

# Perancangan Sistem Pencatatan Konsultasi Bimbingan Mahasiswa-Dosen Teknik Komputer Universitas Diponegoro Menggunakan Kerangka Kerja CodeIgniter

## *Design of Student-Lecturer Consultation Recording System for Computer Engineering at Diponegoro University Using CodeIgniter Framework*

Akhmad Arief Widodo\*), Rinta Kridalukmana, Adnan Fauzi

*Program Studi Teknik Komputer, Fakultas Teknik, Universitas Diponegoro  
Jl. Prof. Soedarto, SH, Kampus Undip Tembalang, Semarang, Indonesia 50275*

---

**How to cite:** A. A. Widodo, R. Kridalukmana, dan A. Fauzi, “Perancangan Sistem Pencatatan Konsultasi Bimbingan Mahasiswa-Dosen Teknik Komputer Universitas Diponegoro Menggunakan Kerangka Kerja CodeIgniter,” *Jurnal Teknik Komputer*, vol. 1, no. 4, pp. 217-225, 2023. doi: 10.14710/jtk.v1i4.37905 [Online].

---

**Abstract** – In the field of education, student-teacher consultations are crucial in helping students achieve good academic performance. However, the process of recording consultation records often faces difficulties and delays. Therefore, an efficient and effective system is needed to record and manage consultation information. After interviewing several teachers and students, issues regarding the consultation process from both the teacher and student sides were found. To address these issues, a student-teacher consultation record keeping application was developed. The application will be built using PHP and HTML programming languages using the CodeIgniter framework and MariaDB database. The development method used is Rapid Application Development (RAD). The result of this research is an application aimed at addressing the documentation issues of student-teacher consultation processes in Computer Engineering at Diponegoro University.

**Keywords** – student-teacher consultation, recording system, PHP and HTML, CodeIgniter, Rapid Application Development (RAD), MariaDB database, Diponegoro University.

**Abstrak** – Dalam dunia pendidikan, konsultasi bimbingan antara mahasiswa dan dosen sangat penting untuk membantu mahasiswa mencapai prestasi akademis yang baik. Namun, proses pencatatan konsultasi bimbingan sering kali mengalami kesulitan dan keterlambatan. Oleh karena itu, dibutuhkan suatu sistem yang efisien dan efektif untuk mencatat dan mengelola informasi konsultasi bimbingan. Setelah melakukan wawancara terhadap beberapa dosen dan mahasiswa, ditemukan permasalahan mengenai proses bimbingan dari sisi dosen dan mahasiswa. Untuk mengatasi masalah tersebut, dibuatlah aplikasi pencatatan konsultasi bimbingan

mahasiswa-dosen berbasis web. Aplikasi ini akan dibangun menggunakan bahasa pemrograman PHP dan HTML dengan kerangka kerja CodeIgniter dan basis data MariaDB. Metode pengembangan yang digunakan adalah Rapid Application Development (RAD). Hasil dari penelitian ini adalah aplikasi yang bertujuan untuk mengatasi permasalahan terkait dokumentasi proses konsultasi bimbingan mahasiswa-dosen pada Teknik Komputer Universitas Diponegoro.

**Kata kunci** – konsultasi bimbingan mahasiswa-dosen, sistem pencatatan, PHP dan HTML, CodeIgniter, Rapid Application Development (RAD), basis data MariaDB, Universitas Diponegoro.

## I. PENDAHULUAN

Perkembangan dunia teknologi dan ilmu pengetahuan membuka peluang bagi pelajar untuk memilih jenjang pendidikan yang sesuai dengan minat dan bakat mereka. Teknik Komputer Universitas Diponegoro menerima mahasiswa baru lebih dari 100 orang setiap tahun dan memiliki 43 dosen dengan 11 dosen S3 dan 32 dosen S2. Dalam dunia pendidikan, konsultasi bimbingan antara mahasiswa dan dosen sangat penting, namun sering mengalami kesulitan dan keterlambatan dalam pencatatannya. Setelah melakukan wawancara terhadap dosen dan mahasiswa, salah satu permasalahan proses bimbingan adalah terpisahnya beberapa sistem dan kesulitan dalam komunikasi antara dosen dan mahasiswa. Solusinya adalah dengan membuat aplikasi atau perangkat lunak yang mempermudah proses bimbingan dalam satu sistem yang terintegrasi dengan fitur chat.

## II. TINJAUAN PUSTAKA

### A. Penelitian Terdahulu

Penelitian yang dilakukan oleh Dinar Ajeng Kristiani yang berjudul “Sistem Informasi Monitoring

---

\*) Akhmad Arief Widodo  
Email: [akhmadaw@student.ce.undip.ac.id](mailto:akhmadaw@student.ce.undip.ac.id)



Skripsi Berbasis Web (Studi Kasus: Prodi Akuntansi Universitas Mercu Buana)”. Metode pengembangan yang digunakan untuk membangun sistem dalam penelitian ini adalah metode RAD, dengan menggunakan bahasa pemrograman PHP dan basis data MySQL. Permasalahan yang dihadapi dalam penelitian ini adalah pengelolaan skripsi di Fakultas Ekonomi dan Bisnis Universitas Mercu Buana Program Studi Akuntansi yang masih menggunakan metode lama. Hal ini menyebabkan kesulitan dalam penyampaian informasi kepada mahasiswa, dosen dan ketua prodi akuntansi dan menyebabkan terselipnya berkas atau bahkan manipulasi. Penelitian ini menghasilkan sistem informasi monitoring skripsi berbasis web dengan tiga level akses pengguna, yaitu mahasiswa, dosen, dan administrator. Sistem yang dibangun dapat mengelola proses skripsi dari pendaftaran hingga sidang beserta yudisium [1].

Berikutnya, ada penelitian yang dilakukan oleh Mochamad Arif Kurniawan yang berjudul “Sistem Informasi Bimbingan Skripsi Menggunakan Metode Rapid Application Development Berbasis User Centered Design”. Metode pengembangan yang digunakan untuk membangun sistem dalam penelitian ini adalah metode RAD, dengan menggunakan bahasa pemrograman PHP dengan kerangka kerja CodeIgniter dan basis data MySQL. Permasalahan yang dihadapi dalam penelitian ini adalah pelaksanaan bimbingan skripsi di mana mahasiswa harus melakukan konsultasi secara reguler dengan dosen pembimbing untuk membahas progres penulisan skripsi dan menerima masukan dan koreksi. Namun, kendala muncul saat ini karena pandemi COVID-19 yang mengharuskan penerapan jarak fisik antara mahasiswa dan dosen pembimbing, sehingga menyulitkan pelaksanaan proses bimbingan skripsi. Penelitian ini menghasilkan sistem informasi bimbingan skripsi dengan empat level akses pengguna, yaitu admin, kaprodi, dosen, dan mahasiswa. Sistem yang dibangun dapat mengelola proses skripsi dari pendaftaran hingga sidang, disertai dengan fitur *live chat* menggunakan layanan Tawk.to [2].

Selanjutnya, ada pula penelitian yang dilakukan oleh Muhammad Ikhsan Fakhri yang berjudul “Perancangan dan Implementasi Sistem Informasi Bimbingan Tugas Akhir Online”. Sistem yang dibangun dalam penelitian ini menggunakan bahasa pemrograman PHP dengan kerangka kerja CodeIgniter dan basis data MySQL. Permasalahan yang dihadapi dalam penelitian ini adalah proses bimbingan tugas akhir yang dilakukan secara manual yang cukup melelahkan dan memakan waktu yang cukup banyak, karena mahasiswa harus melewati beberapa tahap dan prosedur sebelum dapat melakukan bimbingan dengan dosen pembimbing. Penelitian ini menghasilkan sistem informasi tugas akhir *online* dengan empat level akses pengguna, yaitu admin, mahasiswa, kaprodi, dan dosen pembimbing. Sistem yang dibangun dapat mengelola proses

bimbingan skripsi antara mahasiswa dan dosen, yang progresnya dapat dilihat oleh pengguna kaprodi [3].

Selanjutnya ada penelitian yang dilakukan oleh Dedi Saputra yang berjudul “Sistem Informasi Bimbingan Tugas Akhir Mahasiswa Berbasis Website Menggunakan Metode Waterfall”. Metode pengembangan yang digunakan untuk membangun sistem dalam penelitian ini adalah metode Waterfall, dengan menggunakan bahasa pemrograman PHP dengan kerangka kerja CodeIgniter dan basis data MySQL. Permasalahan yang dihadapi dalam penelitian ini adalah proses bimbingan/konsultasi yang terbatas pada masa pandemi Covid-19, karena harus memperhatikan protokol kesehatan dalam melakukan tatap muka langsung. Penelitian ini menghasilkan sistem informasi bimbingan tugas akhir mahasiswa berbasis website dengan dua level akses pengguna, yaitu mahasiswa dan dosen. Sistem yang dibangun berfokus pada proses pengajuan bimbingan tugas akhir [4].

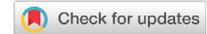
Terakhir, ada penelitian yang dilakukan oleh Bagus Dewo Anjano yang berjudul “Sistem Informasi Bimbingan Tugas Akhir di Universitas Batam Berbasis Web Menggunakan PHP dan SQL”. Sistem yang dibangun dalam penelitian ini menggunakan bahasa pemrograman PHP dan basis data MySQL. Permasalahan yang dihadapi dalam penelitian ini adalah proses bimbingan tugas akhir yang dilakukan secara tatap muka yang terkesan kurang efektif. Hal ini dikarenakan jumlah mahasiswa yang dapat bertemu dengan dosen pembimbing terbatas dan harus menentukan waktu yang tepat untuk bertemu dengan pembimbing yang berbeda lokasi, yang dapat menyebabkan peningkatan biaya operasional untuk pembelian kertas. Penelitian ini menghasilkan sistem informasi tugas akhir dengan tiga level hak akses pengguna, yaitu admin, dosen, dan mahasiswa. Sistem yang dibangun dapat digunakan mahasiswa atau dosen dalam melakukan bimbingan secara *online* tanpa perlu melakukan tatap muka secara langsung sekaligus menjaga histori data [5].

## B. PHP

PHP (PHP: *Hypertext Preprocessor*) adalah bahasa pemrograman *scripting server-side* yang bersifat *open source*. Sebagai bahasa *scripting*, PHP mengeksekusi perintah pemrograman pada saat runtime dan hasilnya akan berbeda tergantung pada data yang diproses. Karena PHP adalah bahasa pemrograman *server-side*, skrip dari PHP akan diproses di server. Beberapa jenis server yang umum digunakan bersama PHP adalah Apache, Nginx, dan LiteSpeed [6].

## C. Web Server

*Web server* adalah perangkat keras atau perangkat lunak yang digunakan untuk melayani permintaan dari klien dan mengirimkan kembali jawaban yang disebut sebagai respons [7]. *Web server* menerima permintaan



dari klien melalui peramban, seperti Google Chrome, Firefox, atau Safari. *Web server* menggunakan protokol *Hypertext Transfer Protocol* (HTTP) dan *Hypertext Transfer Protocol Secure* (HTTPS) untuk memproses permintaan dari klien [8]. Beberapa contoh *web server* yang populer di antaranya: Apache, Nginx, IIS (*Internet Information Services*), Lighttpd, dan Caddy.

#### D. Database Management System

*Database Management System* (DBMS) adalah sebuah program komputer yang melakukan tugas-tugas manajemen data seperti penyisipan (*insert*), perubahan/modifikasi (*update*), penghapusan (*delete*), dan pengambilan data/informasi (*select*) untuk tujuan tertentu. Contoh DBMS yang sering digunakan adalah MySQL dan MariaDB. MySQL adalah perangkat lunak basis data *open source* yang sangat populer di dunia. MySQL menjadi pilihan utama bagi banyak pengembang perangkat lunak dan aplikasi karena sintak yang mudah dipahami, didukung oleh berbagai bahasa seperti C, C++, Java, PHP, Python. Sedangkan MariaDB mirip dengan MySQL, karena MariaDB adalah versi pengembangan terbuka dan mandiri dari MySQL. Setelah MySQL diakuisisi oleh Oracle pada September 2010, Monty Program sebagai penulis awal kode sumber MySQL membuat versi yang lebih mandiri yaitu MariaDB. Banyak perusahaan dan situs besar telah melakukan migrasi dari MySQL ke MariaDB, seperti Google dan Wikipedia. Kelebihan MariaDB antara lain performa yang bagus dan tidak berat, serta kompatibilitas yang baik dengan MySQL. MariaDB juga kompatibel dengan berbagai platform seperti LINUX, Windows, MacOS, FreeBSD, Solaris [9].

#### E. HTML

HTML (*Hypertext Markup Language*) adalah bahasa *markup* yang digunakan untuk membuat halaman web [10]. HTML digunakan sebagai bahasa untuk menstrukturkan informasi menggunakan tanda-tanda atau *tag*, sehingga peramban web dapat menafsirkannya dan membuat presentasi visual seperti teks, gambar, tautan, dan elemen lain bagi pengguna untuk dilihat [11]. Contoh *tag* HTML yang umum digunakan adalah `<p>` untuk paragraf, `<img>` untuk gambar, dan `<a>` untuk tautan.

#### F. Javascript

JavaScript adalah bahasa pemrograman yang memungkinkan para pengembang untuk berinteraksi dengan fungsi yang diberikan oleh peramban web. Lebih spesifik, JavaScript adalah bahasa *scripting*, yang berarti (a) secara tradisional, kode sumber JavaScript diinterpretasikan saat *runtime* dan tidak dikompilasi menjadi *byte code* dan secara praktis, tujuan utamanya adalah untuk memodifikasi perilaku aplikasi lain yang biasanya ditulis dalam bahasa

pemrograman yang berbeda, di mana diterjemahkan dan dijalankan dalam waktu nyata [12].

#### G. CodeIgniter

CodeIgniter adalah kerangka kerja pengembangan aplikasi atau *toolkit* untuk orang yang membuat situs web menggunakan PHP. Tujuannya adalah untuk memungkinkan *programmer* untuk mengembangkan proyek jauh lebih cepat dibandingkan jika menulis kode dari awal, dengan menyediakan beragam pustaka untuk tugas yang sering dibutuhkan, serta antarmuka yang sederhana dan struktur logis untuk mengakses perpustakaan tersebut. CodeIgniter memungkinkan *programmer* untuk fokus kreatif pada proyek dengan meminimalkan jumlah kode yang dibutuhkan untuk suatu tugas tertentu [13]. CodeIgniter sendiri menggunakan konsep MVC (*Model View Controller*) dalam pengembangan aplikasinya. MVC adalah teknik perancangan yang membagi kode aplikasi ke dalam tiga *layer* yaitu *model* (data), *view* (tampilan), dan *controller* (cara proses) [14].

#### H. Rapid Application Development

*Rapid Application Development* (RAD) adalah metode pengembangan perangkat lunak yang menggunakan pendekatan berorientasi objek (*object oriented approach*) untuk mempercepat proses pengembangan sistem. Tujuan dari metode ini adalah untuk mengurangi waktu yang diperlukan dalam perencanaan, perancangan, dan implementasi sistem dibandingkan dengan metode pengembangan tradisional [15]. Metode RAD mempunyai 3 tahapan utama, yaitu rencana kebutuhan (*requirement planning*), proses desain (*design workshop*), dan implementasi (*implementation*).

#### I. Black-Box Testing

*Black-box testing* adalah metode pengujian yang digunakan untuk mengevaluasi sistem atau komponen tanpa memperhatikan bagaimana sistem atau komponen tersebut dibuat atau bagaimana ia bekerja. Penguji hanya memperhatikan *input* yang diberikan ke sistem atau komponen dan *output* yang dihasilkan, tanpa memperhatikan bagaimana proses internal sistem atau komponen tersebut [16].

### III. ANALISIS DAN PERANCANGAN

#### A. Rencana Kebutuhan

Pada tahap perencanaan, pengguna dan analis bekerja sama untuk menentukan tujuan dan kebutuhan informasi dari aplikasi atau sistem yang akan dikembangkan. Fokus utama pada tahap ini adalah pemecahan masalah yang dihadapi oleh pengguna dan bagaimana sistem dapat membantu menyelesaikan masalah tersebut [17].



## 1. Kebutuhan Fungsional

Pada tahap perencanaan, pengguna dan analis bekerja sama untuk menentukan tujuan dan kebutuhan informasi dari aplikasi atau sistem yang akan dikembangkan. Fokus utama pada tahap ini adalah pemecahan masalah yang dihadapi oleh pengguna dan bagaimana sistem dapat membantu menyelesaikan masalah tersebut. Kebutuhan fungsional adalah kebutuhan akan fitur dan fungsi yang dibutuhkan oleh sistem, serta aktivitas atau proses yang harus dilakukan oleh sistem dalam mencapai tujuan yang diharapkan. Kebutuhan fungsional meliputi fungsi-fungsi yang harus dapat dilakukan oleh sistem dan spesifikasi yang diperlukan untuk menunjang aktivitas tersebut [26]. Aplikasi yang akan dibangun ini terdapat tiga level hak akses, yaitu admin, mahasiswa, dan dosen.

Berikut merupakan kebutuhan fungsional aplikasi pada sisi admin.

1. Admin dapat masuk ke dalam sistem.
2. Admin dapat membuat, melihat, mengubah, dan menghapus pengguna pada sistem.
3. Admin dapat melihat dan menyunting pengajuan bimbingan.
4. Admin dapat melihat dan menyunting bimbingan berjalan.
5. Admin dapat melihat daftar bimbingan selesai.
6. Admin dapat melihat detail bimbingan.
7. Admin dapat melihat dokumen final *capstone*.
8. Admin dapat melihat dokumen final tugas akhir.
9. Admin dapat melihat dan mengubah data profil.
10. Admin dapat mengubah kata sandi

Berikut merupakan kebutuhan fungsional aplikasi pada sisi mahasiswa.

1. Mahasiswa dapat masuk ke dalam sistem.
2. Mahasiswa dapat melihat dan mengirim *chat* dengan dosen dan mahasiswa lain dalam satu bimbingan yang sama.
3. Mahasiswa dapat membuat, melihat, menyunting, dan menghapus pengajuan bimbingan.
4. Mahasiswa dapat melihat dan menyunting bimbingan berjalan.
5. Mahasiswa dapat melihat daftar bimbingan selesai.
6. Mahasiswa dapat melihat detail bimbingan.
7. Mahasiswa dapat melihat, mengunggah, menyunting, dan menghapus dokumen bimbingan.
8. Mahasiswa dapat melihat, mengunggah, menyunting, dan menghapus dokumen final bimbingan.
9. Mahasiswa dapat melihat dan menyunting data profil.
10. Mahasiswa dapat mengubah kata sandi.

Berikut merupakan kebutuhan fungsional aplikasi pada sisi dosen.

1. Dosen dapat masuk ke dalam sistem.

2. Dosen dapat melihat dan mengirim *chat* dengan mahasiswa dan dosen lain dalam satu bimbingan yang sama.
3. Dosen dapat melihat, menyetujui, dan menolak pengajuan bimbingan.
4. Dosen dapat melihat, menyelesaikan, dan membatalkan persetujuan bimbingan berjalan.
5. Dosen dapat melihat dan membatalkan status selesai pada bimbingan selesai.
6. Dosen dapat melihat detail bimbingan.
7. Dosen dapat melihat, merevisi, dan menyetujui dokumen bimbingan.
8. Dosen dapat melihat dokumen final bimbingan.
9. Dosen dapat melihat dan menyunting data profil.
10. Dosen dapat mengubah kata sandi.

## 2. Kebutuhan Non-Fungsional

Kebutuhan non-fungsional merupakan kebutuhan yang menggambarkan bagaimana sistem bekerja ke depannya [18].

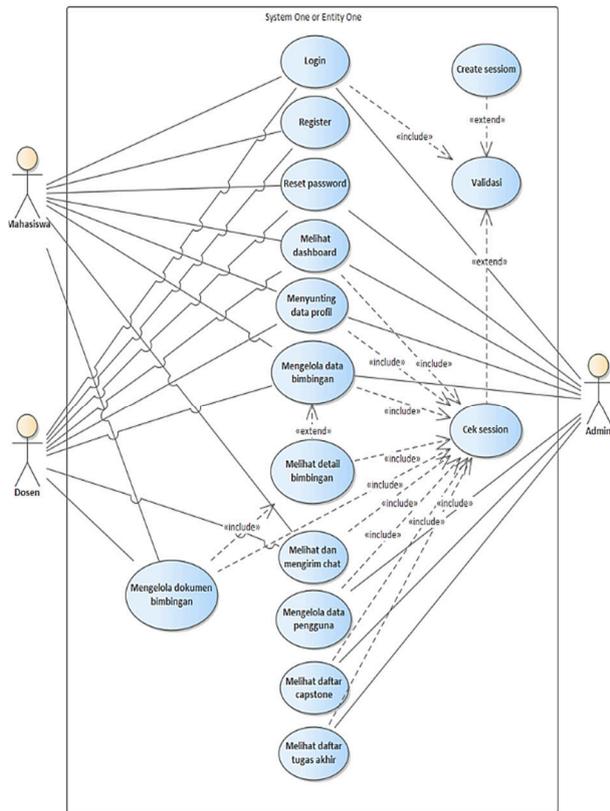
- a. Perangkat keras
  1. Prosesor Pentium® 4 2.0 GHz atau yang lebih tinggi.
  2. RAM 1 GB atau yang lebih tinggi.
  3. Harddisk drive atau SSD (*solid-state drive*) dengan ruang kosong sebesar 10 GB atau yang lebih tinggi.
  4. Monitor SVGA dengan resolusi 1366x768 atau yang lebih tinggi.
  5. Keyboard yang kompatibel.
  6. Mouse atau *touchpad* dan *keyboard* yang kompatibel.
- b. Perangkat lunak
  1. Sistem operasi Windows 11.
  2. XAMPP versi 3.3.0 yang di dalamnya sudah termasuk PHP 7.4.26 sebagai bahasa pemrograman, Apache2 2.4.51 sebagai web server, MariaDB 10.4.22 sebagai penyimpanan basis data atau *Database Management System* (DBMS), dan PHPMyAdmin 5.1.1 sebagai manajemen basis data.
  3. Visual Studio Code sebagai teks editor untuk *coding*.
  4. Microsoft Edge dan Google Chrome sebagai peramban

## B. Proses Desain

### 1. Diagram *use case*

*Use case* adalah sebuah metode untuk menggambarkan fungsi dari sebuah sistem dari perspektif atau sudut pandang pengguna sistem. *Use case* digunakan untuk menjelaskan tindakan yang dilakukan pengguna terhadap sistem atau tindakan yang dilakukan oleh sistem terhadap pengguna. Diagram *use case* digunakan untuk menggambarkan interaksi antara pengguna dan sistem [19]. Berikut

merupakan diagram *use case* yang digunakan dalam penelitian ini yang ditunjukkan pada Gambar 1.



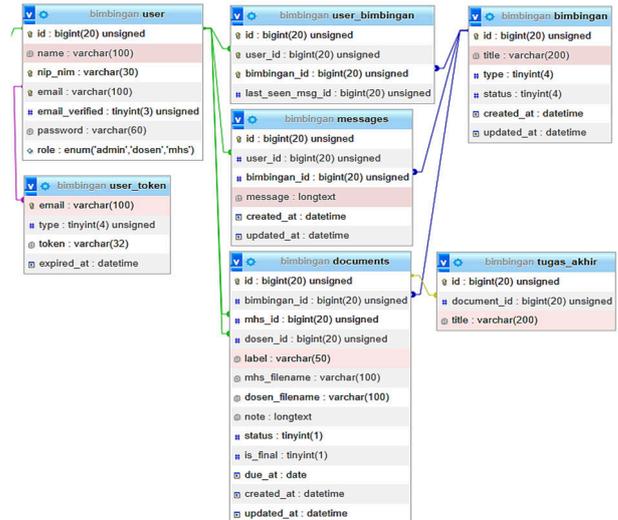
Gambar 1. Diagram *use case* aplikasi

Pada diagram *use case* tersebut, dimodelkan beberapa buah *case* yang menunjukkan interaksi antara pengguna dengan sistem. Terdapat tiga buah aktor dalam sistem, yaitu mahasiswa, dosen, dan admin. Keterangan *extend* menunjukkan bahwa *case* tersebut merupakan tambahan fungsional *case* yang ditunjuk tanda panah. Sedangkan keterangan *include* menunjukkan bahwa sebelum melakukan *case* yang ditunjuk anak panah, harus sudah menjalankan *case* tersebut. Dalam sistem yang dirancang, pengguna harus melakukan aktivitas *cek session* terlebih dahulu sebelum melakukan *case* apa pun, baru kemudian dapat menjalankan *case* sesuai dengan hak akses aktor yang digunakan untuk *login*.

Dalam sistem yang dirancang, aktor mahasiswa dan dosen dapat melakukan aktivitas *login*, *register*, *reset password*, melihat *dashboard*, menyunting data profil, mengelola data bimbingan, melihat detail bimbingan, mengelola dokumen bimbingan, serta melihat dan mengirim *chat*. Sedangkan untuk aktor admin dapat melakukan aktivitas *login*, *reset password*, melihat *dashboard*, menyunting data profil, mengelola data bimbingan, melihat detail bimbingan, mengelola data pengguna, melihat daftar *capstone*, melihat daftar tugas akhir.

## 2. Perancangan basis data

Skema basis data dari sistem yang dibuat ditunjukkan pada Gambar 2.



Gambar 2. Skema basis data aplikasi

## C. Implementasi

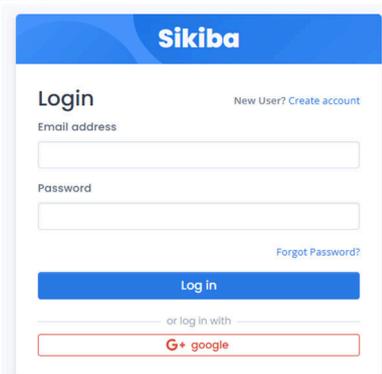
Setelah desain sistem disetujui oleh pengguna dan analis, tahap selanjutnya adalah pengembangan. *Programmer* akan mengimplementasikan desain menjadi suatu program yang dapat dijalankan. Setelah pengembangan selesai, baik itu sebagian atau keseluruhan, program tersebut akan diuji untuk menemukan kesalahan atau *bug* sebelum diterapkan pada organisasi. Pada tahap ini, pengguna juga dapat memberikan tanggapan dan persetujuan terhadap sistem yang dikembangkan.

## IV. HASIL DAN PEMBAHASAN

### A. Kajian Hasil Penelitian

#### 1. Halaman *login*

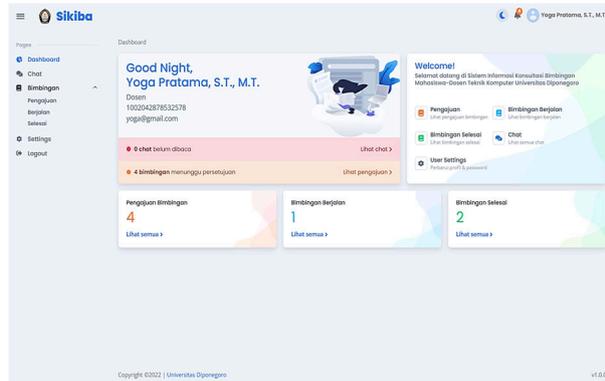
Halaman *login* merupakan halaman yang pertama kali dibuka ketika pengguna baru pertama kali membuka aplikasi. Pada halaman *login* ini, akan dilakukan pengecekan *session* terlebih dahulu. Apabila sebelumnya pengguna masih memiliki *session login*, maka akan diarahkan ke halaman *dashboard* sesuai *role*. Sedangkan apabila pengguna tidak memiliki *session login*, barulah akan ditampilkan halaman *login*. Tampilan halaman *login* dapat dilihat pada Gambar 3.



Gambar 3. Tampilan halaman *login*

## 2. Halaman dashboard

Halaman *dashboard* merupakan halaman yang pertama kali dibuka setelah pengguna *login* ke sistem. Halaman *dashboard* menampilkan konten dan menu *sidebar* sesuai dengan *role* pengguna, yang terdiri antara lain admin, mahasiswa, dan dosen. Tampilan halaman *dashboard* dosen dapat dilihat pada Gambar 4.



Gambar 4. Tampilan halaman *dashboard*

## 3. Halaman pengajuan bimbingan

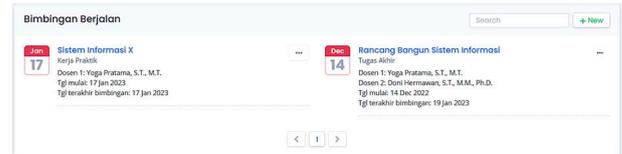
Halaman pengajuan bimbingan menampilkan judul bimbingan yang baru diajukan oleh mahasiswa. Halaman pengajuan bimbingan dapat diakses oleh pengguna admin, mahasiswa, dan dosen. Pada halaman pengajuan bimbingan, pengguna admin hanya dapat melihat daftar pengajuan dan menyuntingnya saja. Sedangkan untuk pengguna mahasiswa, dapat membuat pengajuan bimbingan, menyuntingnya, dan menghapusnya. Tampilan halaman pengajuan bimbingan bagi mahasiswa dapat dilihat pada Gambar 5.



Gambar 5. Tampilan halaman pengajuan bimbingan

## 4. Halaman bimbingan berjalan

Halaman bimbingan berjalan menampilkan pengajuan bimbingan yang sebelumnya dibuat oleh pengguna mahasiswa dan disetujui oleh pengguna dosen. Halaman bimbingan berjalan dapat diakses oleh pengguna admin, mahasiswa, dan dosen. Pengguna admin dan mahasiswa hanya dapat melihat daftar bimbingan berjalan, menyuntingnya, dan memilih menu detail bimbingan. Sedangkan pengguna dosen dapat melihat daftar bimbingan berjalan, memilih menu detail bimbingan, menandai bimbingan sebagai selesai, dan membatalkan persetujuan bimbingan. Tampilan halaman bimbingan berjalan bagi mahasiswa yang dapat dilihat pada Gambar 6.



Gambar 6. Tampilan halaman bimbingan berjalan

## 5. Halaman bimbingan selesai

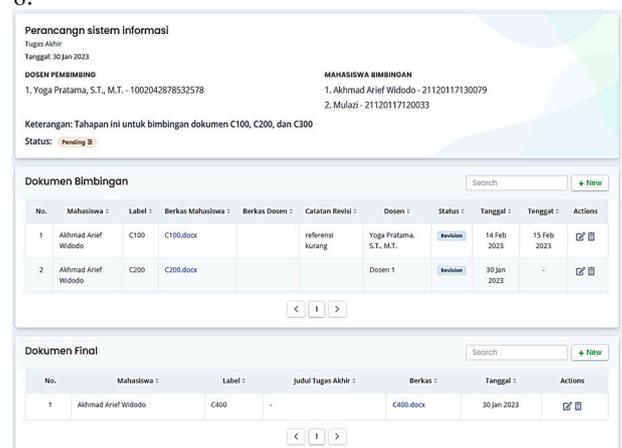
Halaman bimbingan selesai menampilkan bimbingan berjalan yang ditandai sebagai selesai oleh pengguna dosen. Halaman bimbingan selesai dapat diakses oleh pengguna admin, mahasiswa, dan dosen. Pengguna admin dan mahasiswa hanya dapat melihat daftar bimbingan berjalan, menyuntingnya, dan memilih menu detail bimbingan. Sedangkan pengguna dosen dapat melihat daftar bimbingan selesai, memilih menu detail bimbingan, dan membatalkan penyelesaian bimbingan. Berikut merupakan tampilan halaman bimbingan berjalan yang dapat dilihat pada Gambar 7



Gambar 7. Tampilan halaman bimbingan selesai

## 6. Halaman detail bimbingan

Halaman detail bimbingan menampilkan detail dari bimbingan yang dipilih dan dapat diakses oleh pengguna admin, dosen, dan mahasiswa. Pada halaman detail bimbingan, mahasiswa dapat melihat dokumen yang diunggah sendiri maupun mahasiswa lain dalam bimbingan tersebut, mengunggah, merevisi bimbingan, dan menghapus dokumen bimbingan maupun dokumen final. Sedangkan dosen dapat melihat dokumen bimbingan dan dokumen final yang diunggah oleh mahasiswa pada bimbingan tersebut, menyetujui dokumen bimbingan, dan merevisi dokumen bimbingan. Tampilan halaman detail bimbingan bagi mahasiswa dapat dilihat pada Gambar 8.

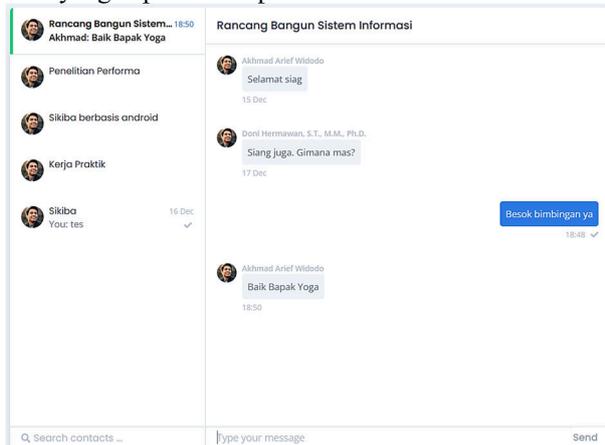


Gambar 8. Tampilan halaman detail bimbingan



## 7. Halaman chat

Halaman *chat* digunakan untuk menampilkan daftar grup *chat* dan pesan yang ada di dalamnya, serta mengirim pesan pada grup *chat* yang dipilih. Halaman *chat* dapat diakses oleh pengguna dosen dan mahasiswa. Berikut merupakan tampilan halaman *chat* yang dapat dilihat pada Gambar 9.



Gambar 9. Tampilan halaman *chat*

## B. Hasil Pengujian Aplikasi

Tahap pengujian ini akan menguji sistem aplikasi sebelum program aplikasi dipublikasikan. Dengan pengujian perangkat lunak diharapkan dapat meminimalkan kesalahan dan cacat pada sebuah perangkat lunak dan sebagai pengukuran kualitas dari perangkat lunak tersebut. Pengujian pada sistem ini menggunakan metode *black-box testing* agar mengetahui apakah sistem yang berjalan sudah sesuai analisis dan rancangan saat pengembangan. Hasil pengujian aplikasi dengan metode *black-box testing* disajikan dalam 1.

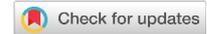
Tabel 1. Hasil pengujian aplikasi dengan metode *black-box testing*

| No. | Deskripsi pengujian  | Hasil pengujian |
|-----|--|-----------------|
| 1.  | Pengujian fungsi <i>login</i>                                | Berhasil        |
| 2.  | Pengujian fungsi <i>login</i> dengan Google                  | Berhasil        |
| 3.  | Pengujian fungsi <i>logout</i>                               | Berhasil        |
| 4.  | Pengujian fungsi register                                    | Berhasil        |
| 5.  | Pengujian fungsi lupa kata sandi                             | Berhasil        |
| 6.  | Pengujian fungsi melihat daftar pengguna dosen dan mahasiswa | Berhasil        |
| 7.  | Pengujian fungsi menambah pengguna baru                      | Berhasil        |
| 8.  | Pengujian fungsi menyunting data pengguna                    | Berhasil        |
| 9.  | Pengujian fungsi menghapus pengguna                          | Berhasil        |
| 10. | Pengujian melihat daftar pengajuan bimbingan,                | Berhasil        |

|     |   |          |
|-----|---|----------|
|     | bimbingan berjalan, dan bimbingan selesai                                     |          |
| 11. | Pengujian fungsi membuat pengajuan bimbingan                                  | Berhasil |
| 12. | Pengujian fungsi menyunting pengajuan bimbingan                               | Berhasil |
| 13. | Pengujian fungsi menghapus pengajuan bimbingan                                | Berhasil |
| 14. | Pengujian fungsi menolak dan menyetujui bimbingan                             | Berhasil |
| 15. | Pengujian fungsi menyelesaikan dan membatalkan persetujuan bimbingan          | Berhasil |
| 16. | Pengujian fungsi membatalkan penyelesaian bimbingan                           | Berhasil |
| 17. | Pengujian fungsi mengunggah dokumen bimbingan dan dokumen final               | Berhasil |
| 18. | Pengujian fungsi merevisi dokumen bimbingan atau dokumen final bagi mahasiswa | Berhasil |
| 19. | Pengujian fungsi hapus dokumen bimbingan atau dokumen final                   | Berhasil |
| 20. | Pengujian fungsi merevisi dokumen bimbingan bagi dosen                        | Berhasil |
| 21. | Pengujian fungsi menyetujui dokumen bimbingan                                 | Berhasil |
| 22. | Pengujian fungsi membaca dan mengirim <i>chat</i>                             | Berhasil |
| 23. | Pengujian fungsi perbarui data profil pengguna                                | Berhasil |
| 24. | Pengujian fungsi mengubah kata sandi  | Berhasil |

## C. Pembahasan

Sistem Pencatatan Konsultasi Bimbingan Mahasiswa-Dosen Teknik Komputer Universitas Diponegoro yang dikembangkan menggunakan kerangka kerja Codeigniter merupakan sistem informasi yang digunakan untuk mengelola bimbingan pada program studi Teknik Komputer Universitas Diponegoro sehingga dapat memudahkan proses bimbingan secara daring. Sistem dapat diakses oleh pengguna yang dibagi menjadi tiga level hak akses, antara lain admin, dosen, dan mahasiswa. Pengguna admin dapat mengelola data pengguna dosen dan mahasiswa serta melihat dokumen final bimbingan dari dosen dan mahasiswa. Sedangkan pengguna dosen dan mahasiswa dapat mengelola bimbingan dimulai dari pengajuan hingga selesai dan juga dapat melakukan *chat* di dalam sistem.



Sistem dimulai dengan pengecekan *session login* terlebih dahulu. Apabila tidak terdapat *session login* yang valid, pengguna akan diarahkan ke halaman *login*. Sedangkan setelah melakukan proses *login*, pengguna baru akan mendapatkan *session login* yang dapat digunakan untuk mengakses fungsi-fungsi yang terdapat dalam sistem sesuai dengan hak akses pengguna. Setelah berhasil masuk ke dalam sistem, pengguna dapat mengakses fungsi-fungsi dari halaman *dashboard* dan menu *sidebar*.

Pengguna admin dapat mengelola data pengguna admin dan mahasiswa meliputi menambah pengguna baru, menyunting data pengguna, dan menghapus data pengguna. Pengguna mahasiswa dapat mengelola data bimbingan meliputi membuat pengajuan bimbingan, menyunting pengajuan bimbingan, dan menghapus pengajuan bimbingan, serta dapat mengelola detail bimbingan meliputi mengunggah dokumen bimbingan, merevisi dokumen bimbingan, mengunggah dokumen final, merevisi dokumen final, dan menghapus dokumen bimbingan maupun dokumen final. Pengguna dosen juga dapat mengelola bimbingan meliputi menyetujui pengajuan, menolak pengajuan, membatalkan persetujuan bimbingan, menyelesaikan bimbingan berjalan, dan membatalkan penyelesaian bimbingan, serta mengelola detail bimbingan meliputi menyetujui dokumen bimbingan dan merevisi dokumen bimbingan. Pengguna dosen dan mahasiswa juga dapat mengakses fungsi *chat* yang dibuat grup dengan anggota sesuai dengan bimbingan yang dibuat, seperti fungsi membaca dan mengirim pesan pada grup *chat* terpilih.

Semua pengguna dapat mengakses fungsi memperbarui profil, mengubah kata sandi, dan *logout*. Pengguna dapat keluar dari sistem dengan cara melakukan *logout*, kemudian *session login* dari pengguna akan dihapus dan diakhiri sehingga memerlukan *login* kembali untuk mengakses sistem. Sistem juga akan otomatis mengakhiri *session login* apabila tidak terdapat aktivitas selama tiga jam.

## V. KESIMPULAN DAN SARAN

### A. Kesimpulan

Berdasarkan penelitian yang telah dilakukan pada “Sistem Pencatatan Konsultasi Bimbingan Mahasiswa-Dosen Teknik Komputer Universitas Diponegoro Menggunakan Kerangka Kerja Codeigniter” diperoleh kesimpulan sebagai berikut.

1. Masalah terkait dokumentasi proses konsultasi bimbingan mahasiswa-dosen di Teknik Komputer Universitas Diponegoro berhasil teratasi dengan dirancang dan dibangunnya Sistem Pencatatan Konsultasi Bimbingan Mahasiswa-Dosen Teknik Komputer Universitas Diponegoro Menggunakan Kerangka Kerja Codeigniter.
2. Dalam perancangannya, Sistem Pencatatan Konsultasi Bimbingan Mahasiswa-Dosen Teknik Komputer Universitas Diponegoro menggunakan

metode pengembangan perangkat lunak *Rapid Application Development* (RAD).

3. Sistem Pencatatan Konsultasi Bimbingan Mahasiswa-Dosen Teknik Komputer Universitas Diponegoro telah berhasil diuji secara fungsional dengan metode pengujian perangkat lunak secara *black-box testing*.

### B. Saran

Berdasarkan penelitian yang telah dilakukan pada “Sistem Pencatatan Konsultasi Bimbingan Mahasiswa-Dosen Teknik Komputer Universitas Diponegoro Menggunakan Kerangka Kerja Codeigniter” didapatkan saran sebagai berikut.

1. Sistem dapat dikembangkan ke dalam aplikasi versi *mobile* pada Android ataupun iOS sehingga dapat lebih mudah diakses melalui telepon genggam.
2. Integrasi sistem dengan *Single Sign On* (SSO) Universitas Diponegoro sehingga proses registrasi dan *login* sistem menjadi lebih mudah melalui satu akun SSO Universitas Diponegoro

## DAFTAR PUSTAKA

- [1] D. A. Kristiyanti dan A. Mulyana, “Sistem Informasi Monitoring Skripsi Berbasis Web (Studi Kasus: Prodi Akuntansi Universitas Mercu Buana),” *Jurnal Sistem Informasi Bisnis*, vol. 10, no. 1, hlm. 56–63, Jun 2020, doi: 10.21456/vol10iss1pp56-63.
- [2] M. A. Kurniawan, I. Fitri, dan D. Hidayatullah, “Sistem Informasi Bimbingan Skripsi Menggunakan Metode Rapid Application Development Berbasis User Centered Design,” *Jurnal Media Informatika Budidarma*, vol. 5, no. 3, hlm. 838, Jul 2021, doi: 10.30865/mib.v5i3.3068.
- [3] M. I. Fakhri dan V. I. Delianti, “Perancangan dan Implementasi Sistem Informasi Bimbingan Tugas Akhir Online,” *Jurnal Vocational Teknik Elektronika dan Informatika*, vol. 9, no. 1, hlm. 103–115, Mar 2021, doi: 10.24036/voteteknika.v9i1.111205.
- [4] D. Saputra, H. Haryani, A. Surniadari, M. Martias, dan F. Akbar, “Sistem Informasi Bimbingan Tugas Akhir Mahasiswa Berbasis Website Menggunakan Metode Waterfall,” *MATRIK : Jurnal Manajemen, Teknik Informatika dan Rekayasa Komputer*, vol. 21, no. 2, hlm. 403–416, Mar 2022, doi: 10.30812/matrik.v21i2.1591.
- [5] B. D. Anjano dan Nurhatisyah, “Sistem Informasi Bimbingan Tugas Akhir di Universitas Batam Berbasis Web Menggunakan PHP dan SQL,” *Zona Komputer*, vol. 10, no. 1, hlm. 27–35, Apr 2020, doi: 10.37776/zk.v10i1.490.
- [6] Muqorobin dan N. A. R. Rais, “Comparison of PHP Programming Language with Codeigniter



- Framework in Project CRUD,” *International Journal of Computer and Information System (IJCIS)*, vol. 3, no. 3, hlm. 94–98, Agu 2022, doi: 10.29040/ijcis.v3i3.77.
- [7] A. Y. Chandra, “Analisis Performansi Antara Apache & Nginx Web Server Dalam Menangani Client Request,” *Jurnal Sistem dan Informatika (JSI)*, vol. 14, no. 1, hlm. 48–56, Nov 2019, doi: 10.30864/jsi.v14i1.248.
- [8] G. A. Jaafar, S. M. Abdullah, dan S. Ismail, “Review of Recent Detection Methods for HTTP DDoS Attack,” *Journal of Computer Networks and Communications*, vol. 2019, 2019, doi: 10.1155/2019/1283472.
- [9] I. Warman dan R. Ramdaniyasyah, “Analisis Perbandingan Kinerja Query Database Management System (DBMS) Antara MySQL 5.7.16 dan MariaDB 10.1,” *JURNAL TEKNOIF*, vol. 6, no. 1, hlm. 32–41, Apr 2018, doi: 10.21063/jtif.2018.v6.1.32-41.
- [10] T. Rahmasari, “Perancangan Sistem Informasi Akuntansi Persediaan Barang Dagang Pada Toserba Selamat Menggunakan Php Dan Mysql,” *aisthebest*, vol. 4, no. 1, hlm. 411–425, Jun 2019, doi: 10.34010/AISTHEBEST.V4I1.1830.
- [11] N. Batalas, V.-J. Khan, dan P. Markopoulos, “Executable HTML,” *SoftwareX*, vol. 14, hlm. 1–6, Jun 2021, doi: 10.1016/j.softx.2021.100691.
- [12] K. J. Theisen, “Programming languages in chemistry: A review of HTML5/JavaScript,” *J Cheminform*, vol. 11, no. 11, hlm. 1–19, Feb 2019, doi: 10.1186/s13321-019-0331-1.
- [13] “Welcome to CodeIgniter4 — CodeIgniter 4.3.0 documentation,” 22 Desember 2022. [https://codeigniter.com/user\\_guide/intro/index.html](https://codeigniter.com/user_guide/intro/index.html) (diakses 5 Januari 2023).
- [14] R. Kusuma Atmaja dan I. Komarudin, “Konsep MVC Pada Rancang Bangun Aplikasi Sistem Informasi Rekrutmen Karyawan Berbasis Web,” *IMTechno: Journal of Industrial Management and Technology*, vol. 2, no. 1, Jan 2021, doi: 10.31294/imtechno.v2i1.163.
- [15] D. Setiawan Putra dan A. Fauziah, “Perancangan Aplikasi Presensi Dosen Realtime Dengan Metode Rapid Application Development (RAD) Menggunakan Fingerprint Berbasis Web,” *Jurnal Informatika: Jurnal Pengembangan IT (JPIT)*, vol. 03, no. 02, hlm. 167–171, Mei 2018, doi: 10.30591/jpit.v3i2.836.
- [16] P. C. Jorgensen dan B. DeVries, *Software Testing: A Craftsman’s Approach*, 5 ed. New York: Auerbach Publications, 2021.
- [17] M. P. Putri dan H. Effendi, “Implementasi Metode Rapid Application Development Pada Website Service Guide ‘Waterfall Tour South Sumatera,’” *Jurnal SISFOKOM*, vol. 07, no. 02, hlm. 130–136, Sep 2018.
- [18] A. Aulia Aziiza dan A. N. Fadhilah, “Analisis Metode Identifikasi dan Verifikasi Kebutuhan Non Fungsional,” *Applied Technology and Computing Science Journal*, vol. 3, no. 1, Jun 2020, doi: 10.33086/atcsj.v3i1.1623.
- [19] L. Setiyani, “Desain Sistem: Use Case Diagram,” 2021.



©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/).