

PENGUNAAN BINDING PADA PENGEMBANGAN WEBSITE PENERIMAAN MAHASISWA BARU DENGAN FRAMEWORK ANGULAR

Rashid Adani Maulana Jatri^{1*}, Zainudin Zukhri²

^{1,2}Informatika, Fakultas Teknologi Industri, Universitas Islam Indonesia, Sleman, Indonesia

Email: ^{1*}rashid.jatri@students.uui.ac.id, ²zainudin@uui.ac.id

(* : corresponding author)

Abstrak-Universitas Islam Indonesia (UII) merupakan salah satu perguruan tinggi yang sudah menggunakan aplikasi berbasis *web* untuk melakukan proses penerimaan mahasiswa baru. Proses ini menggunakan aplikasi yang bernama UIIAdmisi sejak tahun 2016. Akan tetapi, aplikasi tersebut memiliki ketertinggalan teknologi berupa *source code* yang kurang rapi dan terstruktur serta membutuhkan fitur baru yaitu *landing page* untuk mengategorikan pengguna dan fitur pencarian Nomor Induk Utama (NIU) bagi pengguna yang melupakan NIU mereka berdasarkan Nomor Induk Kependudukan (NIK). Oleh karena itu, dibutuhkan sebuah solusi yang dapat membantu *developer* menyederhanakan kode dan proses *coding*. Berdasarkan masalah tersebut penelitian ini bertujuan untuk mengimplementasikan *binding* sebagai solusi kerapian kode dan peremajaan *website*. *Binding* diimplementasikan menggunakan *framework* Angular dengan metode *agile* yang mengedepankan kecepatan sebagai metode penelitian. Hasil dari penelitian ini, *binding* dapat digunakan untuk merapikan struktur *source code* dengan mengurangi jumlah baris dan menyederhanakan proses *coding* sehingga menghasilkan *source code* yang lebih singkat dan struktur proyek yang tertata.

Kata Kunci: *Binding, Angular, Admisi, Agile, Web*

Abstract-Universitas Islam Indonesia (UII) is one of the universities that has used a web-based application to conduct a new student admission process. This process has used an application called UIIAdmisi since 2016. However, the application has a technological lag in the form of *source code* that is not neat and structured and requires new features, namely a *landing page* to categorize users and a Nomor Induk Utama (NIU) search feature for users who forget their NIU based on the Nomor Induk Kependudukan (NIK). Therefore, a solution is needed that can help developers simplify the code and coding process. Based on this problem, this research aims to implement bindings as a solution to code neatness and website rejuvenation. Bindings are implemented using the Angular framework with agile methods prioritizing speed as a research method. As a result of this research, bindings can be used to tidy up the source code structure by reducing the number of lines and simplifying the coding process, resulting in shorter source code and an organized project structure.

Keywords: *Binding, Angular, Admission, Agile, Website*

1 PENDAHULUAN

Salah satu universitas yang mengembangkan serta menggunakan proses digitalisasi adalah Universitas Islam Indonesia (UII). Universitas tersebut mulai menggunakan layanan penerimaan mahasiswa baru secara daring sejak tahun 2016 melalui laman admisi.uui.ac.id, aplikasi ini diberi nama UIIAdmisi. Menurut laporan kerja UII pada tahun 2022, aplikasi UIIAdmisi masih memiliki teknologi yang tertinggal sehingga menyulitkan pengembangan di masa depan. Hal ini dikarenakan pengembangan UIIAdmisi masih menggunakan pemrograman berbasis *web* secara *native* karena *framework* memiliki efisiensi pembuatan fungsi kode program yang lebih tinggi dibandingkan *native*[1]. Hasil pada sebuah penelitian yang membahas tentang perbandingan penggunaan *framework* dan *native* mengatakan bahwa proses *coding* lebih rapi dan lebih efektif ketika menggunakan *framework* dibandingkan tanpa menggunakan *framework* [2].

Berdasarkan permasalahan di atas, penelitian ini mengajukan sebuah penyelesaian dengan menggunakan *binding* pada *framework* Angular. *Binding* adalah menyisipkan potongan kode, nilai variabel, fungsi bahkan sebuah komponen satu ke komponen lain [3]. Sedangkan, *framework* Angular merupakan sebuah *framework* yang berfungsi untuk mengembangkan sebuah aplikasi berbasis *web* dengan HTML dan bahasa pemrograman TypeScript [4]. *Source code* yang digunakan akan menjadi lebih rapi dan mudah dikembangkan berkat penggunaan *binding*[3].

Menurut hasil dari penelitian [5], *framework* dengan fitur pemrosesan data yang terbaik adalah Angular. Pada penelitian tersebut ditemukan bahwa dengan *binding*, berbagai proses bisnis yang kompleks dapat diselesaikan sehingga cocok untuk digunakan proyek berskala besar. Ditambah, dengan dukungan yang stabil dan dapat diandalkan dari Google, *framework* Angular merupakan pilihan yang tepat untuk mengimplementasikan *binding*.

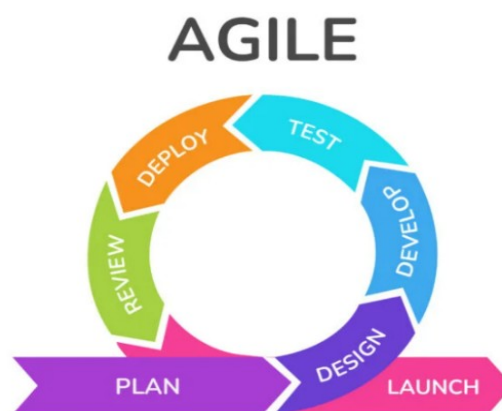
Tujuan dari penelitian ini adalah mengimplementasikan *binding* sebagai solusi penyederhanaan kode untuk aplikasi UIIAdmisi. Dengan disederhanakannya kode aplikasi UIIAdmisi tersebut maka pengembangan aplikasi di masa depan dapat dipermudah dan menghilangkan ketertinggalan teknologi yang sudah disebutkan pada laporan UII tahun 2022. Oleh sebab itu, penelitian yang bertujuan untuk mengimplementasikan *binding* ini dilakukan dengan harapan menyelesaikan permasalahan kerapian kode dan ketertinggalan teknologi.

Objek penelitian yang diterapkan adalah penerimaan mahasiswa baru Universitas Islam Indonesia. Pada penelitian ini, aplikasi yang dikembangkan berbentuk aplikasi berbasis *web* dengan fokus pada pihak *front-end*. Tahapan pengembangan akan menjadi pembahasan utama pada penelitian ini. Dalam makalah ini akan tetap menyantumkan tahapan-tahapan lain, meskipun lebih banyak dititikberatkan pada pembahasan tahap pengembangan. Selain itu, penelitian ini memiliki batasan pada jumlah pola seleksi yang akan digunakan. Pola seleksi yang akan digunakan hanya berjumlah satu pola seleksi pada setiap proses pendaftaran melalui aplikasi UIIAdmisi.

Dalam penelitian ini dibutuhkan pemilihan metode yang cocok agar proses pengimplementasian berjalan lancar. Salah satu metode yang dapat digunakan adalah metode *agile*. Metode ini menggunakan pendekatan proses yang bersifat berulang-ulang, inkremental dan iteratif[6]. Metode *agile* dipilih karena kebutuhan proses bisnis dari UII yang dapat berganti secara mendadak. Tidak hanya itu, tenggat waktu yang diperlukan juga hanya sedikit. Hal ini terjadi karena UII harus melayani berbagai *stakeholder* sekaligus. Para *stakeholder* tersebut juga memiliki kepentingan yang berbeda-beda dengan lini masa yang juga beragam. Oleh sebab itu, agar dapat memenuhi kepentingan dan kebutuhan *stakeholder*, penelitian ini menggunakan metode *agile* yang memiliki tempo cepat dan adaptabilitas tinggi [7].

2 METODE PENELITIAN

Metode *agile* memiliki tujuh tahapan yang terdiri dari perencanaan, desain, pengembangan, pengujian, pengimplementasian, pengulasan, dan peluncuran. Tujuh tahapan tersebut dapat dilihat pada Gambar 1. Dengan menggunakan metode *agile*, hasil yang dikeluarkan berskala kecil dan memiliki tempo yang teratur.



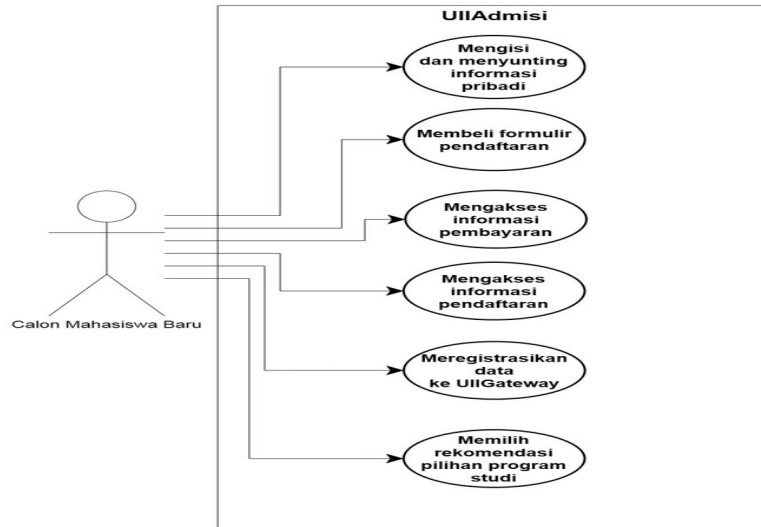
Gambar 1. Ilustrasi proses metode agile[8]

Penggunaan *binding* dilakukan pada tahap pengembangan, dengan tiga tipe yaitu *Interpolation Property Attribute Class Style*, *Event*, dan *Two-way*[9]. *Interpolation Property Attribute Class Style* adalah *data binding* satu arah untuk mengganti nilai di fail HTML yang akan ditampilkan. *Syntax* yang digunakan biasanya menggunakan tanda `{{}}`. *Event binding* adalah metode untuk mengikat suatu fungsi atau *behavior* sebuah komponen HTML, bukan hanya nilainya saja. *Syntax* yang digunakan biasanya (nama *event*)=“*statement event*”. *Two-way binding* adalah proses pengikatan secara dua arah yaitu dari fail HTML atau juga dari fail model suatu tampilan (fail TypeScript) yang bersangkutan. Sebagai contoh yaitu `<input [(ngModel)]="name">`[9].

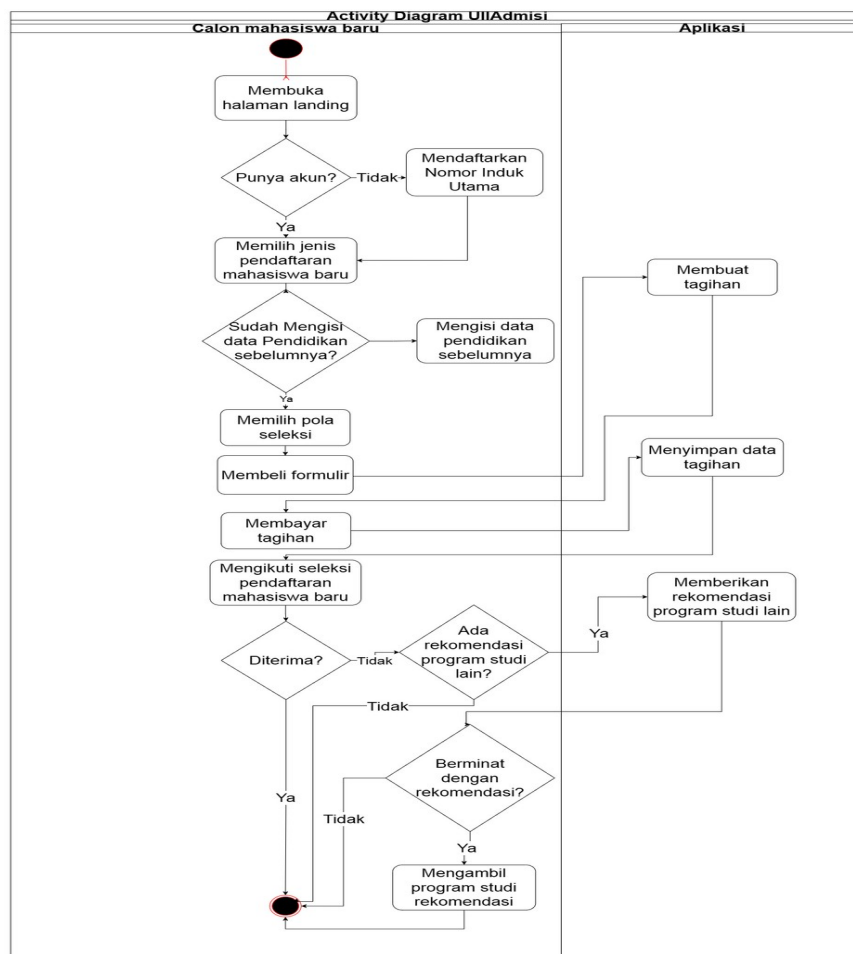
2.1 Tahap Perencanaan

Pada tahap pertama iterasi ini, kegiatan yang dilakukan adalah membuat rancangan rencana-rencana apa saja yang akan dilakukan pada satu iterasi ke depan, tujuan iterasi pada saat ini, dan hasil-hasil yang ingin dicapai. Dengan begitu, iterasi pada saat ini dapat berjalan lancar. Artefak yang didapatkan pada tahap pertama ini adalah sebuah *Use Case Diagram* dan *Activity Diagram*. *Use Case Diagram* merupakan sebuah diagram ilustrasi aksi-aksi interaksi apa saja yang dapat dilakukan oleh pengguna dalam menggunakan aplikasi[10].

Use Case Diagram untuk aplikasi UIIAdmisi dapat dilihat pada Gambar 2. Pada use case diagram terdapat lima aksi-aksi penting yang dapat dilakukan oleh pengguna UIIAdmisi. Aksi-aksi tersebut adalah pengisian dan penyuntingan informasi pribadi, pembelian formulir pendaftaran, mengakses informasi pembayaran, mengakses informasi pendaftaran, meregistrasikan data ke UIIGateway, dan memilih rekomendasi program studi yang ditawarkan.



Gambar 2. Use Case Diagram UIIAdmisi



Gambar 3. Activity Diagram UIIAdmisi

Activity diagram adalah sebuah ilustrasi yang berbentuk sebuah *flowchart* untuk menjelaskan suatu proses dan gabungan semua aksi-aksi oleh pengguna secara kronologis bermula dari awal hingga akhir[11]. Dengan dibuatnya sebuah *activity diagram*, *developer* serta *system analyst* dapat memetakan manakah fungsi yang penting dan harus dibuatkan sebuah halaman khusus. *Activity diagram* UIIAdmisi dapat dilihat pada Gambar 3.

2.2 Tahap Desain

Pada tahap kedua iterasi ini adalah tahap desain aplikasi. Pada tahapan ini mulai dibuat desain untuk antarmuka aplikasi. Tahap ini dilaksanakan menggunakan Figma sebagai alat untuk mendesain tampilan utama UIIAdmisi. Nantinya desain ini akan digunakan sebagai referensi bagi *developer* ketika melakukan *coding* tampilan UIIAdmisi.

2.3 Tahap Pengembangan

Pada tahap ketiga iterasi inilah proses utama dilaksanakan. Seperti namanya, pengembangan aplikasi UIIAdmisi dilaksanakan pada tahap ini. Mulai dari pembuatan basis data, penulisan kode untuk tampilan aplikasi, hingga menghubungkan *back-end* dengan *front-end*.

2.3.1. Halaman Landing Page

Halaman ini merupakan halaman pertama yang akan diakses untuk memilih kategori tipe pengguna UIIAdmisi. Terdapat lima kategori pengguna UIIAdmisi, kategori-kategori tersebut yaitu: mahasiswa baru, mahasiswa baru pascasarjana, mahasiswa baru dari luar negeri, dosen atau tenaga kependidikan, dan orang tua mahasiswa. Daftar kategori pengguna dibuat terpisah karena keperluan kategori yang cukup banyak. Selain itu, hal ini juga membuat penambahan kategori menjadi lebih mudah apabila di masa depan terdapat kategori baru.

Di halaman ini, penulis menggunakan *binding* untuk merubah warna pilihan kategori serta menentukan kata-kata yang akan ditampilkan. Penggunaan *binding* tersebut ada pada Gambar 4 hingga Gambar 7. Cara kerja *binding* adalah ketika fail HTML landing card dimuat, ia akan mengulang pemanggilan komponen kategori sebanyak jumlah kategori yang tersedia pada *array* *itemAction*. Kemudian pada komponen kategori yang bersangkutan akan mengatur nilai warna, atribut *title*, atribut *subtitle*, dan aksi yang akan dilakukan kategori tersebut menggunakan *binding*.

Tipe *binding* yang digunakan ada dua yaitu *interpolation binding* dan *Event binding*. *Interpolation binding* digunakan untuk mengatur atribut *title* dan *subtitle* pada komponen kategori yang bersangkutan. Kemudian warna komponen kategori tersebut diatur dengan fungsi *onClickEvent*. Fungsi tersebut akan merubah nilai *colorState* bergantung pada aksi apa yang sedang pengguna lakukan. Untuk mengganti warna, komponen kategori melakukan *binding* pada atribut *style.background-color*. Ketika ada pemanggilan fungsi tadi, fail HTML dapat merubah warna mengikuti nilai *colorState*-nya.

```

1 <button (mouseover)="onClickEvent($event)" (mouseleave)="onClickEvent($event)" (mousedown
2 (click)="onClick(title)" class="item-card" [style.background-color]="colorState">
3 <div class="d-flex flex-row justify-content-between align-items-center">
4 <div style="gap: 15px;" class="d-flex flex-row">
5 <div class="file-icon d-flex justify-content-center">
6 | <i class="fa-solid fa-file fa-xl"></i>
7 </div>
8 <div style="max-width: 100%; padding-right: 10px;" class="d-flex flex-column">
9 <h2>{{title}}</h2>
10 <p>{{subtitle}}</p>
11 </div>
12 </div>
13 | <i class="fa-solid fa-chevron-right right-arrow fa-2x"></i>
14 </div>
15 </button>
16

```

Gambar 4. Penggunaan binding pada pilihan kategori landing page

Gambar 4 merupakan tangkapan layar dengan penggunaan *binding* pada pilihan kategori landing page. Baris sembilan adalah pemanggilan terhadap variabel *title* untuk memberikan teks judul kepada pilihan kategori. Sedangkan baris sepuluh merupakan pemanggilan kepada variabel *subtitle* yang berisikan nilai penjelas kategori tersebut dengan tipe data *string*. *Binding* akan menyesuaikan setiap nilai yang diberikan sehingga tidak perlu menuliskannya berulang-ulang.

```

14 export class LandingCardComponent implements OnInit {
15   @Input() title: string;
16   @Input() subtitle: string;
17   @Input() color = '';
18   @Input() colorHover = '';
19   @Input() colorClicked = '';
20   @Input() actionCode: string;
21
22   colorState = '';
23
24   public storeKey: string;
25   constructor(private router: Router, private modalSvc: BsModalService) {
26     this.storeKey = window.location.origin;
27   }
28
29   ngOnInit(): void {
30     this.colorState = this.color;
31   }
32
33   onClickEvent(event) {
34     if (event.type === 'mousedown') {
35       this.colorState = this.colorClicked;
36     } else if (event.type === 'mouseover') {
37       this.colorState = this.colorHover;
38     } else {
39       this.colorState = this.color;
40     }
41   }

```

Gambar 5. Fungsi untuk mengatur warna pilihan kategori

Pengaturan warna pada setiap pilihan kategori pengguna terdapat pada Gambar 5. Pada baris 33, terdapat fungsi bernama *onClickEvent* yang membutuhkan *parameter* bertipe data *Event* untuk membaca jenis *event* yang dilakukan oleh pengguna. Kemudian fungsi tersebut akan menyortir apakah *Event* tersebut berupa klik, *hover* atau tidak ada aktivitas apapun.

```

src > app > core > account > landing-page > landing-page.component.html > div.ui-landing
Go to component
1 <div class="ui-landing">
2   <div class="d-flex flex-column flex-md-row justify-content-md-between landing-content">
3
4     <div class="d-flex flex-column justify-content-between logo-text">
5       
6       <div>
7         <h3 id="welcome-text">Welcome to</h3>
8         <h1 id="web-name"><span style="color: #7d7000;">UII</span>Admisi</h1>
9       </div>
10    </div>
11
12    <div class="right-menu">
13
14      <div>
15        <h1>Login</h1>
16        <p>Please select the category of program!</p>
17      </div>
18
19      <div class="right-menu-list">
20        <ul>
21          <li *ngFor="let item of itemAction.student">
22            <uii-landing-card
23              [title]="item.title"
24              [subtitle]="item.subtitle"
25              [color]="item.color"
26              [colorHover]="item.colorHover"
27              [colorClicked]="item.colorClick"
28              [actionCode]="item.actionCode">
29            </uii-landing-card>
30          </li>
31        </ul>
32      </div>
33    </div>
34  </div>
35
36 </div>
37

```

Gambar 6. Fail HTML utama halaman landing untuk memanggil komponen pilihan kategori

Penggunaan *binding* dengan tipe *Interpolation Property Attribute Class Style* terdapat pada Gambar 6. Baris 23 hingga 28 merupakan implementasi *binding* tersebut. Nilai variabel yang dipanggil pertama kali adalah *title* pada setiap kategori pengguna. Variabel kedua yang akan dipanggil adalah *subtitle* yang dimiliki oleh setiap kategori pengguna. Pemanggilan variabel tersebut akan berlanjut hingga variabel terakhir yaitu kode kategori pengguna yang berguna untuk memetakan tipe pengguna UIIAdmisi.

```

22  ngOnInit(): void {
23    this.itemAction.student = [
24      {
25        title: 'Penerimaan Mahasiswa Baru',
26        subtitle:
27          'Pendaftaran dan penerimaan mahasiswa baru UII untuk semua program Diploma, Sarjana Terapan, dan Sarjana.',
28        color: '#333333',
29        colorHover: '#333333',
30        colorClick: '#333333',
31        actionCode: LOGIN_KATEGORI.CALON_MAHASISWA_BARU,
32      },
33      {
34        title: 'Penerimaan Mahasiswa Baru Pasca Sarjana',
35        subtitle:
36          'Pendaftaran dan penerimaan mahasiswa baru UII untuk semua program Profesi, Magister, dan Doktor.',
37        color: '#333333',
38        colorHover: '#333333',
39        colorClick: '#333333',
40        actionCode: LOGIN_KATEGORI.PASCA_SARJANA,
41      },
42      {
43        title: 'International Admission',
44        subtitle:
45          'Admission for foreigners (please use Google Chrome browser to access).',
46        color: '#333333',
47        colorHover: '#333333',
48        colorClick: '#333333',
49        actionCode: LOGIN_KATEGORI.INTERNATIONAL_ADMISSION,
50      },
51      {
52        title: 'Rekrutmen Dosen atau Tenaga Kependidikan',
53        subtitle:
54          'Rekrutmen dosen atau tenaga kependidikan Universitas Islam Indonesia.',
55        color: '#333333',
56        colorHover: '#333333',
57        colorClick: '#333333',
58        actionCode: LOGIN_KATEGORI.CALON_DOSEN_BARU,
59      },
60      {
61        title: 'Orang Tua Mahasiswa',
62        subtitle:
63          'Layanan buat akun NIU untuk memantau perkembangan akademik putra/putri yang bersangkutan dalam menjalani studi di UII melalui gateway.uui.ac.id.',
64        color: '#333333',
65        colorHover: '#333333',
66        colorClick: '#333333',

```

Gambar 7. Fail model untuk menentukan kategori apa saja yang akan dimunculkan

Pemberian nilai variabel-variabel dilakukan pada fail model yang terdapat pada Gambar 7. Pada baris 23 merupakan inisialisasi *array* dari atribut *itemAction* dengan nama variabel *student*. Di dalamnya terdapat tipe-tipe kategori yang tersedia pada UIIAdmisi yaitu mahasiswa baru, mahasiswa pascasarjana, mahasiswa internasional, dosen dan tenaga pendidik, dan orang tua mahasiswa. Setiap kategori memiliki objek di dalamnya dengan nilai yang bervariasi.

2.4 Tahap Pengujian

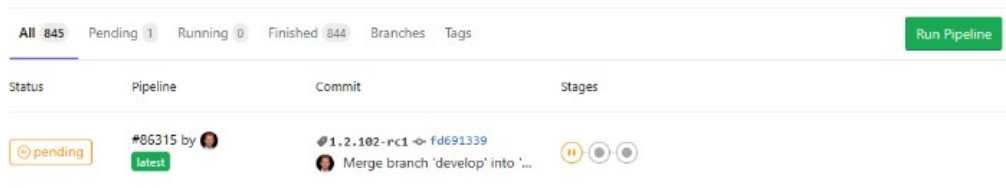
Setelah melalui tahap pengembangan, tahap selanjutnya adalah pengujian fitur-fitur yang ada pada UIIAdmisi. Pengujian dilakukan secara manual dengan metode *blackbox testing*. *Blackbox testing* merupakan sebuah pengujian yang tidak melihat struktur kode dan memosisikan diri sebagai pengguna aplikasi [12]. Tabel butir-butir pengujian dapat dilihat pada Tabel 1. UIIAdmisi diuji oleh dua orang UI/UX Designer yang memberikan respon bahwa setiap pengujian berhasil dilakukan sehingga tidak perlu melakukan proses *debugging*.

Tabel 1. Daftar pengujian UIIAdmisi

Halaman Pengujian	Target Pengujian	Hasil Pengujian
Landing page	Pengguna berhasil masuk dan memilih kategori	Sesuai
Pendaftaran	Informasi formulir dan tagihan berhasil ditampilkan	Sesuai
Detail tagihan	Informasi detail tagihan berhasil ditampilkan	Sesuai
Beli formulir	Formulir berhasil ditampilkan dan dibeli oleh pengguna	Sesuai
Rekomendasi	Pengguna dapat memilih program studi yang direkomendasikan	Sesuai
Registrasi	Pengguna berhasil diarahkan ke UIIGateway.	Sesuai

2.5 Tahap Implementasi

Setelah melalui pengujian dan aplikasi sudah siap untuk digunakan, maka tahap selanjutnya adalah tahap implementasi. Aplikasi akan digunakan oleh pengguna secara langsung. Proses implementasi ini menggunakan Gitlab yang dapat dilihat pada Gambar 8.



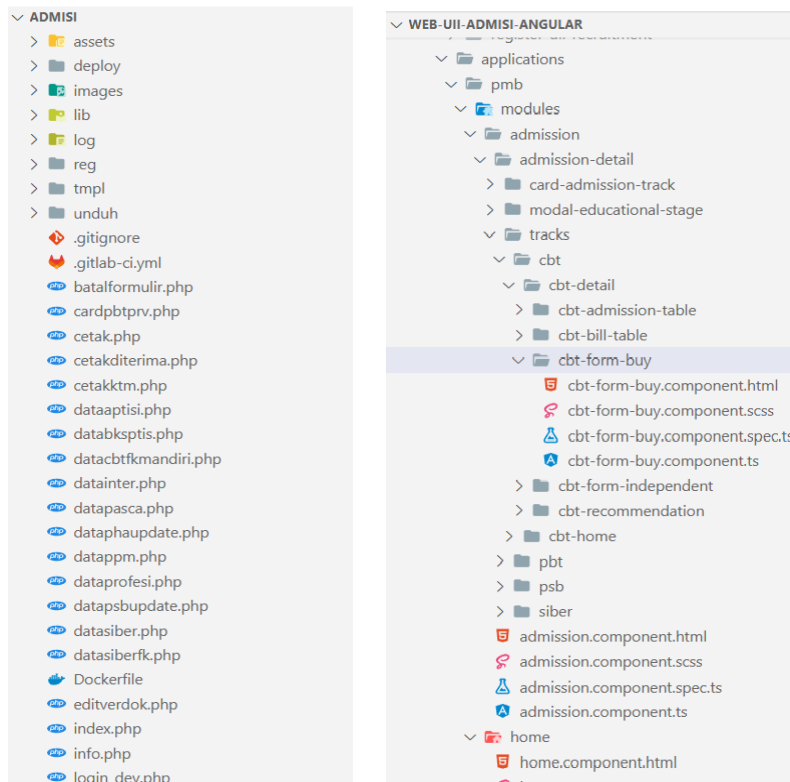
Gambar 8. Proses deploy ke Gitlab

3 HASIL DAN PEMBAHASAN

Penelitian ini berhasil menunjukkan cara kerja *binding* menggunakan *framework* Angular. Selain itu, didapatkan hasil bahwa jumlah fail pada UIIAdmisi baru yang menggunakan *binding* lebih banyak. Jumlah fail yang ada pada UIIAdmisi *native* sebanyak 441 fail, sedangkan UIIAdmisi dengan *binding* memiliki jumlah sebanyak 156.185 fail. Hal ini dikarenakan fitur bawaan milik *framework* Angular yang mengharuskan UIIAdmisi memunculkan fail-fail selain fail PHP pada UIIAdmisi *native*. Hasil ini dibuktikan pada Gambar 9.

Salah satu hasil pengembangan yang dilakukan menggunakan *binding* yaitu halaman *landing page* untuk mengategorikan pengguna UIIAdmisi. Tipe *binding* yang digunakan pada halaman tersebut adalah *Interpolation Property Attribute Class Style* dan *Event binding*. *Interpolation property attribute class style* digunakan untuk mengirimkan nilai-nilai variabel dari fail logika kepada fail HTML untuk ditampilkan pada antarmuka sistem. Sedangkan *event binding* digunakan untuk mengirimkan fungsi-fungsi yang berupa instruksi untuk mengganti warna pada tombol pemilihan kategori pengguna.

Melalui proses pengujian sistem dengan metode *blackbox testing*, diperoleh hasil bahwa setiap fungsi yang dikembangkan berhasil diuji. Setiap interaksi dan aksi-aksi yang diuji pada Tabel 1 berhasil diimplementasikan menggunakan *binding* pada UIIAdmisi. Dengan hasil tersebut, tidak diperlukan proses tambahan yaitu proses *debugging*. Oleh karena itu, penggunaan *binding* berhasil diimplementasikan kepada UIIAdmisi dengan metode Scrum dan *framework* Angular.

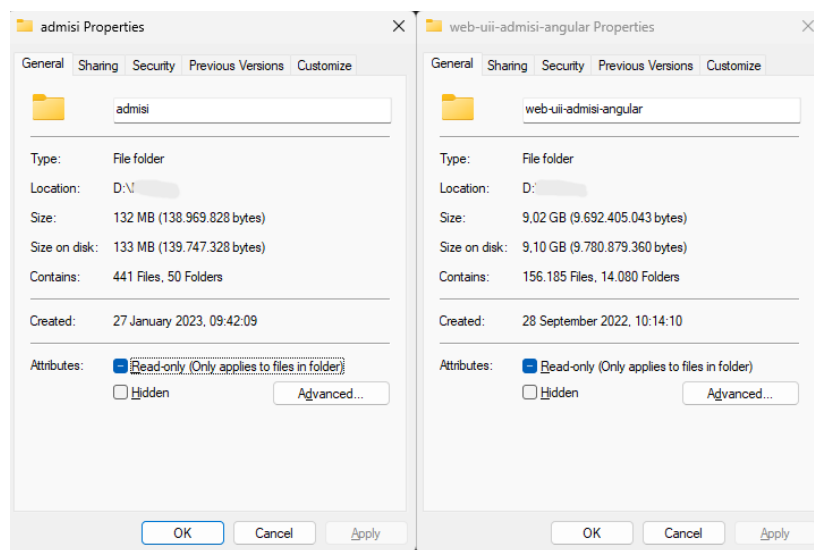
Gambar 9. Perbandingan struktur folder UIIAdmisi *native* (kiri) dan UIIAdmisi yang menggunakan *binding* (kanan)

Hasil kedua yang didapat adalah jumlah baris yang dimiliki oleh setiap halaman berkurang secara signifikan. Hal ini disebabkan karena penggunaan *binding* menghilangkan kebutuhan pada fungsi logika untuk merubah konten HTML secara manual. Dengan hilangnya hal tersebut baris-baris kode yang melakukan fungsi tersebut dapat dihapus sehingga mengurangi jumlah baris pada setiap fail. Perbandingan jumlah baris kode dapat dilihat pada Gambar 10.



Gambar 10. Perbandingan jumlah baris pada fail beli formulir UIAdmisi *native* (kiri) dan UIAdmisi yang menggunakan *binding* (kanan)

Ditemukan hasil lain yaitu ukuran proyek UIAdmisi yang menggunakan *binding* lebih besar dibandingkan yang tidak. Hal ini berkaitan dengan jumlah fail yang dimunculkan lebih banyak pada UIAdmisi yang menggunakan *binding* daripada UIAdmisi *native*. Dengan jumlah fail yang semakin banyak maka ukuran yang dihasilkan juga ikut bertambah. Pembuktian terhadap hasil ini dapat dilihat pada Gambar 11.



Gambar 11. Perbandingan ukuran UIAdmisi *native* (kiri) dengan UIAdmisi menggunakan *binding* (kanan)

4 KESIMPULAN

Berdasarkan penelitian yang telah dilakukan maka dapat ditarik kesimpulan bahwa penggunaan *binding* kepada aplikasi UIAdmisi berpengaruh dalam kerapian dan kemudahan pengembangan *source code*. *Source code* yang digunakan lebih rapi karena jumlah baris yang digunakan lebih sedikit. Hal ini dikarenakan kemudahan dalam mengakses perubahan nilai variabel dan pengimplementasian ke dalam tampilan UIAdmisi. Hasil dari pengujian *blackbox testing* terhadap penggunaan *binding* untuk pengembangan *website* penerimaan mahasiswa baru berhasil diimplementasikan. Dengan begitu, ketika terdapat perubahan proses bisnis atau kebutuhan *stakeholder* di masa depan *developer* tidak perlu merubah kembali struktur kode dengan mengganti konten HTML secara manual. Oleh sebab itu, penggunaan *binding* sangat signifikan pada pengembangan aplikasi berbasis *web*.

DAFTAR PUSTAKA

- [1] Y. Endra, Y. Aprilinda, Y. Y. Dharmawan, and W. Ramadhan, "Analisis Perbandingan Bahasa Pemrograman PHP Laravel dengan PHP Native pada Pengembangan Website," *Expert J. Manaj. Sist. Inf. dan Teknol.*, vol. 11, no. 1, pp. 48–55, Jun. 2021.
- [2] W. Setiawan dan H. Sama, "STUDI KOMPARASI PENGEMBANGAN WEBSITE MENGGUNAKAN FRAMEWORK DAN NON FRAMEWORK: EFEKTIVITAS DAN KUSTOMISASI," *Conference on Business, Social Sciences and Innovation Technology*, vol. 1, no. 1, hlm. 622–629, Nov 2020, Diakses: 1 Juni 2023. [Daring]. Tersedia pada: <https://journal.uib.ac.id/index.php/cbssit/article/view/1470>
- [3] E. Özdikililer, "Data Binding in Front End for Web Applications," *The Journal of CIEES*, vol. 1, no. 2, hlm. 31–34, Des 2021, doi: 10.48149/JCIEES.2021.1.2.6.
- [4] M. Kaluža dan B. Vukelić, "Comparison of front-end frameworks for web applications development," *Zbornik Veleučilišta u Rijeci*, vol. 6, no. 1, hlm. 261–282, 2018, doi: 10.31784/ZVR.6.1.19.
- [5] Y. K. Xing, J. P. Huang, dan Y. Y. Lai, "Research and analysis of the front-end frameworks and libraries in e-business development," *ACM International Conference Proceeding Series*, hlm. 68–72, Feb 2019, doi: 10.1145/3313991.3314021.
- [6] I. Larasati, A. N. Yusril, dan P. Al Zukri, "Systematic Literature Review Analisis Metode Agile Dalam Pengembangan Aplikasi Mobile," *Sistemasi: Jurnal Sistem Informasi*, vol. 10, no. 2, hlm. 369–380, Mei 2021, Diakses: 11 Mei 2023. [Daring]. Tersedia pada: <http://sistemasi.ftik.unisi.ac.id/index.php/stmsi/article/view/1237>
- [7] F. Nurzaman, "PENGEMBANGAN SISTEM OTOMATISASI TAGIHAN MENGGUNAKAN METODE AGILE SOFTWARE DEVELOPMENT," Jakarta Pusat, Mar 2020. Diakses: 1 Juni 2023. [Daring]. Tersedia pada: <http://journals.upi-yai.ac.id/index.php/ikraith-informatika/article/download/619/462>
- [8] A. Jose Luis, "The Agile Development Process for Mobile Apps | Krasamo." <https://www.krasamo.com/agile-development-process/> (diakses 11 Mei 2023).
- [9] "Angular - Binding syntax." <https://angular.io/guide/binding-syntax> (diakses 8 Mei 2023).
- [10] A. A. Arsyad, M. Mashud, dan A. Sumardin, "Implementasi Metode Agile Scrum Pada Sistem Informasi Akuntansi CV Tritama Inti Persada," *Jurnal Ilmiah Sistem Informasi Akuntansi*, vol. 2, no. 2, hlm. 82–87, Nov 2022, Diakses: 12 Mei 2023. [Daring]. Tersedia pada: <http://jim.teknokrat.ac.id/index.php/jimasia/article/view/2241>
- [11] W. Sornkliang dan T. Phetkaew, "Performance Analysis of Test Path Generation Techniques Based on Complex Activity Diagrams," *Informatica*, vol. 45, no. 2, hlm. 231–242, Jun 2021, doi: 10.31449/INF.V45I2.3049.
- [12] A. C. Praniffa, A. Syahri, F. Sandes, U. Fariha, Q. A. Giansyah, dan M. L. Hamzah, "Pengujian Sistem Informasi Parkir Berbasis Web Pada UIN SUSKA RIAU Menggunakan White Box dan Black Box Testing," *Jurnal Testing dan Implementasi Sistem Informasi*, vol. 1, no. 1, hlm. 1–6, Mar 2023, Diakses: 1 Juni 2023. [Daring]. Tersedia pada: <http://www.journal.al-matani.com/index.php/jtisi/article/view/321>