Dashboard*

Halaman *Admin Dashboard* aplikasi pembuatan CV otomatis menampilkan antarmuka pengelolaan aplikasi. Fitur-fitur seperti pengelolaan pengguna,

statistik penggunaan, dan pengaturan aplikasi memudahkan *Administrator* mengawasi dan mengoptimalkan kinerja aplikasi.



Gambar 4. 31 Halaman *Admin Dashboard*

## 6. Halaman *Admin Ganti password*

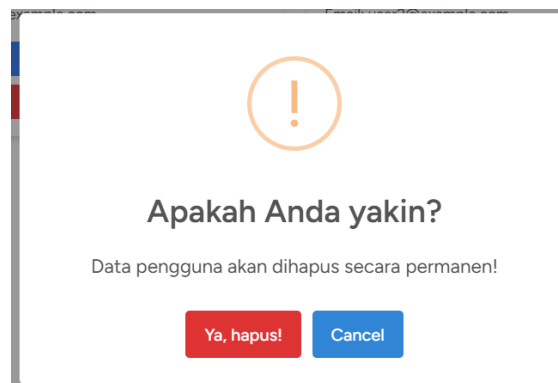
Halaman *Admin Ganti Password* aplikasi memungkinkan *Administrator* mengubah kata sandi akun.

Gambar 4. 32 Halaman *Admin Ganti password*

## 7. Halaman *Admin Hapus User*

Halaman *Admin Hapus User* aplikasi pembuatan CV otomatis memungkinkan *Administrator* menghapus akun pengguna. Fitur ini

memudahkan pengelolaan pengguna dan memastikan keamanan data dengan menghapus akun yang tidak aktif atau melanggar kebijakan.



Gambar 4. 33 Halaman *Admin Hapus User*

## 8. Halaman Biodata *Dashboard*

Halaman Biodata *Dashboard* aplikasi pembuatan CV otomatis menampilkan ringkasan informasi pribadi pengguna, termasuk data pribadi, pendidikan, pengalaman kerja, dan keterampilan. Fitur ini memudahkan pengguna mengedit dan mengelola data untuk pembuatan CV yang akurat dan efektif.



Gambar 4. 34 Halaman *Biodata Dashboard*

## 9. Halaman Tambah Biodata

Halaman Tambah Biodata pada aplikasi pembuatan CV otomatis memungkinkan pengguna memasukkan data pribadi, pendidikan, pengalaman kerja, keterampilan dan informasi relevan lainnya untuk melengkapi profil dan memudahkan proses pembuatan CV yang akurat.

Gambar 4. 35 Halaman *Tambah Biodata*

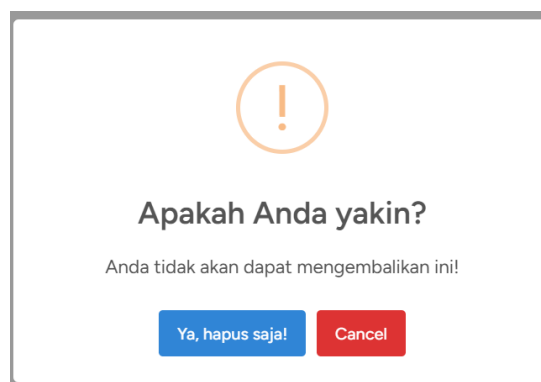
## 10. Halaman *Edit Biodata*

Halaman *Edit Biodata* aplikasi pembuatan CV otomatis memungkinkan pengguna memperbarui dan meng*Edit* informasi pribadi, pendidikan, pengalaman kerja, dan keterampilan untuk memastikan keakuratan dan kesegaran data dalam pembuatan CV.

Gambar 4. 36 Halaman *Edit Biodata*

## 11. Halaman Hapus Biodata

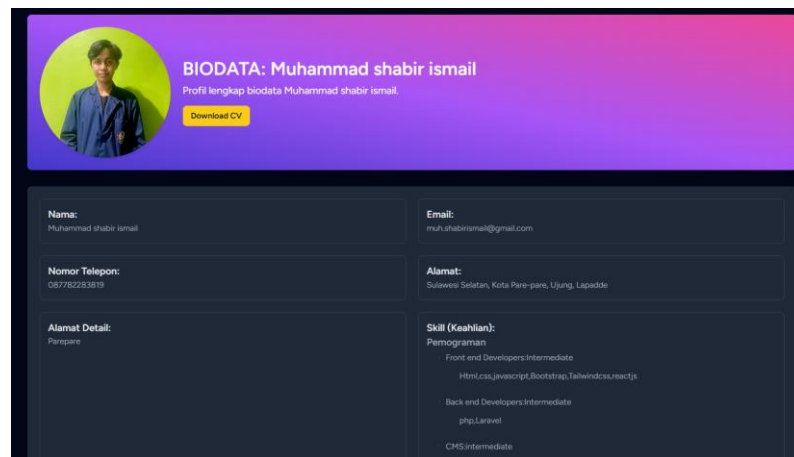
Halaman Hapus Biodata aplikasi pembuatan CV otomatis memungkinkan pengguna menghapus data pribadi, pendidikan, pengalaman kerja dan keterampilan. Fitur ini memudahkan pengelolaan profil dan memastikan keamanan data dengan menghapus informasi yang tidak diperlukan.

Gambar 4. 37 Halaman Hapus *Biodata*

## 12. Halaman Lihat Biodata Secara Detail

Halaman Lihat Biodata aplikasi pembuatan CV otomatis menampilkan

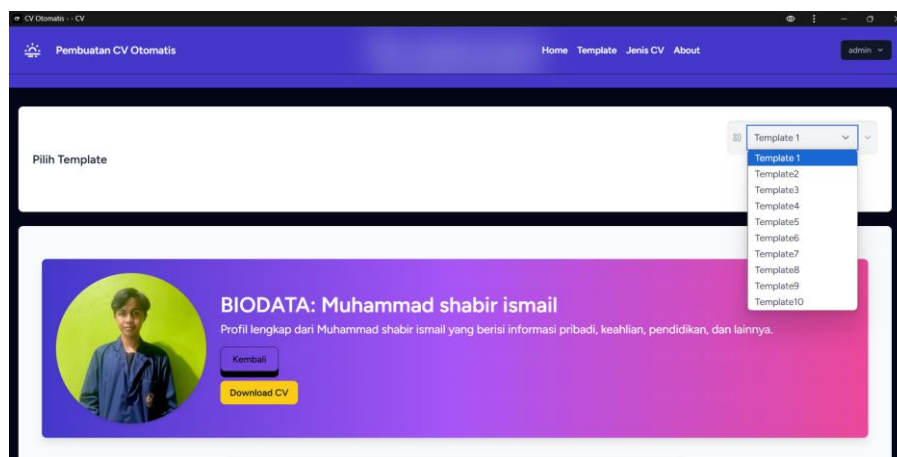
informasi pribadi, pendidikan, pengalaman kerja, keterampilan dan data relevan lainnya secara detail.



Gambar 4. 38 Halaman Lihat *detail Biodata*

### 13. Halaman *Export Biodata*

Halaman *Export Biodata* aplikasi pembuatan CV otomatis memungkinkan pengguna memilih template yang sesuai dan mengunduh data pribadi dalam format PDF. Fitur ini memudahkan pengguna menyimpan dan membagikan CV profesional dengan desain yang menarik.





Gambar 4. 39 Halaman Pilih template dan Export Biodata

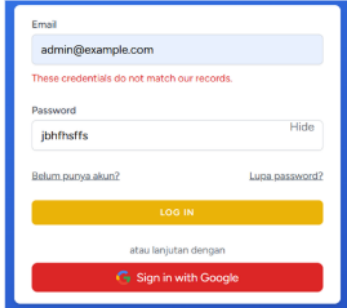
## E. Pengujian sistem

Pengujian sistem dilakukan untuk mendeteksi adanya kesalahan, kelemahan atau ketidaksesuaian dalam sistem sebelum digunakan. Terdapat dua metode utama pengujian sistem, yakni pengujian *Black box* dan *White Box*.

### 1. *Black box*

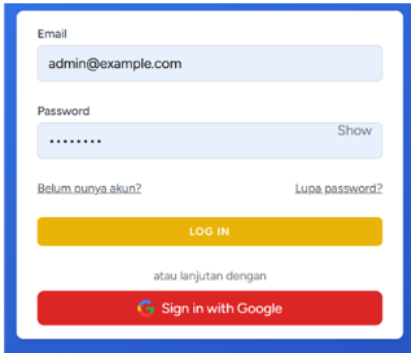
#### a. *Black box testing* kesalahan email dan password

Tabel 4. 3 *Blackbox testing* kesalahan email dan password

| Test faktor                                                                        | Hasil | Keterangan                                                                            |
|------------------------------------------------------------------------------------|-------|---------------------------------------------------------------------------------------|
| Memasukkan<br><i>email atau password</i><br>yang tidak sesuai                      | ✓     | Berhasil ketika <i>email atau password</i> tidak sesuai tampil<br><i>login failed</i> |
| <b>Screenshot</b>                                                                  |       |                                                                                       |
|  |       |                                                                                       |

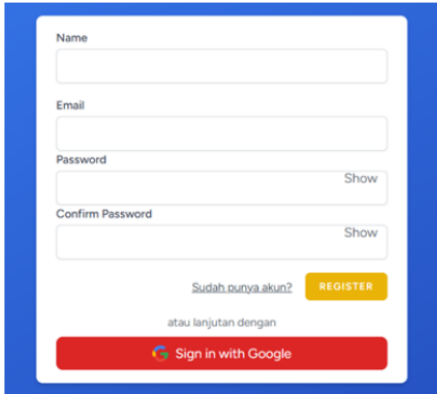
**b. Black box testing Login berhasil**

Tabel 4. 4 *Blackbox testing Login Berhasil*

| Test faktor                                                                          | Hasil | Keterangan                                                   |
|--------------------------------------------------------------------------------------|-------|--------------------------------------------------------------|
| Memasukkan<br><i>email atau password</i><br>yang sesuai                              | ✓     | Sistem berhasil menampilkan<br>halaman<br><i>Dashboard</i> . |
| <b>Screenshot</b>                                                                    |       |                                                              |
|  |       |                                                              |

c. **Black box testing Register**

Tabel 4. 5 *Blackbox testing Register*

| Test faktor                                                                         | Hasil | Keterangan                                             |
|-------------------------------------------------------------------------------------|-------|--------------------------------------------------------|
| Mengisi <i>Form input</i> yang sesuai                                               | ✓     | Sistem berhasil menampilkan halaman <i>Dashboard</i> . |
| <i>Screenshot</i>                                                                   |       |                                                        |
|  |       |                                                        |

d. **Black box testing lupa password**

Tabel 4. 6 *Blackbox testing lupa password*

| Test faktor                                                   | Hasil | Keterangan                                         |
|---------------------------------------------------------------|-------|----------------------------------------------------|
| Mengisi <i>Form input email dan password baru</i> yang sesuai | ✓     | Sistem berhasil menampilkan halaman <i>login</i> . |

**Screenshot**

**Periksa Email Anda**

Masukkan email Anda untuk menerima tautan reset password. Ini akan membantu kami memastikan akun Anda tetap aman.

Email

Masukkan alamat email Anda

KIRIM LINK RESET PASSWORD

Mengapa kami meminta email Anda?

Kami menggunakan alamat email Anda untuk mengirimkan tautan aman yang memungkinkan Anda mengganti kata sandi. Jangan khawatir, email Anda aman bersama kami.

[Kembali ke Halaman Login](#)

**Ganti Password**

Demi keamanan akun Anda, harap ganti password Anda secara berkala atau jika Anda merasa akun Anda tidak aman.

Password Baru

Masukkan password baru Show

Konfirmasi Password Baru

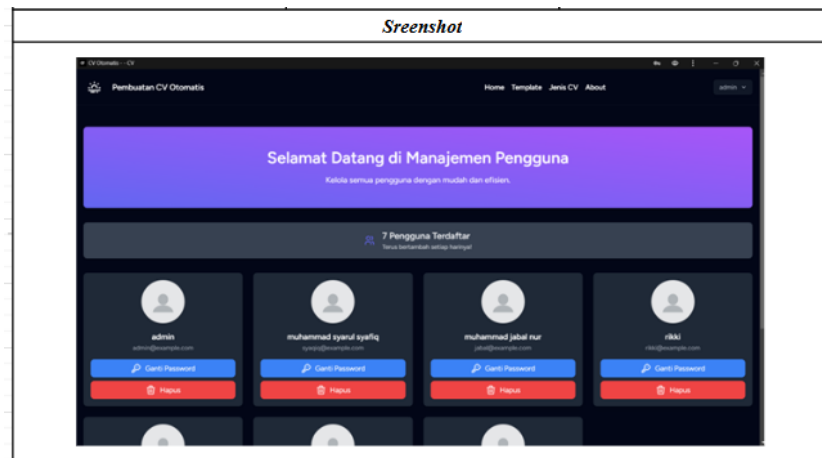
Ulangi password baru Show

SIMPAN PASSWORD

**e. Black box testing Admin Dashboard**

Tabel 4. 7 Blackbox testing Admin Dashboard

| Test faktor                             | Hasil | Keterangan                                           |
|-----------------------------------------|-------|------------------------------------------------------|
| Admin mengakses Halaman Admin Dashboard | ✓     | Sistem berhasil menampilkan halaman Admin Dashboard. |



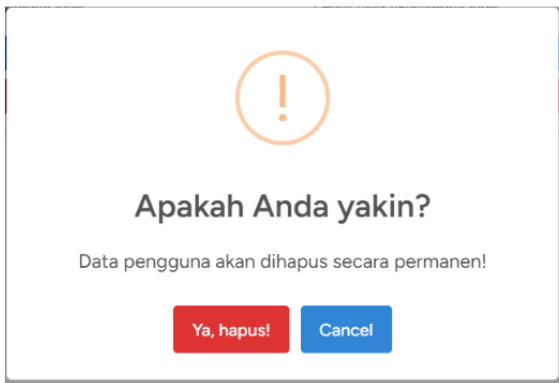
**f. Black box testing Admin ganti password User**

Tabel 4. 8 Blackbox testing Admin ganti password User

| Test faktor                                                                                                                                                                                                                                               | Hasil | Keterangan                                                            |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|-----------------------------------------------------------------------|
| Mengisi <i>Form input</i> yang sesuai.                                                                                                                                                                                                                    | ✓     | Sistem berhasil menampilkan <i>notifikasi password telah di ubah.</i> |
| <i>Screenshot</i>                                                                                                                                                                                                                                         |       |                                                                       |
| <p>The screenshot shows a modal form titled 'Ganti Password untuk user1'. It contains two input fields: 'Password Baru' and 'Konfirmasi Password'. At the bottom right, there are two buttons: 'Simpan' (Save) in green and 'Batal' (Cancel) in grey.</p> |       |                                                                       |

**g. Black box testing Admin Hapus User**

Tabel 4. 9 Blackbox testing Admin hapus User

| Test faktor                                                                         | Hasil | Keterangan                                                        |
|-------------------------------------------------------------------------------------|-------|-------------------------------------------------------------------|
| Mengisi <i>Form input</i> yang sesuai                                               | ✓     | Sistem berhasil menampilkan <i>notifikasi user telah di hapus</i> |
| <b>Screenshot</b>                                                                   |       |                                                                   |
|  |       |                                                                   |

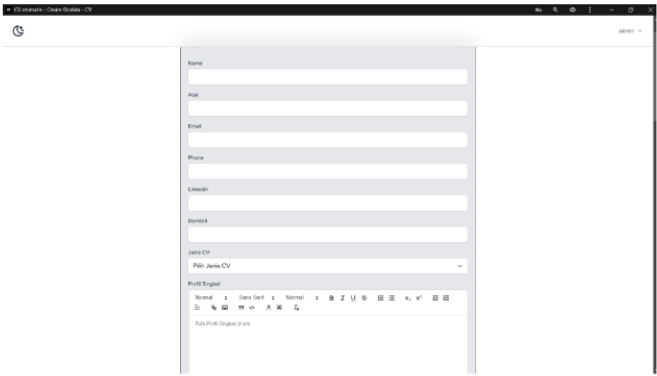
**h. Black box testing Biodata Dashboard**

Tabel 4. 10 Blackbox testing Biodata Dashboard

| Test faktor                                                                          | Hasil | Keterangan                                                     |
|--------------------------------------------------------------------------------------|-------|----------------------------------------------------------------|
| <i>user</i><br>Mengakses halaman <i>dashboard biodata</i> .                          | ✓     | Sistem berhasil menampilkan halaman <i>Dashboard biodata</i> . |
| <b>Screenshot</b>                                                                    |       |                                                                |
|  |       |                                                                |

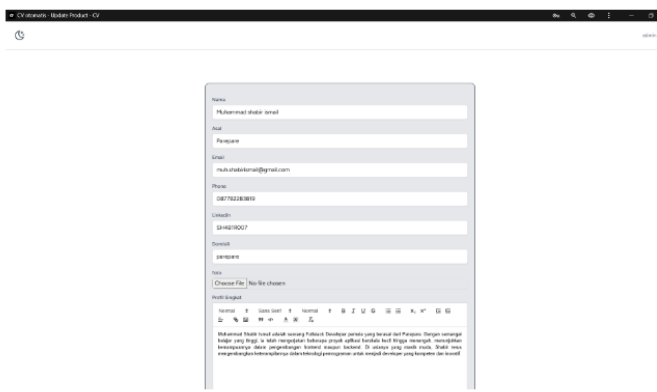
i. **Black box testing** tambah Biodata

Tabel 4. 11 *Blackbox testing* tambah biodata

| Test faktor                                                                         | Hasil | Keterangan                                |
|-------------------------------------------------------------------------------------|-------|-------------------------------------------|
| Mengisi <i>form input</i> yang sesuai                                               | ✓     | Sistem berhasil menambah <i>Biodata</i> . |
| <b>Screenshot</b>                                                                   |       |                                           |
|  |       |                                           |

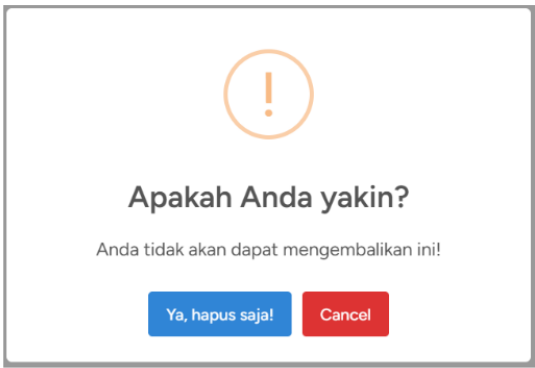
j. **Black box testing** Edit Biodata

Tabel 4. 12 *Blackbox testing* Edit biodata

| Test faktor                                                                          | Hasil | Keterangan                                |
|--------------------------------------------------------------------------------------|-------|-------------------------------------------|
| Mengisi <i>form input</i> yang sesuai                                                | ✓     | Sistem berhasil mengubah <i>Biodata</i> . |
| <b>Screenshot</b>                                                                    |       |                                           |
|  |       |                                           |

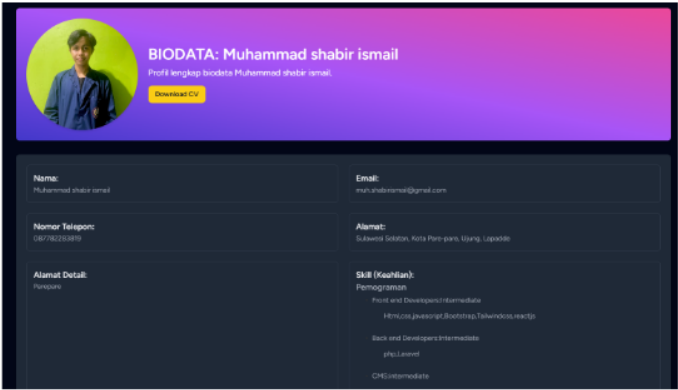
### k. *Black box testing* Hapus Biodata

Tabel 4. 13 *Blackbox testing* hapus biodata

| Test faktor                                                                         | Hasil | Keterangan                         |
|-------------------------------------------------------------------------------------|-------|------------------------------------|
| User<br>Menghapus biodata.                                                          | ✓     | Sistem berhasil menghapus biodata. |
| Screenshot                                                                          |       |                                    |
|  |       |                                    |

### l. *Black box testing* Lihat detail Biodata

Tabel 4. 14 *Blackbox testing* lihat detail biodata

| Test faktor                                                                          | Hasil | Keterangan                                          |
|--------------------------------------------------------------------------------------|-------|-----------------------------------------------------|
| User<br>Mengakses halaman lihat detail biodata .                                     | ✓     | Sistem berhasil menampilkan halaman detail Biodata. |
| Screenshot                                                                           |       |                                                     |
|  |       |                                                     |

**m. Black box testing Biodata Export**

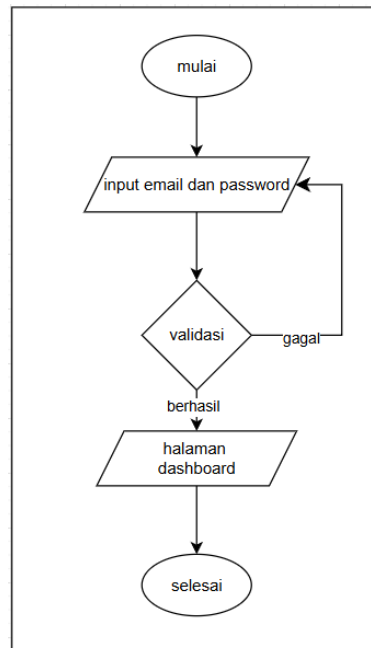
Tabel 4. 15 Blackbox testing Export biodata

| Test faktor                                                                                                                       | Hasil | Keterangan                                  |
|-----------------------------------------------------------------------------------------------------------------------------------|-------|---------------------------------------------|
| <p><i>User</i><br/>Mengakses halaman <i>export biodata</i> dan memilih <i>template</i> kemudian <i>mengexport biodatanya</i>.</p> | ✓     | Sistem berhasil <i>m mengexport Biodata</i> |
| Screenshot                                                                                                                        |       |                                             |
|                                                |       |                                             |

## 2. *White box*

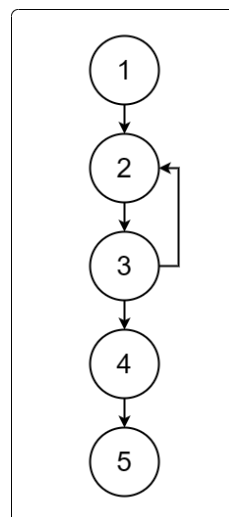
### 1. *White box testing kesalahan email dan password*

#### 1) *Flowchart*



Gambar 4. 40 *Flowchart kesalahan email dan password*

#### 2) *Flowgraph*



Gambar 4. 41 *Flowgraph Kesalahan email dan password*

Berdasarkan gambar 4.41 diatas, dilakukan perhitungan seperti di

bawah ini:

(1) Analisis dilakukan menggunakan rumus kompleksitas siklomatik

$$V(G) = E - N + 2P \quad V(G) = E - N + 2P, \text{ di mana:}$$

$$E \text{ (edge)} = 5$$

$$N \text{ (node)} = 5$$

$$P \text{ (Predikat Node)} = 1$$

Perhitungan:

$$V(G) = E - N + 2$$

$$= 5 - 5 + 2 = 2$$

$$\text{Predikat (P)} = P + 1$$

$$= 1 + 1 = 2$$

(2) Perhitungan tersebut menunjukkan jumlah region sebanyak 2.

(3) Terdapat dua jalur independen, yaitu:

$$\text{Jalur 1} = 1 - 2 - 3 - 2$$

$$\text{Jalur 2} = 1 - 2 - 3 - 4 - 5$$

(4) Berikut Tabel matriks kesalahan *email* dan *password*

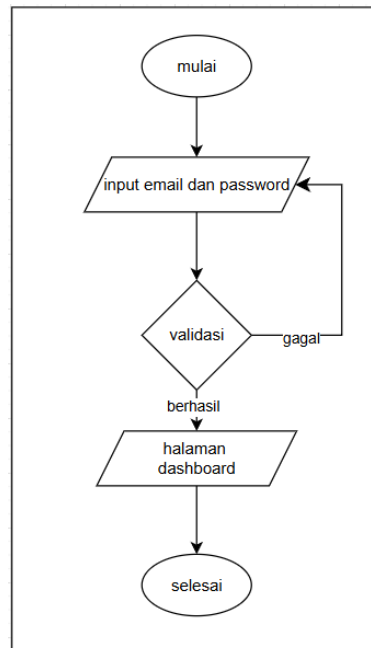
Tabel 4. 16 Tabel *matriks* kesalahan *email* dan *password*

| Node | 1 | 2 | 3 | 4 | 5 | E-1         |
|------|---|---|---|---|---|-------------|
| 1    |   | 1 |   |   |   | $1 - 1 = 0$ |
| 2    |   |   | 1 |   |   | $1 - 1 = 0$ |
| 3    |   | 1 |   | 1 |   | $2 - 1 = 1$ |
| 4    |   |   |   |   | 1 | $1 - 1 = 0$ |
| 5    |   |   |   |   |   | 0           |

|       |             |           |
|-------|-------------|-----------|
| Total | SUM (E + 1) | 1 + 1 = 2 |
|-------|-------------|-----------|

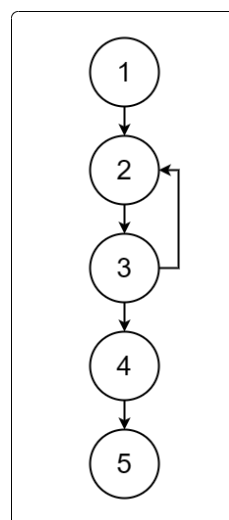
## 2. *White box testing Login berhasil*

### 1) *Flowchart*



Gambar 4. 42 *Flowchart Login berhasil*

### 2) *Flowgraph*



Gambar 4. 43 *Flowgraph Login berhasil*

Berdasarkan gambar 4.43 diatas, dilakukan perhitungan seperti di bawah ini:

(1) Analisis dilakukan menggunakan rumus kompleksitas siklomatik

$$V(G)=E-N+2V(G) = E - N + 2V(G)=E-N+2, \text{ di mana:}$$

$$E \text{ (edge)} = 5$$

$$N \text{ (node)} = 5$$

$$P \text{ (Predikat Node)} = 1$$

Perhitungan:

$$V(G) = E - N + 2$$

$$= 5 - 5 + 2 = 2$$

$$\text{Predikat (P)} = P + 1$$

$$= 1 + 1 = 2$$

(2) Perhitungan tersebut menunjukkan jumlah region sebanyak 2.

(3) Terdapat dua jalur independen, yaitu:

$$\text{Jalur 1} = 1 - 2 - 3 - 2$$

$$\text{Jalur 2} = 1 - 2 - 3 - 4 - 5$$

(4) Berikut Tabel matriks *Login* berhasil

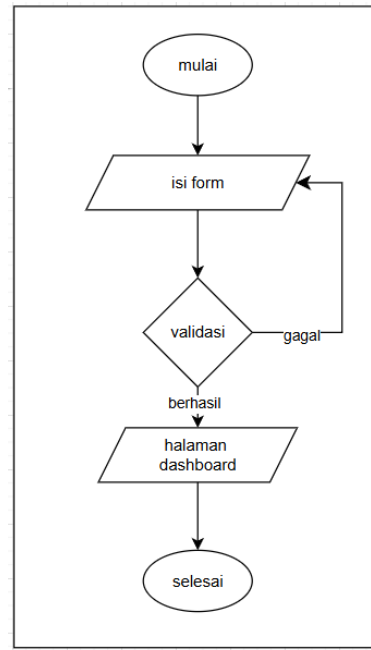
Tabel 4. 17 *Tabel matriks Login* berhasil

| Node | 1 | 2 | 3 | 4 | 5 | E-1         |
|------|---|---|---|---|---|-------------|
| 1    |   | 1 |   |   |   | $1 - 1 = 0$ |
| 2    |   |   | 1 |   |   | $1 - 1 = 0$ |
| 3    |   | 1 |   | 1 |   | $2 - 1 = 1$ |
| 4    |   |   |   |   | 1 | $1 - 1 = 0$ |

|       |             |  |  |  |  |             |
|-------|-------------|--|--|--|--|-------------|
| 5     |             |  |  |  |  | 0           |
| Total | SUM (E + 1) |  |  |  |  | $1 + 1 = 2$ |

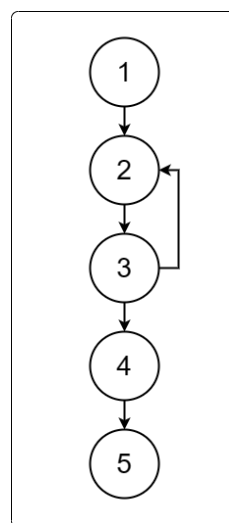
### 3. White box testing Register

#### 1) Flowchart



Gambar 4. 44 Flowchart Register

#### 2) Flowgraph



Gambar 4. 45 Flowgraph Register

Berdasarkan gambar 4.45 diatas, dilakukan perhitungan seperti di bawah ini:

(1) Analisis dilakukan menggunakan rumus kompleksitas siklomatik

$$V(G) = E - N + 2P \quad V(G) = E - N + 2, \text{ di mana:}$$

$$E (\text{edge}) = 5$$

$$N (\text{node}) = 5$$

$$P (\text{Predikat Node}) = 1$$

Perhitungan:

$$V(G) = E - N + 2$$

$$= 5 - 5 + 2 = 2$$

$$\text{Predikat (P)} = P + 1$$

$$= 1 + 1 = 2$$

(2) Perhitungan tersebut menunjukkan jumlah region sebanyak 2.

(3) Terdapat dua jalur independen, yaitu:

$$\text{Jalur 1} = 1 - 2 - 3 - 2$$

$$\text{Jalur 2} = 1 - 2 - 3 - 4 - 5$$

(4) Berikut Tabel matriks *Register*

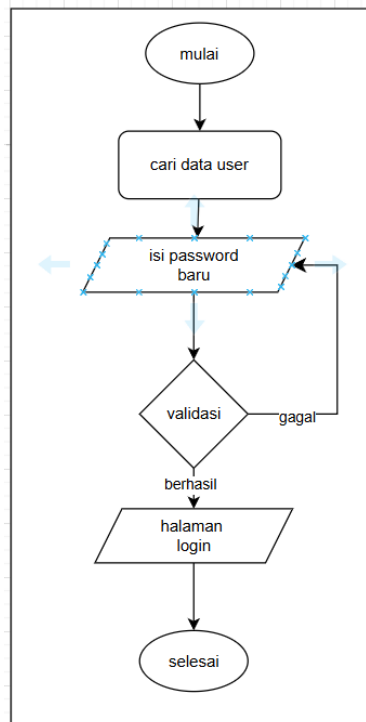
Tabel 4. 18 Tabel *matriks Register*

| Node | 1 | 2 | 3 | 4 | 5 | E-1         |
|------|---|---|---|---|---|-------------|
| 1    |   | 1 |   |   |   | $1 - 1 = 0$ |
| 2    |   |   | 1 |   |   | $1 - 1 = 0$ |
| 3    |   | 1 |   | 1 |   | $2 - 1 = 1$ |
| 4    |   |   |   |   | 1 | $1 - 1 = 0$ |

|       |             |  |  |  |  |           |
|-------|-------------|--|--|--|--|-----------|
| 5     |             |  |  |  |  | 0         |
| Total | SUM (E + 1) |  |  |  |  | 1 + 1 = 2 |

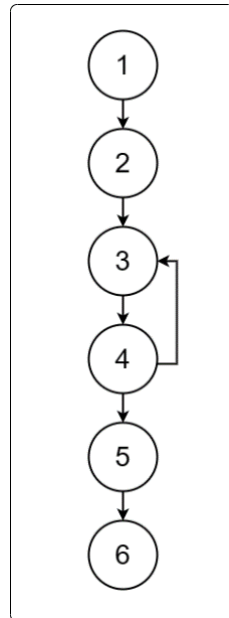
#### 4. *White box testing lupa password*

##### 1) *Flowchart*



Gambar 4. 46 *Flowchart lupa password*

## 2) Flowgraph



Gambar 4. 47 Flowgraph lupa password

Berdasarkan gambar 4.47 diatas, dilakukan perhitungan seperti di bawah ini:

(1) Analisis dilakukan menggunakan rumus kompleksitas siklomatik

$$V(G) = E - N + 2 \quad V(G) = E - N + 2 \quad V(G) = E - N + 2, \text{ di mana:}$$

$$E \text{ (edge)} = 6$$

$$N \text{ (node)} = 6$$

$$P \text{ (Predikat Node)} = 1$$

Perhitungan:

$$V(G) = E - N + 2$$

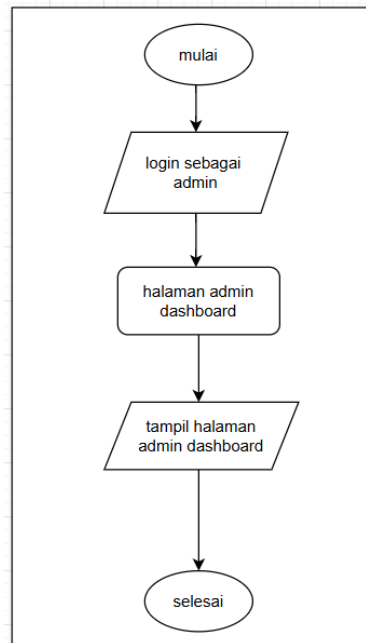
$$= 6 - 6 + 2 = 2$$

$$\text{Predikat (P)} = P + 1$$



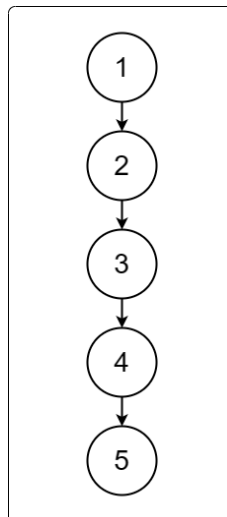
## 5. *White box testing Admin Dashboard*

### 1) *Flowchart*



Gambar 4. 48 *Flowchart Admin Dashboard*

### 2) *Flowgraph*



Gambar 4. 49 *Flowgraph Admin Dashboard*

Berdasarkan gambar 4.49 diatas, dilakukan perhitungan seperti di bawah ini:

(1) Analisis dilakukan menggunakan rumus kompleksitas siklomatik

$$V(G) = E - N + 2P \quad V(G) = E - N + 2P, \text{ di mana:}$$

$$E (\text{edge}) = 5$$

$$N (\text{node}) = 5$$

$$P (\text{Predikat Node}) = 1$$

Perhitungan:

$$V(G) = E - N + 2$$

$$= 5 - 5 + 2 = 2$$

$$\text{Predikat (P)} = P + 1$$

$$= 1 + 1 = 2$$

(2) Perhitungan tersebut menunjukkan jumlah region sebanyak 2.

(3) Terdapat dua jalur independen, yaitu:

$$\text{Jalur 1} = 1 - 2 - 3 - 2$$

$$\text{Jalur 2} = 1 - 2 - 3 - 4 - 5$$

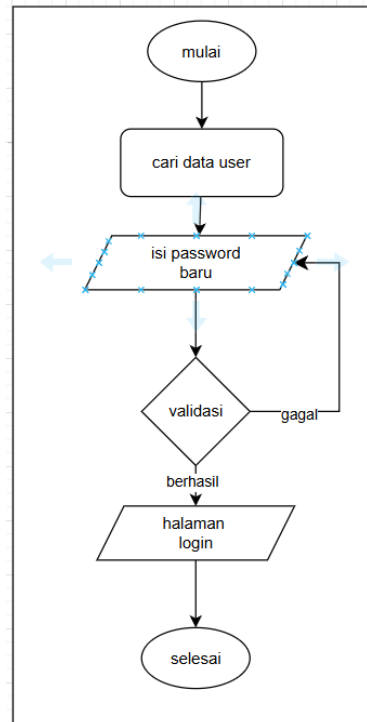
(4) Berikut Tabel matriks *Admin Dashboard*

Tabel 4. 20 Tabel *matriks Admin Dashboard*

| Node  | 1           | 2 | 3 | 4 | 5 | E-1         |
|-------|-------------|---|---|---|---|-------------|
| 1     |             | 1 |   |   |   | $1 - 1 = 0$ |
| 2     |             |   | 1 |   |   | $1 - 1 = 0$ |
| 3     |             | 1 |   | 1 |   | $2 - 1 = 1$ |
| 4     |             |   |   |   | 1 | $1 - 1 = 0$ |
| 5     |             |   |   |   |   | 0           |
| Total | SUM (E + 1) |   |   |   |   | $1 + 1 = 2$ |

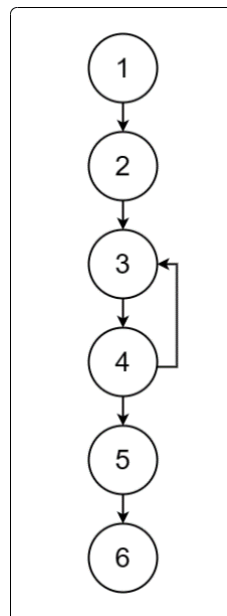
## 6. White box testing Admin Ganti password User

### 1) Flowchart



Gambar 4. 50 Flowchart Admin ganti password

### 2) Flowgraph



Gambar 4. 51 Flowgraph Admin ganti password

Berdasarkan gambar 4.51 diatas, dilakukan perhitungan seperti di bawah ini:

(1) Analisis dilakukan menggunakan rumus kompleksitas siklomatik

$$V(G)=E-N+2V(G) = E - N + 2V(G)=E-N+2, \text{ di mana:}$$

$$E \text{ (edge)} = 6$$

$$N \text{ (node)} = 6$$

$$P \text{ (Predikat Node)} = 1$$

Perhitungan:

$$V(G) = E - N + 2$$

$$= 6 - 6 + 2 = 2$$

$$\text{Predikat (P)} = P + 1$$

$$= 1 + 1 = 2$$

(2) Perhitungan tersebut menunjukkan jumlah region sebanyak 2.

(3) Terdapat dua jalur independen, yaitu:

$$\text{Jalur 1} = 1 - 2 - 3 - 4 - 3$$

$$\text{Jalur 2} = 1 - 2 - 3 - 4 - 5 - 6$$

(4) Berikut Tabel matriks *Admin ganti password*

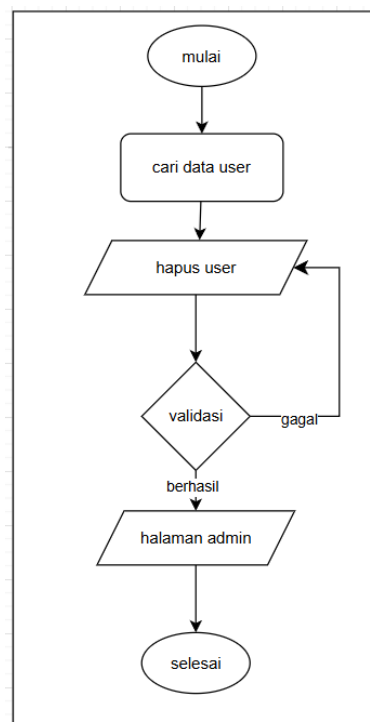
Tabel 4. 21 Tabel *matriks Admin password*

| Node | 1 | 2 | 3 | 4 | 5 | 6 | E-1         |
|------|---|---|---|---|---|---|-------------|
| 1    |   | 1 |   |   |   |   | $1 - 1 = 0$ |
| 2    |   |   | 1 |   |   |   | $1 - 1 = 0$ |
| 3    |   |   |   | 1 |   |   | $1 - 1 = 0$ |
| 4    |   |   | 1 |   | 1 |   | $2 - 1 = 1$ |

|       |             |  |  |  |  |   |             |
|-------|-------------|--|--|--|--|---|-------------|
| 5     |             |  |  |  |  | 1 | $1 - 1 = 0$ |
| 6     |             |  |  |  |  |   | 0           |
| Total | SUM (E + 1) |  |  |  |  |   | $1 + 1 = 2$ |

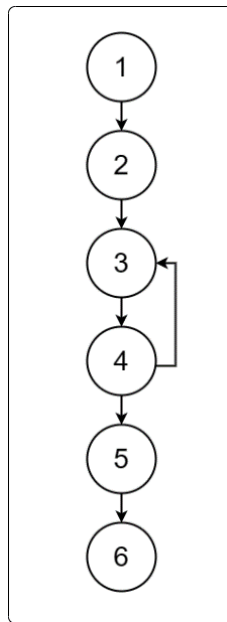
## 7. *White box testing Admin hapus User*

### 1) *Flowchart*



Gambar 4. 52 *Flowchart Admin hapus User*

## 2) Flowgraph



Gambar 4. 53 Flowgraph Admin hapus User\

Berdasarkan gambar 4.53 diatas, dilakukan perhitungan seperti di bawah ini:

(1) Analisis dilakukan menggunakan rumus kompleksitas siklomatik

$$V(G) = E - N + 2 \quad V(G) = E - N + 2, \text{ di mana:}$$

$$E \text{ (edge)} = 6$$

$$N \text{ (node)} = 6$$

$$P \text{ (Predikat Node)} = 1$$

Perhitungan:

$$V(G) = E - N + 2$$

$$= 6 - 6 + 2 = 2$$

$$\text{Predikat (P)} = P + 1$$

$$= 1 + 1 = 2$$

(2) Perhitungan tersebut menunjukkan jumlah region sebanyak 2.

(3) Terdapat dua jalur independen, yaitu:

Jalur 1 = 1 – 2 – 3 – 4 – 3

Jalur 2 = 1 – 2 – 3 – 4 – 5 – 6

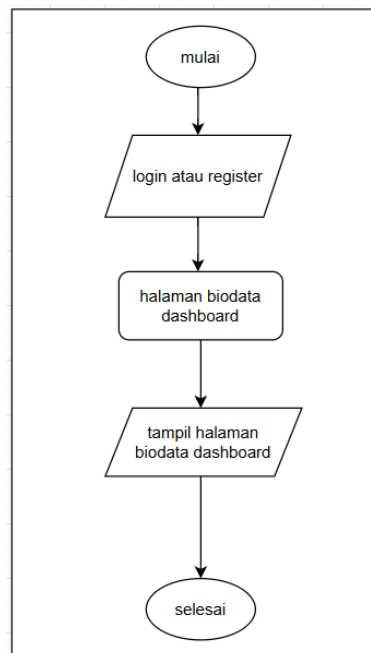
(4) Berikut Tabel matriks hapus *User*

Tabel 4. 22 Tabel *matriks Admin hapus User*

[illegible]

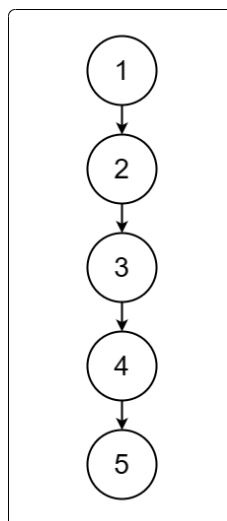
## 8. White box testing Biodata Dashboard

### 1) Flowchart



Gambar 4. 54 Flowchart Biodata Dashboard

### 2) Flowgraph



Gambar 4. 55 Flowgraph biodata Dashboard

Berdasarkan gambar 4.55 diatas, dilakukan perhitungan seperti di bawah ini:

(1) Analisis dilakukan menggunakan rumus kompleksitas siklomatik

$$V(G)=E-N+2 \quad V(G) = E - N + 2 \quad V(G)=E-N+2, \text{ di mana:}$$

$$E (\text{edge}) = 5$$

$$N (\text{node}) = 5$$

$$P (\text{Predikat Node}) = 1$$

Perhitungan:

$$V(G) = E - N + 2$$

$$= 5 - 5 + 2 = 2$$

$$\text{Predikat (P)} = P + 1$$

$$= 1 + 1 = 2$$

(2) Perhitungan tersebut menunjukkan jumlah region sebanyak 2.

(3) Terdapat dua jalur independen, yaitu:

$$\text{Jalur 1} = 1 - 2 - 3 - 2$$

$$\text{Jalur 2} = 1 - 2 - 3 - 4 - 5$$

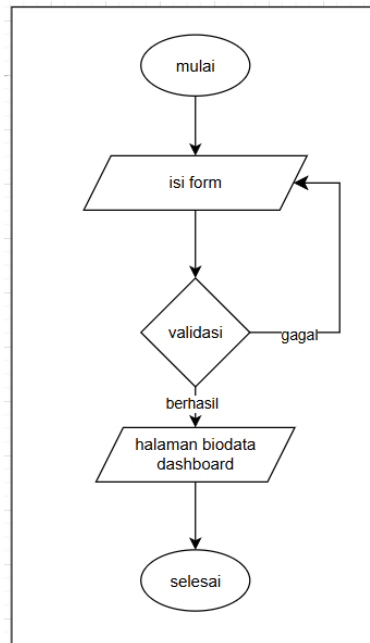
(4) Berikut Tabel matriks biodata *Dashboard*

Tabel 4. 23 Tabel *matriks biodata Dashboard*

| Node  | 1           | 2 | 3 | 4 | 5 | E-1         |
|-------|-------------|---|---|---|---|-------------|
| 1     |             | 1 |   |   |   | $1 - 1 = 0$ |
| 2     |             |   | 1 |   |   | $1 - 1 = 0$ |
| 3     |             | 1 |   | 1 |   | $2 - 1 = 1$ |
| 4     |             |   |   |   | 1 | $1 - 1 = 0$ |
| 5     |             |   |   |   |   | 0           |
| Total | SUM (E + 1) |   |   |   |   | $1 + 1 = 2$ |

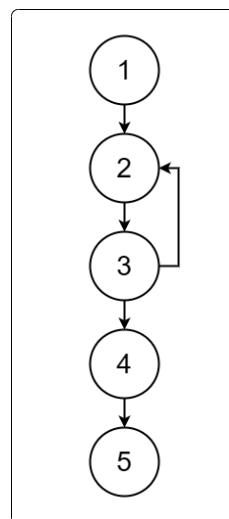
## 9. *White box testing* tambah biodata

### 1) *Flowchart*



Gambar 4. 56 *Flowchart* tambah biodata

### 2) *Flowgraph*



Gambar 4. 57 *Flowgraph* tambah biodata

Berdasarkan gambar 4.57 diatas, dilakukan perhitungan seperti di bawah ini:

(1) Analisis dilakukan menggunakan rumus kompleksitas siklomatik

$$V(G)=E-N+2 \quad V(G)=E-N+2, \text{ di mana:}$$

$$E (\text{edge}) = 5$$

$$N (\text{node}) = 5$$

$$P (\text{Predikat Node}) = 1$$

Perhitungan:

$$V(G) = E - N + 2$$

$$= 5 - 5 + 2 = 2$$

$$\text{Predikat (P)} = P + 1$$

$$= 1 + 1 = 2$$

(2) Perhitungan tersebut menunjukkan jumlah region sebanyak 2.

(3) Terdapat dua jalur independen, yaitu:

$$\text{Jalur 1} = 1 - 2 - 3 - 2$$

$$\text{Jalur 2} = 1 - 2 - 3 - 4 - 5$$

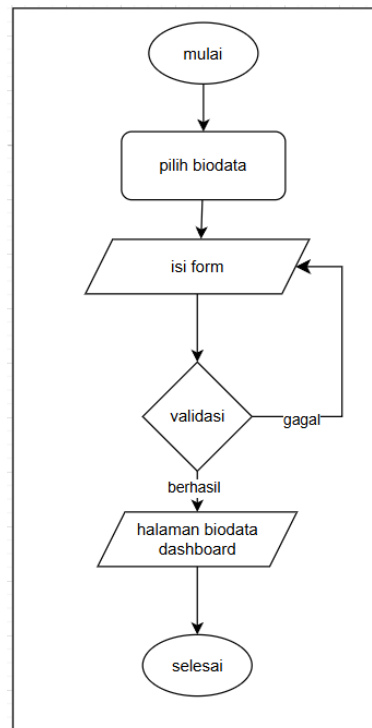
(4) Berikut Tabel matriks tambah biodata

Tabel 4. 24 *Tabel matriks tambah biodata*

| Node  | 1           | 2 | 3 | 4 | 5 | E-1         |
|-------|-------------|---|---|---|---|-------------|
| 1     |             | 1 |   |   |   | $1 - 1 = 0$ |
| 2     |             |   | 1 |   |   | $1 - 1 = 0$ |
| 3     |             | 1 |   | 1 |   | $2 - 1 = 1$ |
| 4     |             |   |   |   | 1 | $1 - 1 = 0$ |
| 5     |             |   |   |   |   | 0           |
| Total | SUM (E + 1) |   |   |   |   | $1 + 1 = 2$ |

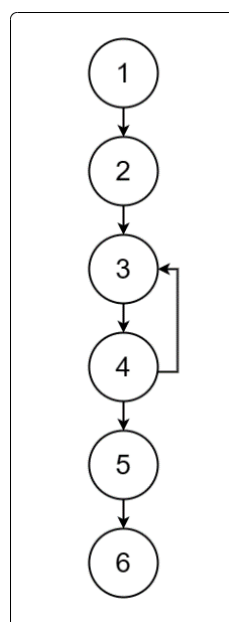
## 10. White box testing *Edit biodata*

### 1) *Flowchart*



Gambar 4. 58 *Flowchart Edit biodata*

### 2) *Flowgraph*



Gambar 4. 59 *Flowgraph Edit biodata*

Berdasarkan gambar 4.59 diatas, dilakukan perhitungan seperti di bawah ini:

(1) Analisis dilakukan menggunakan rumus kompleksitas siklomatik

$$V(G)=E-N+2V(G) = E - N + 2V(G)=E-N+2, \text{ di mana:}$$

$$E \text{ (edge)} = 6$$

$$N \text{ (node)} = 6$$

$$P \text{ (Predikat Node)} = 1$$

Perhitungan:

$$V(G) = E - N + 2$$

$$= 6 - 6 + 2 = 2$$

$$\text{Predikat (P)} = P + 1$$

$$= 1 + 1 = 2$$

(2) Perhitungan tersebut menunjukkan jumlah region sebanyak 2.

(3) Terdapat dua jalur independen, yaitu:

$$\text{Jalur 1} = 1 - 2 - 3 - 4 - 3$$

$$\text{Jalur 2} = 1 - 2 - 3 - 4 - 5 - 6$$

(4) Berikut Tabel matriks *Edit* biodata

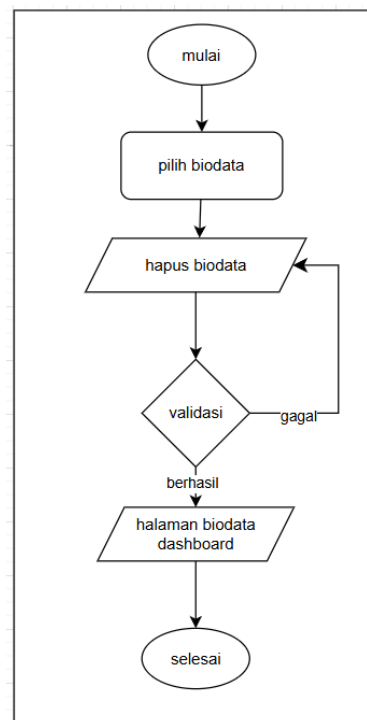
Tabel 4. 25 Tabel *matriks Edit biodata*

| Node | 1 | 2 | 3 | 4 | 5 | 6 | E-1         |
|------|---|---|---|---|---|---|-------------|
| 1    |   | 1 |   |   |   |   | $1 - 1 = 0$ |
| 2    |   |   | 1 |   |   |   | $1 - 1 = 0$ |
| 3    |   |   |   | 1 |   |   | $1 - 1 = 0$ |
| 4    |   |   | 1 |   | 1 |   | $2 - 1 = 1$ |

|       |             |  |  |  |  |   |             |
|-------|-------------|--|--|--|--|---|-------------|
| 5     |             |  |  |  |  | 1 | $1 - 1 = 0$ |
| 6     |             |  |  |  |  |   | 0           |
| Total | SUM (E + 1) |  |  |  |  |   | $1 + 1 = 2$ |

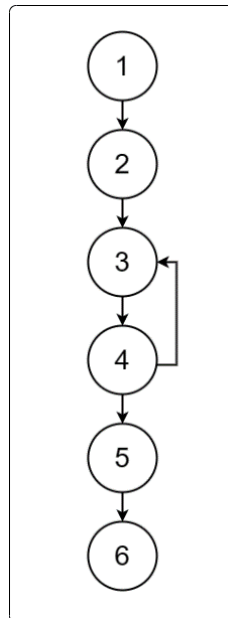
## 11. White box testing hapus biodata

### 1) Flowchart



Gambar 4. 60 Flowchart hapus biodata

## 2) Flowgraph



Gambar 4. 61 Flowgraph hapus biodata

Berdasarkan gambar 4.61 diatas, dilakukan perhitungan seperti di bawah ini:

(1) Analisis dilakukan menggunakan rumus kompleksitas siklomatik

$$V(G) = E - N + 2 \quad V(G) = E - N + 2 \quad V(G) = E - N + 2, \text{ di mana:}$$

$$E \text{ (edge)} = 6$$

$$N \text{ (node)} = 6$$

$$P \text{ (Predikat Node)} = 1$$

Perhitungan:

$$V(G) = E - N + 2$$

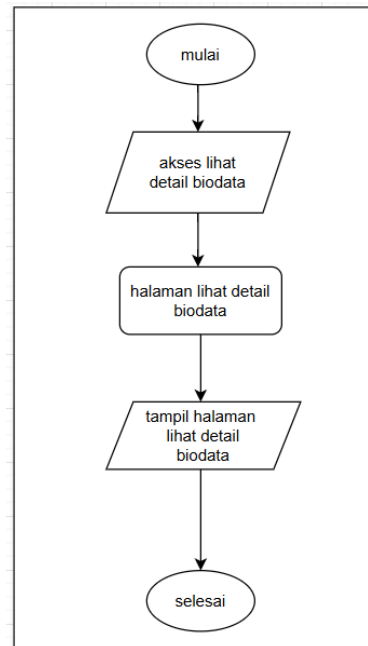
$$= 6 - 6 + 2$$

$$= 2$$



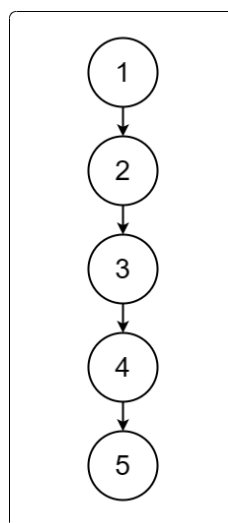
## 12. *White box testing* lihat detail Biodata

### 1) *Flowchart*



Gambar 4. 62 *Flowchart* lihat detail Biodata

### 2) *Flowgraph*



Gambar 4. 63 *Flowgraph* lihat detail biodata

Berdasarkan gambar 4.63 diatas, dilakukan perhitungan seperti di bawah ini:

(1) Analisis dilakukan menggunakan rumus kompleksitas siklomatik

$$V(G)=E-N+2 \quad V(G) = E - N + 2 \quad V(G)=E-N+2, \text{ di mana:}$$

$$E \text{ (edge)} = 5$$

$$N \text{ (node)} = 5$$

$$P \text{ (Predikat Node)} = 1$$

Perhitungan:

$$V(G) = E - N + 2$$

$$= 5 - 5 + 2 = 2$$

$$\text{Predikat (P)} = P + 1$$

$$= 1 + 1 = 2$$

(2) Perhitungan tersebut menunjukkan jumlah region sebanyak 2.

(3) Terdapat dua jalur independen, yaitu:

$$\text{Jalur 1} = 1 - 2 - 3 - 2$$

$$\text{Jalur 2} = 1 - 2 - 3 - 4 - 5$$

(4) Berikut Tabel matriks lihat detail biodata

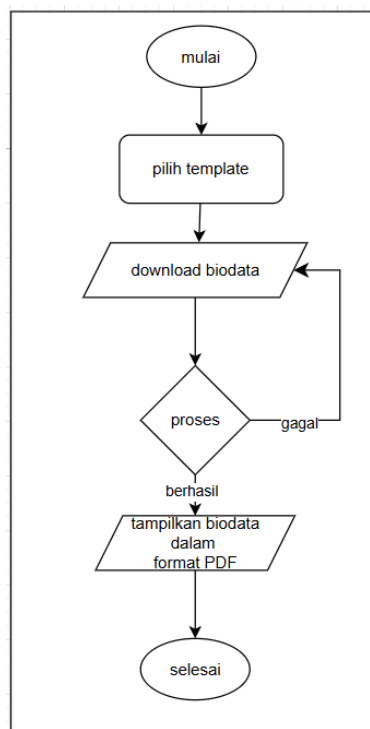
Tabel 4. 27 Tabel *matriks lihat detail biodata*

| Node | 1 | 2 | 3 | 4 | 5 | E-1         |
|------|---|---|---|---|---|-------------|
| 1    |   | 1 |   |   |   | $1 - 1 = 0$ |
| 2    |   |   | 1 |   |   | $1 - 1 = 0$ |
| 3    |   | 1 |   | 1 |   | $2 - 1 = 1$ |

|       |             |  |  |  |   |             |
|-------|-------------|--|--|--|---|-------------|
| 4     |             |  |  |  | 1 | $1 - 1 = 0$ |
| 5     |             |  |  |  |   | 0           |
| Total | SUM (E + 1) |  |  |  |   | $1 + 1 = 2$ |

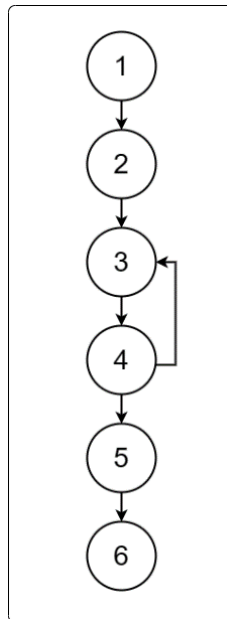
### 13. White box testing Export biodata

#### 1) Flowchart



Gambar 4. 64 Flowchart Export biodata

#### 2) Flowgraph



Gambar 4. 65 *Flowgraph Export biodata*

Berdasarkan gambar 4.65 diatas, dilakukan perhitungan seperti di bawah ini:

(1) Analisis dilakukan menggunakan rumus kompleksitas siklomatik

$$V(G) = E - N + 2 \quad V(G) = E - N + 2, \text{ di mana:}$$

$$E \text{ (edge)} = 6$$

$$N \text{ (node)} = 6$$

$$P \text{ (Predikat Node)} = 1$$

Perhitungan:

$$V(G) = E - N + 2$$

$$= 6 - 6 + 2 = 2$$

$$\text{Predikat (P)} = P + 1$$

$$= 1 + 1 = 2$$

(2) Perhitungan tersebut menunjukkan jumlah region sebanyak 2.

(3) Terdapat dua jalur independen, yaitu:



## BAB V

### PENUTUP

#### A. Kesimpulan

Berdasarkan penelitian yang telah dilakukan, kesimpulan yang dapat ditarik adalah sebagai berikut:

1. Dalam penelitian yang telah dilakukan, penulis berhasil mengembangkan sebuah sistem yang memudahkan pembuatan CV tanpa perlu memiliki kemampuan *Editor*.
2. Menggunakan teknologi *Progressive web apps* dalam pembuatan aplikasi *website* untuk memudahkan masyarakat dalam mengakses situs web tanpa perlu membuka *browser* atau mencari alamat situs, sehingga mempermudah untuk membuat CV dengan cepat tanpa perlu memiliki kemampuan *Editor*.
3. Sistem ini dibangun dengan menggunakan framework Laravel untuk *backend* dan React untuk *frontend* yang diintegrasikan melalui Inertia.js. Visual Studio Code digunakan sebagai *text Editor* dan MySQL sebagai sistem manajemen *database*.

## **B. Saran**

Pada penelitian ini, penulis mengakui adanya beberapa kekurangan yang memerlukan perbaikan dan pengembangan lebih lanjut. Untuk itu, penulis menawarkan beberapa saran untuk penelitian yang akan datang, yaitu:

1. Mengembangkan aplikasi agar pengguna dapat membuat template CV sendiri sesuai kebutuhan dan preferensi pribadi.
2. Mengintegrasikan aplikasi dengan platform pencari kerja untuk memudahkan pengguna mengunggah dan membagikan CV mereka secara langsung.
3. Menyediakan opsi untuk menerjemahkan resep ke dalam bahasa lain untuk memudahkan pengguna yang tidak memahami bahasa Indonesia.

## DAFTAR PUSTAKA

- Ahmad Saifullah. (2024). Sistem Pembuatan CV Berbasis *Progressive Web Apps* untuk Meningkatkan Aksesibilitas Pengguna. Seminar Nasional Teknologi Informasi.
- Alan Hume (2021). What is a Progressive Web App (PWA)? <https://www.amazon.com/Progressive-Apps-Dean-Alan-Hume/dp/1617294586>
- Angga Pratama. (2021). Pengembangan Aplikasi Pembuatan CV Otomatis Berbasis Web. *Jurnal Teknologi Informasi dan Aplikasi*.
- Ater, R. (2019). Progressive Web Apps: A New Era for Mobile Web Applications. *International Journal of Advanced Computer Science and Applications*, 10(3), 257-263.
- Awaluddin, M. I. (2020). *Implementasi Framework Laravel*. Universitas Bina Insani.
- Azhariyah, S., & Muhammad Mukhlis. (2024). "Framework CSS: Tailwind CSS Untuk Front-End Website Store PT. XYZ." *Jurnal Informatika*, 3(1), 30-36. <https://doi.org/10.57094/ji.v3i1.1601>
- Brown, A. (2023). The Rise of AI in Recruitment: Optimizing Your CV for ATS. *Journal of Career Development*, 45(3), 178-195.
- Castro, P. (2022, June 14). Introduction to CSS.
- Dian Septiani. (2022). Penggunaan *Progressive Web Apps* untuk Aplikasi Berbasis Web yang Dapat Diakses Secara Offline. *Jurnal Rekayasa Sistem dan Teknologi Informasi*.
- Duckett, J. (2022, March 8). What is CSS and how does it work? <https://css-tricks.com/learn-css/>
- Garcia, L. (2024). Personalization in Professional Branding: A Study on CV Customization. *International Journal of Human Resource Management*, 35(2), 312-330.
- Hartl, M., & Donahoe, K. (2019). *HTML and CSS: Design and Build Websites*. Pearson Education.
- Johnson, R. (2022). Digital CVs: Revolutionizing Professional Representation in the Online Era. *Tech in Careers*, 18(4), 205-220.

- Johnson, R. D., & Gueutal, H. G. (2021). Transforming HR Through Technology: The Use of E-HR and HRIS in Organizations. Society for Human Resource Management.
- Kurniawan, A. A., & Saputra, A. D. (2021). Rancang bangun Aplikasi E-Learning Berbasis *Progressive Web Apps* (PWA) untuk Meningkatkan Kualitas Pembelajaran di Masa Pandemi Covid-19. *Jurnal Teknologi Pendidikan*, 12(2), 223-232.
- Kusumawardhani, A., & Febriyanti, R. (2018). Pentingnya *Curriculum Vitae* (CV) dalam Melamar Pekerjaan. *Jurnal Manajemen dan Bisnis*, 14(2), 121-130.
- Meyer, E. A., & Weyl, S. (2020). *Cascading Style Sheets: The definitive guide* (4th ed.). O'Reilly Media.
- Morgan, E. (2021). MDN Web Docs: HTML <https://developer.mozilla.org/en-US/docs/Web/HTML>. React: Building Modern *User Interfaces*.
- Nematian, A. (2020). The Evolution of HTML: A Historical Perspective. *International Journal of Advanced Computer Science and Applications*, 11(5), 106-112.
- Patel, R., & Desai, M. (2020). Student's Perception on Resume Building Tools. *International Journal of Computer Applications*, 176(39), 18-22.
- Penerapan Konsep MVC Pada Aplikasi Web Menggunakan Framework Laravel. (2016). SISITI: Seminar Ilmiah Sistem Informasi dan Teknologi Informasi.
- Ridwan Hidayat. (2023). Rancang Bangun Aplikasi Pembuat CV Berbasis *Mobile* dengan Implementasi *Progressive Web Apps* (PWA). *Jurnal Teknologi dan Informatika*.
- Rifandi, F., Adriansyah, T. V., & Kurniawati, R. (2022). "Website Gallery Development Using Tailwind CSS Framework." *Jurnal E-Komtek (Elektro-Komputer-Teknik)*, 6(2), 205-214. [https://doi.org/10.37339/e-komtek.v6i2.937&#8203;;contentReference\[oaicite:1\]{index=1}](https://doi.org/10.37339/e-komtek.v6i2.937&#8203;;contentReference[oaicite:1]{index=1}).
- Setiaji, R. S. (2021). Implementasi *Diagram UML* Pada Perancangan Sistem Informasi Penggajian. *Jurnal Teknik Komputer AMIK BSI*, 7(1), 108-110.
- Setyawan, A. D., & Sari, R. F. (2019). Design and Implementation of E-Resume Application Based on *Progressive Web Apps* (PWA). *Jurnal Sains dan Teknologi Komputer*, 8(2), 115-122.
- Setyawan, A., & Sari, R. P. (2019). Rancang Bangun Aplikasi Pembuat CV Online Berbasis Web Responsif. *Jurnal Teknologi Informasi dan Komunikasi*, 10(1), 1-6.

- Siti Fadhillah. (2020). Pemanfaatan PWA untuk Peningkatan Pengalaman Pengguna pada Aplikasi Pembuatan CV Otomatis. *Jurnal Sistem Informasi*.
- Smith, J. (2021). The Evolution of Curriculum Vitae: From Paper to Digital Presence. *Career Development Quarterly*, 69(1), 45-60.
- Smith, J., & Brown, E. (2022). Modern Resume Strategies: Adapting to AI-driven Recruitment. *Journal of Career Development*, 45(3), 201-215.
- Supriadi, A., & Budiman, A. (2020). Building Progressive Web Apps (PWA) for *Mobile Learning: A Case Study*. In 2020 International Conference on Information Technology, Computer, and Electrical Engineering (ICITCEE) (pp. 254-257). IEEE.
- Sutarman, A. (2020). Pemrograman Web dengan PHP dan MySQL. Graha Ilmu.
- Tailwind Labs, *Tailwind CSS Documentation*, <https://tailwindcss.com/>, diakses pada 16 Juli 2024.
- Taylor, E. (2020). AI and the Future of Recruitment: Implications for Job Seekers. *HR Technology Review*, 12(2), 87-102.
- ulandari, E. R., & Sari, R. D. (2018). Implementasi *Framework* CodeIgniter Pada Rancang bangun Aplikasi E-Learning Berbasis Web. *Jurnal Sistem Informasi*, 15(1), 1-10. <https://ejournal.upi.edu/>
- W3Schools: HTML Tutorial <https://www.w3schools.com/html/>
- Wargo, J. (2020). Progressive Web Apps: Building Modern Web *Applications* with HTML, CSS, and Javascript. Apress.
- Wijianto, et al. (2018). PHPMyAdmin sebagai aplikasi web untuk *Administrasi* database MySQL. *Evolusi: Jurnal Sains dan Manajemen*, Vol. 7 No. 2, 2019. Diakses dari BSI E-Journa
- Wilson, M. (2024). Personal Branding in the Digital Age: Integrating Social Media into Your CV. *Journal of Professional Communication*, 28(1), 55-70.
- Yulianto Andika. (2023). Implementasi Fitur Personal Branding pada Aplikasi Pembuatan CV Berbasis PWA. *Jurnal Aplikasi dan Pengembangan Teknologi Informasi*.
- Yulianto, A., & Handayani, W. (2017). Rancang bangun Aplikasi Sistem Informasi Perpustakaan Berbasis Web Menggunakan *Framework Laravel*. *Jurnal Teknik Informatika*, 12(1), 1-10. <https://informatika.ums.ac.id/jurnal-khazanah-informatika/>

