



# Pembuatan Back-End Sistem Informasi Kerja Praktik pada Departemen Teknik Komputer Menggunakan Framework Laravel

## *Creating a Back-End Information System for Practical Work at the Department of Computer Engineering, Diponegoro University*

Indriawan Muhammad Akbar<sup>\*1)</sup>, Risma Septiana<sup>2)</sup>, Patricia Evericho Mountaines<sup>3)</sup>

<sup>1,2,3)</sup>Program Studi Teknik Komputer, Fakultas Teknik, Universitas Diponegoro  
Jl. Prof. Soedarto, SH, Kampus Undip Tembalang, Semarang, Indonesia 50275

---

**How to cite:** I. M. Akbar, R. Septiana, and P. E. Mountaines, "Pembuatan Back-End Sistem Informasi Kerja Praktik pada Departemen Teknik Komputer Menggunakan Framework Laravel," *Jurnal Teknologi dan Sistem Komputer*, vol. 2, no. 3, pp. 233-239, Oct 2023. doi: 10.14710/jtk.v2i3.38985 [Online].

---

**Abstract** – *The Practical Work Information System is a web application that was built to facilitate practical work information management for students and lecturers in the Department of Computer Engineering, Diponegoro University. This system development methodology adopts the Waterfall method, through the stages of needs analysis, system design, implementation, testing, and maintenance. In the needs analysis stage, literature studies and direct observations are carried out to identify existing problems. The Job Training Information System uses the Laravel framework and MySQL database, by implementing the Model-View-Controller (MVC) concept to ensure structured and organized application development. System testing shows the success of the Practical Work Information System in facilitating online Practical Work submissions by students, speeding up the process of verification and validation by lecturers, and producing reports and statistics that facilitate decision making.*

**Key Terms:** *Information System, Practical Work, Laravel, Waterfall*

**Abstrak** – *Sistem Informasi Kerja Praktik merupakan sebuah aplikasi web yang dibangun untuk memfasilitasi manajemen informasi Kerja Praktik bagi mahasiswa dan dosen di Departemen Teknik Komputer Universitas Diponegoro. Metodologi pengembangan sistem ini mengadopsi metode Waterfall, melalui tahap analisis kebutuhan, perancangan sistem, implementasi, pengujian, dan pemeliharaan. Dalam tahap analisis kebutuhan, dilakukan studi literatur dan observasi langsung untuk mengidentifikasi permasalahan yang ada. Sistem Informasi Kerja Praktik menggunakan kerangka kerja Laravel dan basis data MySQL, dengan mengimplementasikan konsep Model-View-Controller (MVC) untuk memastikan pengembangan aplikasi yang*

*terstruktur dan terorganisir. Pengujian sistem menunjukkan keberhasilan Sistem Informasi Kerja Praktik dalam memfasilitasi pengajuan Kerja Praktik secara daring oleh mahasiswa, mempercepat proses verifikasi dan validasi oleh dosen, serta menghasilkan laporan dan statistik yang memudahkan pengambilan keputusan.*

**Kata Kunci:** *Sistem Informasi, Kerja Praktik, Laravel, Waterfall*

### I. PENDAHULUAN

Kerja Praktik (KP) adalah salah satu persyaratan untuk memperoleh gelar sarjana yang harus dilakukan oleh mahasiswa Departemen Teknik Komputer di Universitas Diponegoro (Undip). Tujuan dari KP adalah memberikan pengalaman pada mahasiswa untuk memahami situasi di dunia kerja yang sebenarnya. Dalam pelaksanaannya, mahasiswa diharuskan mencari tempat magang di perusahaan selama periode 1-6 bulan. Setelah menyelesaikan masa magang dan mendapat penilaian dari pembimbing lapangan, mahasiswa harus menyusun laporan dan makalah KP dengan bimbingan dosen pembimbing. Terakhir, mahasiswa harus menyampaikan hasil seminar KP dalam waktu 6 bulan setelah periode magang berakhir untuk mendapatkan nilai mata kuliah KP.

Agar pelaksanaan kerja praktik berjalan dengan lancar, mahasiswa harus melengkapi berkas resmi dari Departemen Teknik Komputer. Sayangnya, prosedur pengajuan hingga pemberkasan kerja praktik di Departemen Teknik Komputer Undip masih memiliki beberapa kekurangan. Contohnya, beberapa mahasiswa sudah memulai kerja praktik namun belum melapor ke koordinator kerja praktik. Selain itu, beberapa mahasiswa tidak mengetahui tempat pengumuman kerja praktik, langkah-langkah untuk mencari tempat kerja praktik, serta pengumpulan berkas kerja praktik masih menggunakan cara manual. Dalam cara manual ini, setiap

<sup>\*</sup>Penulis Korespondensi (Zulfa Fatah Akbar Ahmad)  
Email: [zulfafatahakar@gmail.com](mailto:zulfafatahakar@gmail.com)



mahasiswa harus merekap semua berkas yang diperlukan menggunakan aplikasi Google Drive dan mengirimkan link-nya melalui email atau Whatsapp ke koordinator kerja praktik. Cara pengumpulan berkas seperti ini dinilai tidak efektif karena koordinator kerja praktik harus mengecek email atau Whatsapp secara berkala.

Sistem informasi memiliki peranan penting dalam suatu institusi. Dengan adanya sistem informasi, institusi dapat menjamin kualitas informasi yang diberikan dan dapat mengambil keputusan dengan cepat dan tepat [1]. Kebutuhan informasi yang cepat, tepat, dan akurat, membuat institusi perlu melakukan pemanfaatan sistem informasi, tidak terkecuali pada Departemen Teknik Komputer Undip. Sekalipun di Departemen Teknik Komputer Undip sudah terdapat Sistem Informasi Kerja Praktik. Namun, sistem yang ada saat ini hanya dibuat dari sisi admin kampus guna mendata siapa saja mahasiswa yang sedang mengambil mata kuliah KP di suatu semester dan seberapa banyak kuota bimbingan KP per dosen. Dengan kata lain, sistem saat ini tidak mampu menjawab kebutuhan mahasiswa dan tidak dapat menjadi jembatan penghubung antara mahasiswa KP dengan dosen pembimbing KP dan koordinator KP. Demi melancarkan proses pengajuan KP bagi mahasiswa dan membantu setiap *stakeholder* yang terlibat hingga proses pemberkasan KP, penulis memandang bahwa perlu dibuat sebuah sistem informasi kerja praktik yang memiliki fitur lengkap agar mata kuliah Kerja Praktik di Departemen Teknik Komputer dapat berjalan dengan lebih baik.

Pembuatan Sistem Informasi Kerja Praktik ini dilakukan dengan menggunakan kerangka kerja Laravel. Kerangka kerja Laravel menggunakan arsitektur *Model-View-Controller* (MVC) yang mana akan memisahkan komponen aplikasi menjadi model, view, dan controller. Dengan memisahkan ketiga komponen tersebut, maka pengembang dapat lebih mudah memahami dan mengelola kode aplikasi.

Pembuatan *backend* aplikasi dilakukan dengan menggunakan metode *waterfall* yang merupakan salah satu model pengembangan perangkat lunak. Metode ini diawali dengan menganalisis kebutuhan, merancang sistem, mengimplementasi, pengujian, dan pemeliharaan.

## II. TINJAUAN LITERATUR

### A. Penelitian Terdahulu

Penelitian yang dilakukan berdasarkan pada studi penelitian sebelumnya yang diajukan sebagai acuan dan pembanding. Studi penelitian sebelumnya merujuk pada penelitian-penelitian yang telah dilakukan dan akan digunakan sebagai referensi pada penelitian "Pembuatan Sistem Informasi Kerja Praktik Departemen Teknik Komputer".

Penelitian yang telah dilakukan oleh Masjudin Masjudin, Cakra Adipura Wicaksana, Agusutrisno

Agusutrisno, Rian Fahrizal pada tahun 2022 dengan judul penelitian "Rancang Bangun Sistem Informasi Arsip dan Persuratan Program Studi Teknik Elektro Untirta Berbasis Framework Laravel". Sistem Arsip atau repository merupakan sistem yang digunakan sebagai media penyimpanan data berupa dokumen kearsipan seperti surat, karya tulis, laporan kerja praktek dan dokumen lainnya pada sebuah lembaga. Budaya smart and green yang diterapkan di Universitas Sultan Ageng Tirtayasa di dukung oleh Program Studi Teknik Elektro dengan menerapkan budaya paperless melalui sistem informasi repository dan persuratan. Sistem informasi ini dirancang menggunakan framework Laravel. Metode pengembangan menggunakan metode waterfall. Hasil uji implementasi menunjukkan bahwa sistem informasi arsip berjalan dengan baik dan efektif dalam membantu proses pengarsipan data-data dosen yang berkaitan dengan tridharma dan sistem persuratan membantu proses pengajuan dan pengambilan surat dengan lebih cepat, efektif, dan efisien [2].

Muhammad Hafizhuddin melakukan penelitian pada tahun 2022 dengan judul "Implementasi Framework Laravel Pada Sistem Informasi Manajemen Peserta Magang Berbasis Web" di PT. Mascitra Teknologi Informasi yang merupakan perusahaan konsultan IT dan software. PT. Mascitra Teknologi Informasi membutuhkan sebuah sistem informasi manajemen peserta magang untuk membantu mereka dalam mengelola dan menyimpan data peserta magang serta manajemen kegiatan harian peserta. Penelitian ini difokuskan pada bagian front-end dan back-end, dengan penulis bertanggung jawab untuk melakukan implementasi framework Laravel pada sistem informasi tersebut. Tujuan dari implementasi framework ini adalah untuk memudahkan developer dalam membangun aplikasi [3].

Pada penelitian yang telah dilakukan oleh Setiawan, Ridwan, Yoga Handoko Agustin, dan Intan Hartanti Rahman Ningsih dengan judul "Perancangan Sistem Informasi Kerja Praktik Berbasis Web." Dijelaskan bahwa kerja praktik adalah implementasi dari pengetahuan yang diperoleh selama kuliah ke dalam lingkungan kerja yang sebenarnya. Penelitian ini bertujuan untuk membuat sistem informasi yang dapat mengatur kegiatan kerja praktik bagi mahasiswa ITG. Penelitian ini menggunakan metodologi Rational Unified Approach (RUP) dengan tiga tahap yaitu inception, elaboration, dan construction, serta memodelkan sistem dengan menggunakan Unified Modeling Language (UML), termasuk use case diagram, activity diagram, sequence diagram, dan class diagram. Dalam hasil penelitian ini, sistem informasi dibuat untuk mengatur kegiatan kerja praktik dan memantau kemajuan pelaksanaannya, mulai dari pendaftaran hingga penyelesaian kerja praktik di tempat industri. Sistem ini juga dapat membantu mahasiswa dalam mencari

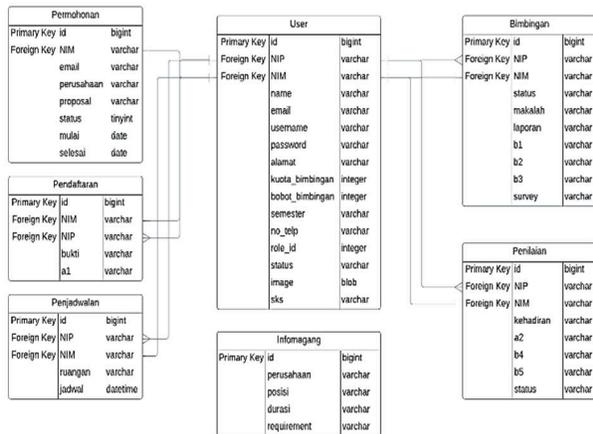




antara Server dengan sistem informasi yang tersedia di browser, serta user itu sendiri sebagai pelaku utama kegiatan distribusi data. Berikut merupakan tabel untuk menjelaskan lebih rinci mengenai arsitektur sistem dari Sistem Informasi Kerja Praktik yang diajukan

## 2. Rancangan Basis Data

Desain database untuk Sistem Informasi Kerja Praktik akan diilustrasikan melalui Table Relational berdasarkan analisis kebutuhan yang telah ditentukan sebelumnya, sebagai berikut:



Gambar 3 Relasi tabel

Pada Gambar 3 menampilkan 7 tabel dari basis data yang digunakan. Terdapat tabel user, info magang, permohonan, pendaftaran, bimbingan, penjadwalan, dan penilaian. Masing-masing tabel memiliki *primary key* dan beberapa memiliki *foreign key* untuk dijadikan penghubung suatu entitas di tabel yang berbeda.

## 3. Pembuatan kode program

Pada tahap implementasi, pengembang mulai mengimplementasikan rancangan yang telah dibuat pada tahap perancangan. Hal ini meliputi menulis kode program, membuat basis data, membangun antarmuka pengguna, dan mengintegrasikan semua komponen ke dalam satu sistem

## 4. Pengujian

Pada tahap pengujian, pengembang akan menguji sistem secara menyeluruh untuk memastikan bahwa perangkat lunak dapat berjalan dengan baik dan memenuhi semua persyaratan fungsional dan non-fungsional.

Pengujian yang dilakukan pengembang untuk menguji sistem yang telah dibuat merupakan blackbox testing. Untuk lebih detailnya metode blackbox testing artinya

menguji perangkat lunak yang dilakukan tanpa mengetahui detail implementasi internal dari sistem yang diuji. Pengujian dilakukan dengan memasukkan berbagai input ke dalam sistem atau program dan memeriksa output yang dihasilkan untuk memastikan bahwa sistem atau program berjalan sesuai dengan spesifikasi dan kebutuhan fungsional yang telah ditentukan.

## 5. Pemeliharaan

Pada tahap pemeliharaan, pengembang akan terus memantau dan memperbaiki sistem seiring waktu, sehingga sistem tetap dapat berjalan dengan baik dan memenuhi kebutuhan pengguna. Apakah masih ada kekurangan dari sistem informasi kerja praktik yang telah dibuat

## IV. IMPLEMENTASI SISTEM

### A. Implementasi

#### 1. Pembuatan Controller

Dalam pembuatan sistem informasi menggunakan kerangka kerja laravel yang menggunakan konsep *Model-View-Controller* (MVC) terdapat *controller* yang digunakan untuk menampung fungsi-fungsi yang diperlukan.

a. Fungsi index pada koorcontroller

Pada fungsi index dalam KoorController didefinisikan beberapa variabel, diantaranya digunakan untuk menampung data mahasiswa yang sudah selesai KP, Permohonan KP mahasiswa yang sudah dan belum disetujui, dan data berkas akhir yang belum disetujui. Pada bagian akhir dari fungsi ini terdapat fungsi yang mengembalikan sebuah view dengan mengirimkan data ke view tersebut.

```

class KoorController extends Controller
{
    public function index()
    {
        $mahasiswa = User::rightJoin('permohonans', 'users.NIM', '=', 'permohonans.NIM')->where('permohonans.status', 0)->count();
        $mahasiswa1 = User::rightJoin('permohonans', 'users.NIM', '=', 'permohonans.NIM')->where('permohonans.status', 1)->count();
        $dataBulan = Penilaian::leftJoin('users', 'users.NIM', '=', 'penilaian.NIM')
        ->join('pendaftarans', 'pendaftarans.NIM', '=', 'penilaian.NIM')
        ->join('bimbings', 'bimbings.NIM', '=', 'penilaian.NIM')
        ->join('permohonans', 'permohonans.NIM', '=', 'permohonans.NIM')
        ->select('penilaian.id', 'users.name', 'users.NIM', 'perusahaan', 'pendaftarans.aL', 'b1', 'b5', 'penilaian.status as status')
        ->where('penilaian.status', 0)->count();
        $mahasiswaSelesai = User::where('role_id', 1)->where('status', 'Selesai KP')->count();

        return view('koordinator.home', [
            'mahasiswa' => $mahasiswa,
            'mahasiswa1' => $mahasiswa1,
            'dataBulan' => $dataBulan,
            'mahasiswaSelesai' => $mahasiswaSelesai,
        ]);
    }
}
    
```

Gambar 4 fungsi index

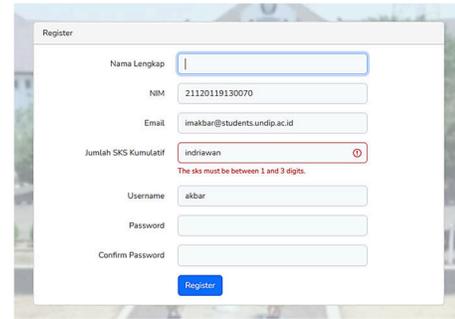
Dalam contoh gambar ini, seluruh data yang didefinisikan pada fungsi dapat digunakan dalam view 'koordinator.home' untuk menampilkan informasi terkait data-data yang dikirimkan. Kode tersebut mengambil view 'koordinator.home' dan mengirimkan data melalui array dengan kunci, kemudian view tersebut akan dirender dan ditampilkan dalam respons yang dikirimkan ke pengguna.



b. LoginController

```
protected function authenticated(Request $request, $user){
    if ($user->hasRole('admin')){
        return redirect()->route('admin');
    }
    else if ($user->hasRole('dosen')){
        return redirect()->route('dosen');
    }
    else if ($user->hasRole('koor')){
        return redirect()->route('koordinator');
    }
    else if ($user->hasRole('mahasiswa')){
        return redirect()->route('mahasiswa');
    }
    return redirect()->route('welcome');
}
```

Gambar 5 LoginController



Gambar 7 tampilan halaman registrasi

Pada loginController terdapat fungsi untuk *routing* tiap role, hal ini dilakukan untuk memetakan halaman yang berbeda untuk masing-masing role. Pada fungsi ini diberikan pengondisian ketika pengguna yang login merupakan admin, dosen, mahasiswa, atau koordinator KP. Setiap *role* akan dialihkan ke rute yang berbeda atau bisa dibilang akan dialihkan ke halaman yang berbeda. Hal ini dilakukan agar pengguna yang *login* dapat langsung beralih menuju halamannya masing-masing.

Gambar di atas merupakan tampilan halaman registrasi pengguna dan terdapat contoh keluaran yang akan terjadi ketika masukan yang dilakukan oleh pengguna tidak sesuai dengan ketentuan. Dalam kasus ini peringatan keluar ketika pengguna memasukkan nilai *string* pada *textfield* "Jumlah SKS Kumulatif" yang mana seharusnya diisi dengan 1 sampai 3 digit bilangan bulat

c. RegisterController

Pada RegisterController dibuat sebuah fungsi validator untuk memvalidasi data yang dikirimkan oleh pengguna yang melakukan registrasi. Ketika masukan yang dikirimkan oleh pengguna tidak sesuai dengan data yang seharusnya dikirimkan maka data tidak akan diproses.

2. Pembuatan Model

Model dalam laravel digunakan untuk menghubungkan dan berinteraksi dengan basis data dengan menggunakan objek-objek PHP. Dalam model, pengembang dapat mendefinisikan relasi antara tabel atau koleksi basis data, seperti relasi one to one, one to many, many to many, dan sebagainya.

```
protected function validator(array $data)
{
    return Validator::make($data, [
        'name' => ['required', 'string', 'max:255'],
        'NIM' => ['required', 'string', 'max:255', 'unique:users'],
        'email' => ['required', 'string', 'email', 'max:255', 'unique:users'],
        'sks' => ['required', 'digits_between:1,3'],
        'username' => ['required', 'string', 'max:255'],
        'password' => ['required', 'string', 'min:8', 'confirmed'],
    ]);
}

/**
 * Create a new user instance after a valid registration.
 */
protected function create(array $data)
{
    $user = User::create([
        'name' => $data['name'],
        'NIM' => $data['NIM'],
        'email' => $data['email'],
        'sks' => $data['sks'],
        'username' => $data['username'],
        'password' => Hash::make($data['password']),
        'role_id' => 1,
    ])->assignRole('mahasiswa');
    return $user;
}
```

Gambar 6 RegisterController

```
public function mhspendaftaran(){
    return $this->hasOne(Pendaftaran::class, 'NIM', 'NIM')->latest();
}
public function mhspermohonan(){
    return $this->hasOne(Permohonan::class, 'NIM', 'NIM')->latest();
}
public function mhsbimbingan(){
    return $this->hasOne(Bimbingan::class, 'NIM', 'NIM')->latest();
}
public function mhspenilaian(){
    return $this->hasOne(Penilaian::class, 'NIM', 'NIM')->latest();
}
public function mhspenjadwalan(){
    return $this->hasOne(Penjadwalan::class, 'NIM', 'NIM')->latest();
}

public function dosenpendaftaran(){
    return $this->hasMany(Pendaftaran::class, 'NIP', 'NIP')->latest();
}
public function dosenbimbingan(){
    return $this->hasMany(Bimbingan::class, 'NIP', 'NIP')->latest();
}
public function dosenpenilaian(){
    return $this->hasMany(Penilaian::class, 'NIP', 'NIP')->latest();
}
public function dosenpenjadwalan(){
    return $this->hasMany(Penjadwalan::class, 'NIP', 'NIP')->latest();
}
```

Gambar 8 Model user

Pada bagian bawah gambar terdapat fungsi yang digunakan untuk mendaftarkan pengguna dengan *role* mahasiswa. Akun akan dibuat berdasarkan data yang telah dikirimkan oleh pengguna lewat halaman registrasi, kemudian akan diberikan *role* mahasiswa. Setelah membuat akun maka pengguna akan otomatis *login* dan diharuskan untuk memverifikasi akun lewat email yang telah didaftarkan.

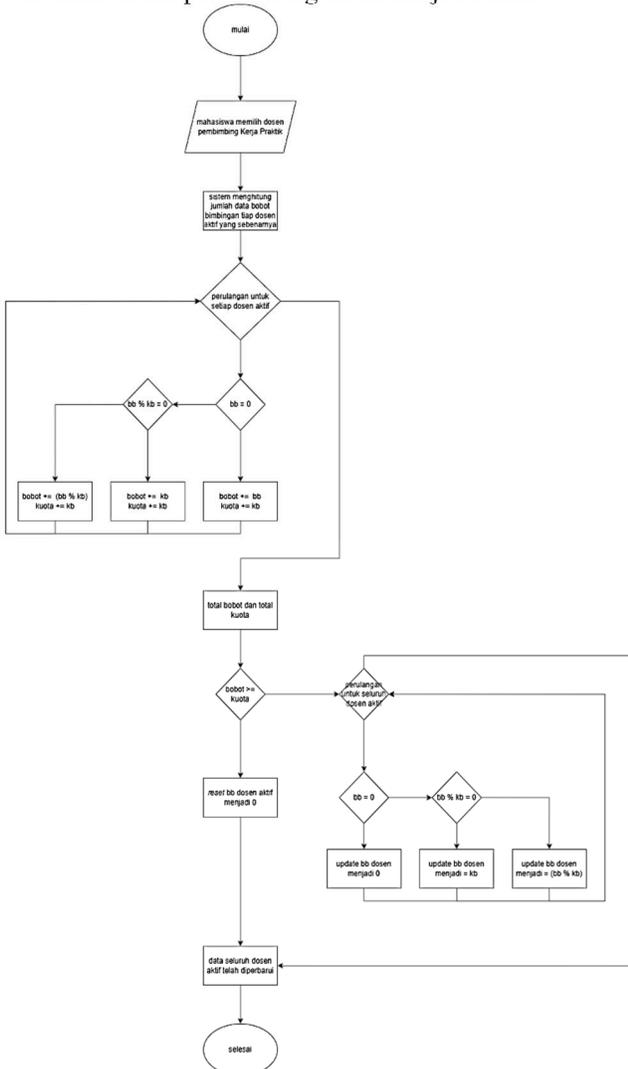
Dalam model User terdapat relasi dengan tabel-tabel lainnya. Pada gambar terdapat relasi antara tabel User dengan tabel Permohonan, Pendaftaran, Bimbingan, Penilaian, dan Penjadwalan dengan relasi *hasOne* yang berarti satu buah model memiliki sebuah record lain pada model yang berbeda dan *hasMany* yang berarti sebuah model memiliki banyak record yang terkait pada model yang berbeda melalui kunci asing (*foreign key*) yang disimpan pada tabel model tersebut. Dalam relasi



*hasMany*, setiap record pada model utama dapat memiliki banyak *record* yang terkait pada model terkait. Contohnya, jika sebuah model User memiliki banyak *record* pada model Pendaftaran, maka User memiliki *hasMany* dengan Pendaftaran. Sedangkan dalam relasi *hasOne*, setiap *record* pada model utama memiliki satu *record* yang terkait dengan model terkait. Contohnya, jika sebuah model User memiliki satu *record* pada model Pendaftaran, maka User memiliki relasi *hasOne* dengan Pendaftaran.

### 3. Balancing bobot bimbingan dosen pembimbing

Pada sistem informasi yang dibuat ini menerapkan fitur balancing bobot bimbingan dosen pembimbing Kerja Praktik. Fitur ini diterapkan untuk mencegah adanya dosen pembimbing yang kelebihan beban bimbingan lebih dari dosen-dosen lainnya. Dengan adanya fitur ini diharapkan mahasiswa dapat dipermudah dalam memilih dosen pembimbing untuk Kerja Praktik.



Gambar 9 flowchart balancing

Pada gambar 9 terdapat *flowchart* yang menjadi acuan untuk membuat logika dalam fitur *balancing* ini. Penghitungan akan dimulai ketika mahasiswa telah memilih dosen pembimbing pada halaman pendaftaran Kerja Praktik. Pada halaman Pendaftaran Kerja Praktik mahasiswa hanya dapat memilih dosen pembimbing yang berstatus 'aktif' dan memiliki kuota bimbingan yang tersedia. Setelah mahasiswa mendaftar dan memilih dosen pembimbing Kerja Praktik, maka bobot bimbingan dari dosen pembimbing tersebut akan ditambahkan.

```
// definisikan variabel
$semua = User::where('role_id', 4)->where('status', '1')->get();

foreach($semua as $bimbingan){
    $bobot = Pendaftaran::where('NIP', $bimbingan['NIP']->get());
    $jumlah = count($bobot);
    User::where('NIP', $bimbingan['NIP']->first()->update([
        'bobot_bimbingan' => $jumlah
    ]);
}
```

Gambar 10 Source code balancing 1

Variabel yang didefinisikan di atas diisi dengan data dosen yang aktif dan mengabaikan data dosen yang tidak aktif. Kemudian melakukan pembaruan bobot bimbingan dari tabel pendaftaran pada masing-masing dosen. Hal ini dilakukan untuk meminimalisir adanya kesalahan penghitungan bobot bimbingan dari tabel pendaftaran.

```
// untuk menghitung kuota bimbingan dan bobot bimbingan
$bobotbimbingan = 0;
$kuotabimbingan = 0;
$i = 0;
$semua = User::where('role_id', 4)->where('status', '1')->get();
foreach ($semua as $bimbingan) {
    if ($bimbingan['kuota_bimbingan'] == 0) {
        $bobotbimbingan += 0;
        $kuotabimbingan += 0;
    }
    else if ($bimbingan['bobot_bimbingan'] == 0) {
        $bobotbimbingan += $bimbingan['bobot_bimbingan'];
        $kuotabimbingan += $bimbingan['kuota_bimbingan'];
    }
    else if ($bimbingan['bobot_bimbingan'] % $bimbingan['kuota_bimbingan'] == 0) {
        $bobotbimbingan += $bimbingan['kuota_bimbingan'];
        $kuotabimbingan += $bimbingan['kuota_bimbingan'];
    }
    else {
        $bobotbimbingan += $bimbingan['bobot_bimbingan'] % $bimbingan['kuota_bimbingan'];
        $kuotabimbingan += $bimbingan['kuota_bimbingan'];
    }
}
```

Gambar 11 Source code balancing 2

Mendefinisikan ulang variabel untuk memperbarui nilai dari variabel, kemudian melakukan perulangan dan memasukkannya ke dalam variabel baru untuk melakukan kalkulasi kuota dan bobot. *\$bobotbimbingan* dan *\$kuotabimbingan* merupakan sebuah variabel yang digunakan untuk menyimpan nilai sementara yang dibutuhkan untuk penghitungan



```
// pengondisian untuk update bobot bimbingan dosen
if ($bobotbimbingan >= $kuotabimbingan){
    foreach ($semua as $bimbingan) {
        if ($bimbingan['role_id'] == 4 && $bimbingan['status'] == '1') {
            User::where('NIP', $bimbingan['NIP'])->first()->update([
                'bobot_bimbingan' => 0,
            ]);
        }
    }
}else{
    foreach ($semua as $bimbingan){
        if($bimbingan['kuota_bimbingan'] == 0){
            $sini = 0;
        }else{
            $sini = $bimbingan['bobot_bimbingan'] % $bimbingan['kuota_bimbingan'];
        }
        if($sini == 0 && $bimbingan['bobot_bimbingan'] == 0){
            User::where('NIP', $bimbingan['NIP'])->first()->update([
                'bobot_bimbingan' => 0,
            ]);
        }
        else if ($sini == 0){
            User::where('NIP', $bimbingan['NIP'])->first()->update([
                'bobot_bimbingan' => $bimbingan['kuota_bimbingan'],
            ]);
        }
        }else{
            User::where('NIP', $bimbingan['NIP'])->first()->update([
                'bobot_bimbingan' => $sini,
            ]);
        }
    }
}
```

Gambar 12 source code balancing 3

Ketika bobot bimbingan telah sama atau melebihi dari kuota bimbingan, maka bobot bimbingan seluruh dosen aktif akan kembali di set menjadi nol. Hal ini dilakukan untuk menyeimbangkan dosen pembimbing yang dipilih oleh mahasiswa yang melaksanakan Kerja Praktik. Kemudian jika bobot dari seorang dosen telah mencapai kuotanya, namun bobot dari dosen lain masih belum mencapai kuotanya, maka dosen yang telah mencapai kuotanya tidak akan dapat dipilih oleh mahasiswa yang akan mendaftar Kerja Praktik dan bobot bimbingannya akan diberi nilai sama dengan kuota bimbingan dosen tersebut.

#### 4. Pengujian aplikasi

Pengujian yang dilakukan merupakan *blackbox testing* atau pengujian kotak hitam. Pengujian ini dilakukan tanpa mengetahui detail implementasi dan kode program. Dalam pengujian kotak hitam ini dilakukan dari sudut pandang pengguna dimana pengujian dilakukan dengan memberikan suatu masukan dan mengevaluasi keluaran yang dihasilkan.

## V. KESIMPULAN

Dari penelitian ini ditemukan hasil kesimpulan sebagai, yaitu (1) pada pengujian kotak hitam sistem informasi kerja praktik dapat disimpulkan bahwa sistem informasi kerja praktik dapat berfungsi dengan baik dan dapat membantu pengguna untuk melaksanakan mata kuliah kerja praktik, juga (2) metode *waterfall* telah diimplementasikan dengan baik pada penelitian ini untuk menganalisis kebutuhan pengguna, merencanakan fitur-fitur aplikasi yang akan dibuat, merancang basis data sistem, hingga mengimplementasikan rancangan tersebut.

Dalam pengembangan sistem informasi kerja praktik, terdapat beberapa pengembangan yang dapat dilakukan ke depannya, di antaranya yaitu: (1) pengembang dapat melakukan validasi jumlah sks kumulatif mahasiswa dengan mengintegrasikan dengan SIAP UNDIP dan (2) pengembang dapat menambahkan fitur untuk konsultasi bimbingan mahasiswa-dosen.

## DAFTAR PUSTAKA

- [1] D. Susanti, "Pembangunan Sistem Informasi Administrasi Pelayanan Publik yang Berdampak terhadap Pelayanan Kepada Masyarakat," *Infotech J.*, vol. 4, no. 2, pp. 36–41, 2018, doi: 10.31949/inf.v4i2.1107.
- [2] M. Masjudin, C. A. Wicaksana, A. Agusutrisno, and R. Fahrizal, "Rancang Bangun Sistem Informasi Arsip dan Persuratan Program Studi Teknik Elektro Untirta Berbasis Framework Laravel," *Setrum Sist. Kendali-Tenaga-elektronika-telekomunikasi-komputer*, vol. 11, no. 1, 2022.
- [3] B. Firma Sahrul, M. A. Safi'ie, and O. D. WA, "Implementasi Sistem Informasi Akademik Berbasis Web Menggunakan Framework Laravel," *TRANSFORMASI*, vol. 12, no. 1, 2016.
- [4] R. Setiawan, Y. H. Agustin, and I. H. R. Ningsih, "Perancangan Sistem Informasi Kerja Praktik Berbasis Web," *J. Algoritma.*, vol. 19, no. 1, pp. 42–53, 2022.
- [5] A. Kasim, L. Hadjaratie, and R. H. Dai, "Rancang bangun sistem informasi skripsi dan kerja praktik berbasis web," *Jambura J. Informatics*, vol. 2, no. 2, pp. 95–107, 2020.



©2023. This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/).