

PENGEMBANGAN APLIKASI SISTEM INFORMASI RESERVASI RUANGAN BERBASIS WEBSITE

Febrian Ardi Pangestu^{1*)}, Aghus Sofwan² dan Yuli Christyono³

¹²³Program Studi Sarjana Departemen Teknik Elektro, Universitas Diponegoro
Jl. Prof. Sudharto, SH, Kampus UNDIP Tembalang, Semarang 50275, Indonesia

^{*)} E-mail: febrianap128@gmail.com

Abstrak

Pengelolaan ketersediaan ruangan di lingkungan pendidikan merupakan aspek vital dalam mendukung berbagai kegiatan akademik dan administratif. Namun, seringkali proses reservasi ruangan masih dilakukan secara manual, menyebabkan ketidakefisienan dan ketidakakuratan informasi. Berdasarkan latar belakang tersebut, Sistem Informasi Reservasi Ruangan (SISERU) dirancang sebagai solusi yang mengintegrasikan teknologi informasi untuk mengelola reservasi ruangan dengan lebih efisien dan efektif. Perancangan sistem ini menggunakan framework Codeigniter 3 dengan API yang dapat memungkinkan integrasi fitur antara dua aplikasi agar bisa berinteraksi. SISERU memiliki fitur antara lain, melakukan pencarian ruangan berdasarkan tanggal dan waktu, pemesanan online, pembatalan pemesanan, dan notifikasi status pemesanan. Pengujian aplikasi dilakukan dengan metode blackbox, pengujian performa, dan pengujian user acceptance test (UAT) guna mengetahui keandalan dari aplikasi. Berdasarkan hasil pengujian, aplikasi berhasil menyelesaikan semua skenario yang telah ditentukan dengan performa pemakaian stabil di respon 0,0086 hingga 0,2781 detik dan pemakaian memori 0,62 hingga 0,67 MB. Selain itu aplikasi juga diterima dengan baik oleh pengguna berdasarkan hasil pengujian metode UAT. SISERU diharapkan dapat membantu dalam optimalisasi penggunaan ruangan di Universitas Diponegoro.

Kata kunci : SISERU, Sistem Informasi Reservasi Ruangan, Codeigniter 3, Website

Abstract

The management of room availability in an educational environment is a vital aspect in supporting various academic and administrative activities. However, often the room reservation process is still done manually, causing inefficiency and inaccuracy of information. Based on this background, the Room Reservation Information System (SISERU) is designed as a solution that integrates information technology to manage room reservations more efficiently and effectively. The design of this system uses the Codeigniter 3 framework with APIs that can allow integration of features between two applications to interact. SISERU has features such as searching rooms by date and time, online booking, booking cancellation, and booking status notification. Application testing is carried out using the blackbox method, performance testing, and user acceptance test (UAT) to determine the reliability of the application. Based on the test results, the application successfully completed all predetermined scenarios with stable usage performance in response 0.0086 to 0.2781 seconds and memory usage 0.62 to 0.67 MB. In addition, the application is also well received by users based on the results of the UAT method testing. SISERU is expected to help in optimising the use of rooms at Diponegoro University.

Keywords : SISERU, Room Reservation Information System, Codeigniter 3, Website

1. Pendahuluan

Penggunaan teknologi saat ini telah diimplementasikan dalam berbagai sektor industri seperti otomotif, elektronik, makanan, dan lain-lain [1]. Dalam upaya meningkatkan efisiensi dan produktivitas, banyak industri yang beralih ke otomatisasi [2]. Salah satu aspek krusial dari otomatisasi adalah penggunaan Sistem Informasi yang menyediakan manajemen data yang efisien dan efektif untuk meningkatkan kinerja organisasi. Penerapan sistem informasi telah membawa dampak positif dalam bidang militer dengan meningkatkan produktivitas kerja dan kemampuan pengambilan keputusan [3]. Di lingkungan

akademik, sistem informasi juga membantu mengurangi beban manajerial yang berat dan meningkatkan kinerja [4], serta menyederhanakan proses seperti persiapan dan penanganan dokumen [5].

Salah satu kegiatan yang sering dilakukan oleh berbagai organisasi adalah persiapan dan penanganan dokumen, termasuk di lembaga pendidikan. Di tingkat universitas, pengelolaan ruangan merupakan aspek penting yang perlu diperhatikan. Universitas memiliki berbagai jenis ruangan yang digunakan untuk kegiatan seperti ruang

kelas, laboratorium, seminar, dan rapat. Namun, seringkali mengelola ketersediaan ruangan menjadi sulit karena kurangnya informasi yang akurat dan terkini. Dalam konteks ini, sistem informasi dapat mempercepat, mempermudah, dan meningkatkan efisiensi penanganan dan persiapan dokumen di lembaga pendidikan [6]. Penggunaan sistem informasi juga memudahkan akses dan operasional bagi orang-orang dengan fasilitas dan kinerja yang lebih baik [7]. Manajemen fasilitas dan reservasi menjadi proses penting dalam penyelenggaraan kelas yang melibatkan berbagai pengguna, seperti manajer fasilitas, mahasiswa, dan dosen. Untuk mengelola ruangan dengan baik, sistem reservasi membutuhkan penanganan dan pemrosesan dokumen yang efisien, sehingga tumpang tindih reservasi dapat diminimalisir [8].

Pengembangan sistem reservasi ruangan otomatis berbasis web membantu pihak perguruan tinggi untuk mengurangi berbagai pemborosan (waste) saat proses reservasi ruangan secara manual berlangsung meliputi pemborosan transportasi, menunggu persetujuan permintaan reservasi, mengulang isian formulir reservasi, mencari jadwal ruangan yang akan digunakan, inventaris kertas fisik untuk formulir, produksi laporan dan salinan dokumen, dan cacat kesalahan entri data reservasi. Melalui pengembangan sistem informasi reservasi ruangan berbasis web, sebuah platform dapat berguna untuk memfasilitasi pengelolaan ruang dan fasilitas, berbagi informasi secara real-time, meminimalkan usaha dan waktu yang dibutuhkan untuk melakukan reservasi, memungkinkan pengguna mengakses dan bertukar informasi dengan mudah, serta membantu dalam melestarikan lingkungan dengan transaksi tanpa kertas [9].

2. Metode

2.1 Pembuatan Aplikasi

Dalam pembuatan serta pengembangan “Aplikasi Sistem Informasi Reservasi Ruangan Berbasis Website” dengan menggunakan metode pengembangan *waterfall*. Metode

waterfall digunakan karena memungkinkan manajemen sumber daya yang lebih mudah diprediksi dan disetiap fase sumber daya yang diperlukan dapat memperkirakan dengan lebih baik karena disetiap fase memiliki cakupan dan tujuannya tersendiri [10-13]. Aplikasi ini menggunakan *framework* CodeIgniter yang berbasis bahasa pemrograman PHP [14][15].

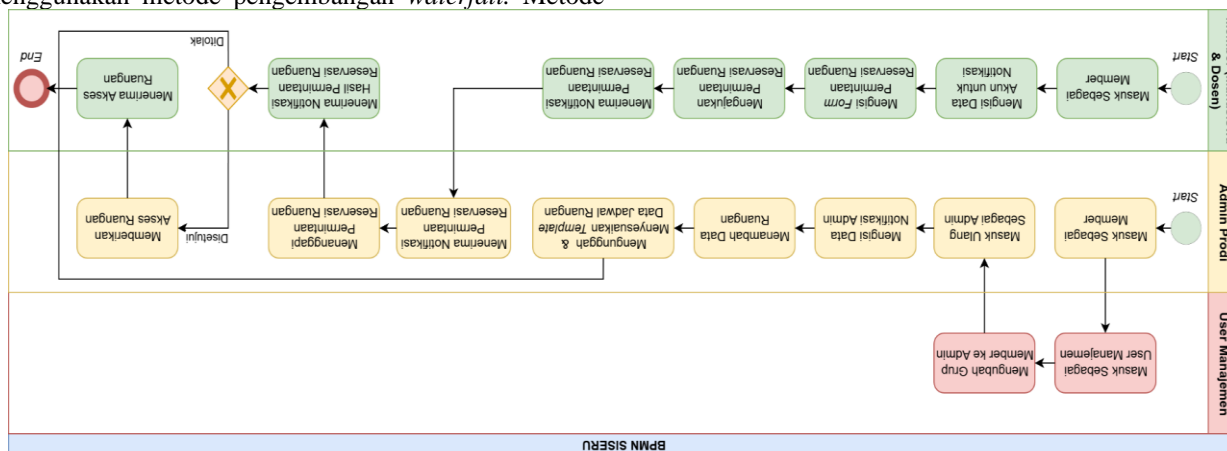
2.1.1 Business Process Aplikasi SISERU

Proyek pembuatan aplikasi SISERU ini membutuhkan sebuah *business process* yang dapat digunakan untuk memberikan penjelasan bagi *user* yang ingin menggunakan aplikasi ini. *Business process* merupakan urutan aktivitas atau tugas yang dilakukan oleh individu atau sistem untuk mencapai suatu tujuan bisnis. Pembuatan *business process* penting dilakukan selain untuk memberikan penjelasan kepada *user*, juga dapat menjadi dasar dari perancangan arsitektur aplikasi nantinya. *Business process* dibuat di dalam bentuk BPMN (*Business Process Model Notations*) [16].

Tujuan dari penyusunan BPMN adalah untuk memberikan panduan visual yang jelas dan terstruktur terkait langkah yang harus diikuti oleh pengguna dalam menjalankan aplikasi SISERU. Secara umum, terdapat tiga buah *user* di dalam sistem aplikasi SISERU ini, diantaranya yaitu:

- User* manajemen dimana sebagai pemegang kekuasaan tertinggi. Ini dapat melibatkan atasan pihak prodi. Perannya untuk menentukan antara admin dan *member*.
- Admin prodi berperan sebagai penanggung jawab aplikasi SISERU, dimana mengelola data ruangan, jadwal ruangan, reservasi ruangan, dan manajemen notifikasi.
- Member yaitu mahasiswa dan dosen berperan yang dapat mengajukan reservasi ruangan melalui aplikasi.

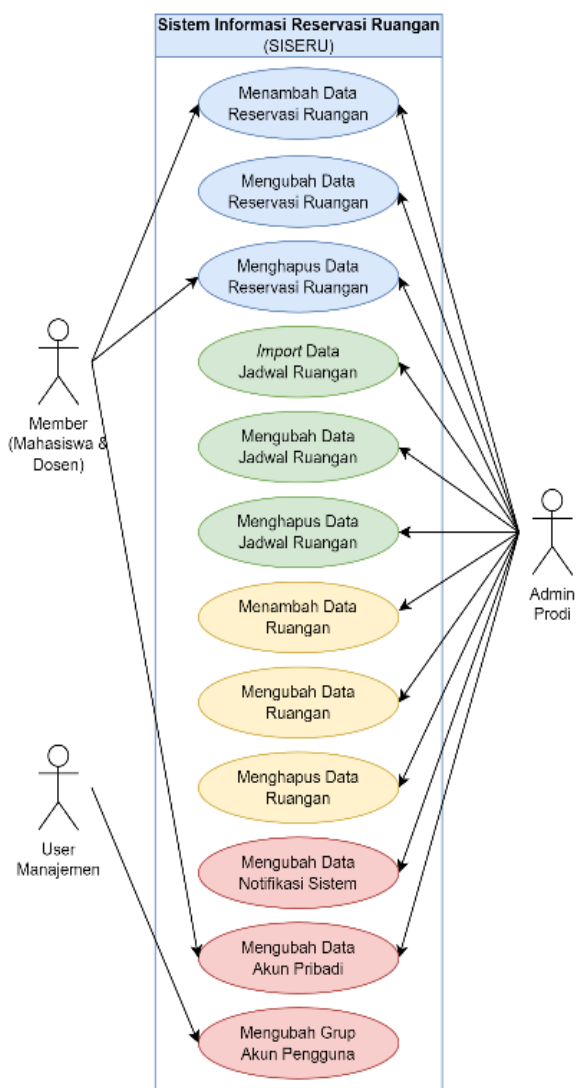
Proses bisnis aplikasi SISERU dapat dilihat pada Gambar 1.



Gambar 1. Business Process Aplikasi SISERU

2.1.2 Use Case Diagram Perancangan Aplikasi SISERU

Dalam SISERU dirancang *use case* diagram untuk mengorganisasikan dan memodelkan perilaku dari suatu sistem yang dibutuhkan serta diharapkan pengguna [17]. Berikut ini adalah *use case* yang menggambarkan interaksi antara aktor (pengguna) dengan sistem. *Use Case* Diagram Aplikasi SISERU dapat dilihat pada Gambar 2.

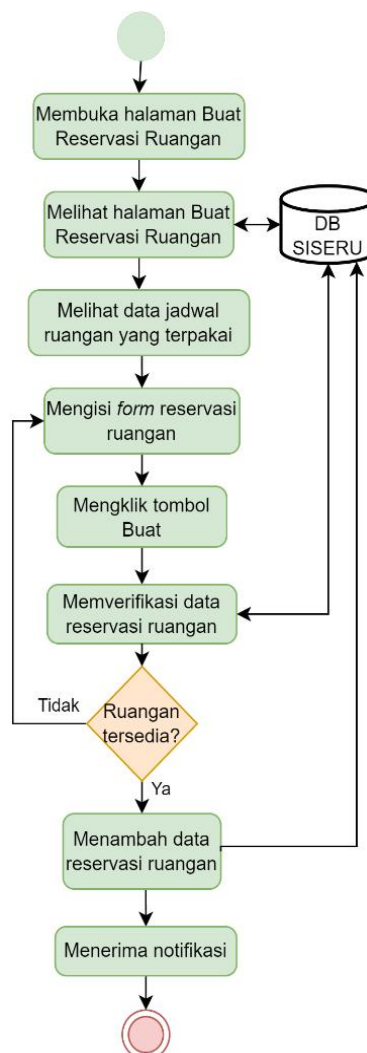


Gambar 2. Use Case Diagram Aplikasi SISERU

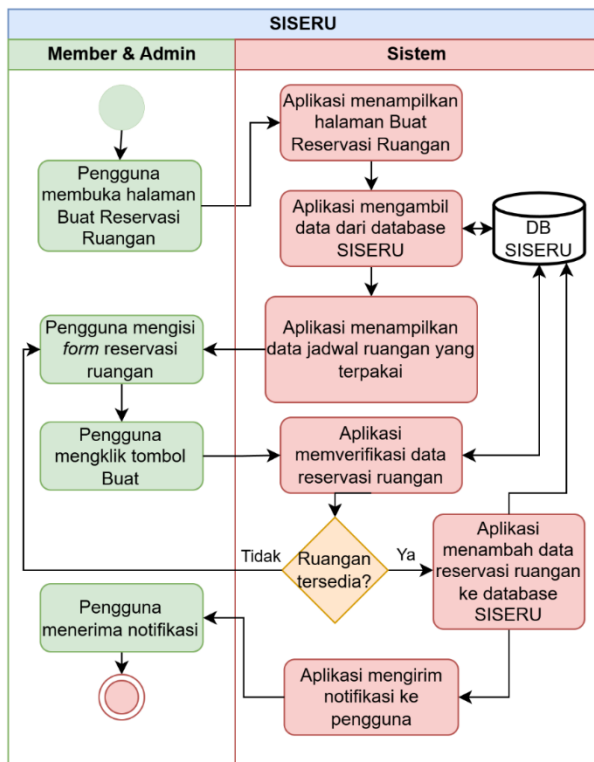
2.1.3 Diagram Aktivitas Aplikasi SISERU

Aktivitas diagram pada aplikasi SISERU dapat menggambarkan proses dari pengguna masuk ke dalam sistem, hingga memilih halaman tertentu. Setiap aktivitas mencakup interaksi antara pengguna dan antarmuka aplikasi, serta keterlibatan *database*. Aktivitas diagram akan membantu dalam merancang, memahami, dan memperbaiki alur kerja aplikasi, sehingga memastikan pengalaman pengguna yang lancar dan efisien. Aktivitas utama pada aplikasi SISERU adalah membuat reservasi ruangan.

Pada tahap ini, pengguna akan melakukan reservasi ruangan. Untuk memulai proses reservasi ruangan, pengguna harus masuk ke halaman Buat Reservasi Ruang. Sistem akan menampilkan data jadwal ruangan yang telah terpakai. Setelah itu, pengguna akan diminta untuk mengisi formulir reservasi ruangan dan akan diminta untuk mengklik tombol Buat. Sistem akan memverifikasi data reservasi ruangan yang diinputkan oleh pengguna. Jika data reservasi ruangan yang diinputkan oleh pengguna sudah sesuai dan ruangan masih tersedia, maka sistem akan menyimpan data reservasi ruangan tersebut ke dalam *database* dan mengirimkan notifikasi ke pengguna. Notifikasi dikirimkan oleh sistem ke pengguna yang meminta reservasi ruangan dan ke admin. Jika data reservasi ruangan yang diinputkan oleh pengguna belum sesuai dan ruangan tidak tersedia, maka sistem akan meminta pengguna untuk mengisi ulang formulir reservasi ruangan. Aktivitas diagram ini dapat dilihat pada Gambar 3. Gambar 4 merupakan aktivitas diagram yang terhubung dengan sistem.



Gambar 3. Diagram Aktivitas pada Halaman Membuat Reservasi Ruang



Gambar 4. Diagram Aktivitas pada Halaman Membuat Reservasi Ruang Terhubung dengan Sistem

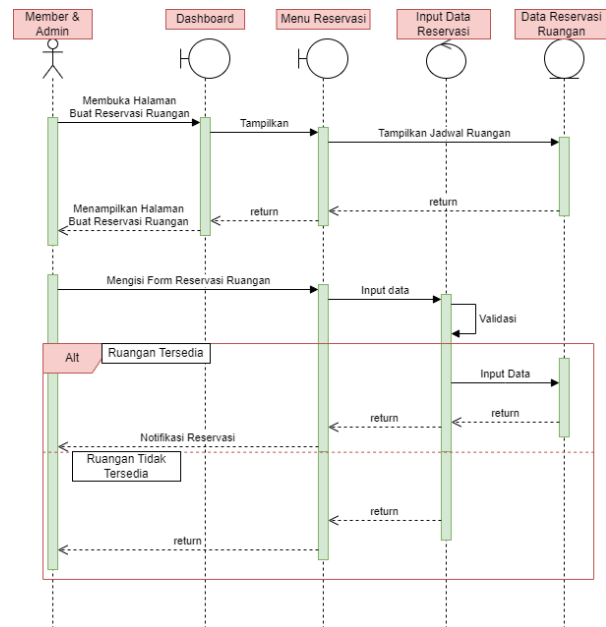
2.1.4 Sequence Diagram Perancangan Aplikasi SISERU

Sequence diagram adalah salah satu jenis diagram dari UML yang menunjukkan bagaimana suatu kegiatan atau objek berinteraksi dalam urutan waktu tertentu. Sequence diagram merepresentasikan alur kegiatan dan berbagai komponen lainnya dalam sebuah sistem secara berurutan dari waktu ke waktu.

Sequence diagram pada aplikasi SISERU menggambarkan proses kegiatan dan interaksi antar komponen secara berurutan. Setiap aktivitas dimulai dari kegiatan yang dilakukan oleh pengguna hingga hasil yang diterima oleh pengguna tersebut. Sequence diagram dibuat guna membantu diagram aktivitas untuk memperjelas alur bekerja suatu sistem dalam melakukan suatu aktivitas dari waktu ke waktu. Berikut merupakan hasil sequence diagram dari SISERU.

Sequence diagram fitur utama dari aplikasi SISERU adalah membuat reservasi ruangan. Sequence diagram ini merupakan alur ketika member dan admin melakukan aktivitas membuat reservasi ruangan. Pertama pengguna membuka halaman buat reservasi ruangan melalui dashboard. Kedua, halaman reservasi kemudian akan membaca data reservasi ruangan untuk ditampilkan kepada pengguna. Setelah halaman reservasi tertampil, maka pengguna dalam melakukan reservasi dengan cara mengisi form reservasi ruangan. Controller kemudian akan melakukan validasi setelah data pada form dimasukkan oleh pengguna. Apabila ruangan yang dimasukkan oleh

pengguna tersedia, maka controller akan memasukkan data ke dalam database dan akan memberikan notifikasi kepada pengguna. Apabila ruangan yang dimasukkan oleh pengguna tidak tersedia, maka hasil akan dikembalikan kepada pengguna agar pengguna dapat mengisi ulang ruangan lain yang tersedia. Sequence diagram dapat dilihat pada Gambar 5.



Gambar 5. Sequence Diagram pada Aktivitas Membuat Reservasi Ruang

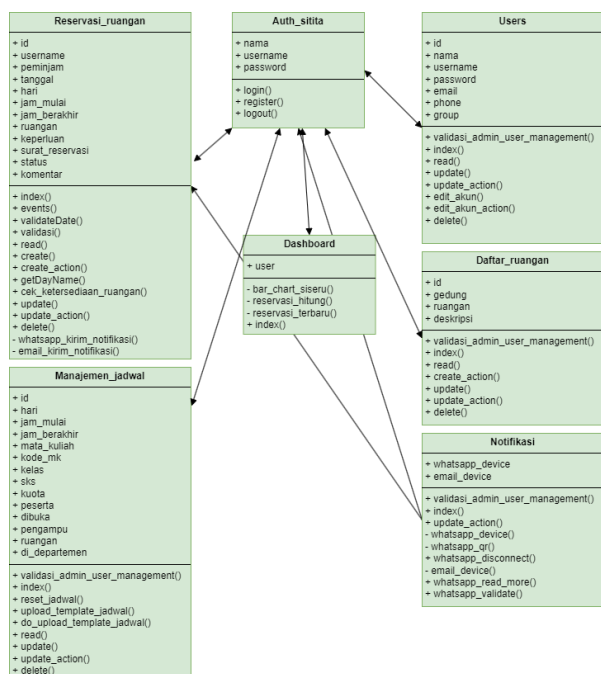
2.1.5 Class Diagram Perancangan Aplikasi SISERU

Class diagram merupakan suatu diagram bagian dari UML yang menggambarkan struktur sebuah sistem dengan menyertakan class pada sistem, atribut, metode, hingga hubungan atau relationship antar kelas yang digunakan. Diagram ini digunakan untuk membantu dalam melakukan perancangan sistem agar lebih lebih jelas dan lengkap sesuai dengan kaidah pengembangan sistem. Class diagram pada aplikasi SISERU berisi kelas yang ada pada aplikasi. Setiap kelas tersebut memiliki atribut dan metodenya dan saling terhubung satu sama lain. Class diagram aplikasi SISERU dapat dilihat pada Gambar 6.

Aplikasi SISERU memiliki 7 kelas yaitu Auth_sitita, User, Dashboard, Reservasi_ruangan, Manajemen_jadwal, Daftar_ruangan, dan Notifikasi. Setiap kelas memiliki atribut dan metodenya masing-masing.

Kelas Reservasi_ruangan berguna melakukan proses yang berhubungan dengan reservasi ruangan. Atribut pada kelas Reservasi_ruangan meliputi id, username, peminjam, tanggal, hari, jam_mulai, jam_berakhir, ruangan, keperluan, surat_reservasi, status, dan komentar. Sedangkan metode pada kelas Reservasi adalah index(), events(), validateDate(), validasi(), read(), create(), create_action(), getDayName(), cek_ketersediaan_ruangan(), update(),

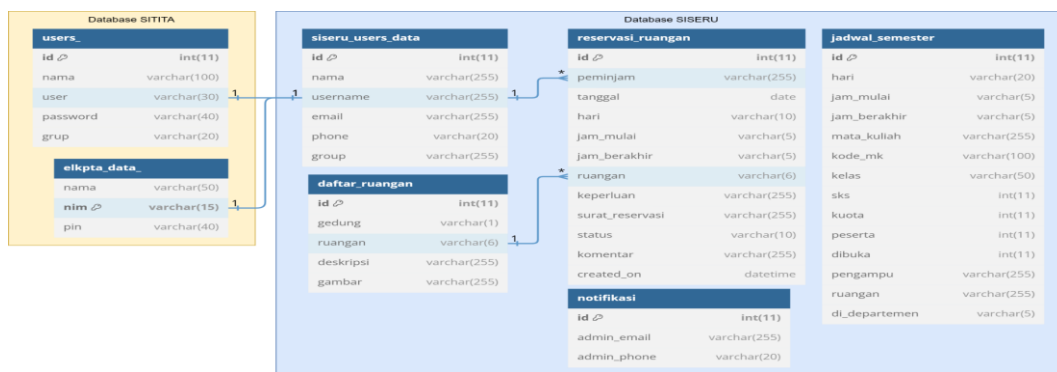
update_action(), delete(), whatsapp_kirim_notifikasi(), dan email_kirim_notifikasi(). Dari sini dapat diketahui bahwa kelas Reservasi memiliki metode untuk melakukan proses yang berhubungan dengan peminjaman atau reservasi ruangan. *Class Diagram* pada Aplikasi SISERU dapat dilihat pada Gambar 6



Gambar 6. *Class Diagram* pada Aplikasi SISERU

2.1.6 Diagram Database Perancangan Aplikasi SISERU

Sistem reservasi ruangan SISERU (Sistem Informasi Reservasi Ruangan) dirancang dengan menghubungkan dua *database* utama, yaitu *database* SITITA (Sistem Informasi Kerja Praktik dan Tugas Akhir) serta *database* SISERU itu sendiri. Integrasi dengan *database* SITITA diperlukan karena SISERU akan menggunakan data pengguna yang sudah terdaftar pada aplikasi SITITA untuk proses autentikasi dan otorisasi akses ke dalam sistem reservasi ruangan. Pada *database* SITITA, terdapat tabel *users_* dan *elkpta_data_* yang menyimpan informasi seperti nama, NIM/NIP/Username, dan *password/pin* dari pengguna SITITA. Dalam *database* SISERU terdapat tabel *siseru_users_data* yang isinya adalah data pengguna seperti nama, *username*, email, nomor telepon, dan *role*. Tabel *daftar_ruangan* berisi data ruangan yang disediakan seperti gedung, dan kode ruangan. Tabel *reservasi_ruangan* berisi data reservasi ruangan yang sudah dikirimkan oleh pengguna seperti peminjam, tanggal, hari, jam mulai, jam berakhir, ruangan, keperluan, surat pernyataan reservasi ruangan, status, komentar, dan waktu reservasi dibuat. Tabel *jadwal_semester* berisi data jadwal ruangan yang terpakai setiap minggu mengikuti jadwal mata kuliah semester seperti hari, jam mulai, jam berakhir, mata kuliah, kode mata kuliah, kelas, SKS, kuota, peserta, pengampu, dan ruangan. Tabel *notifikasi* berisi data notifikasi admin yang digunakan sebagai pengiriman notifikasi ke admin seperti email admin dan nomor telepon admin. Diagram basis data yang menggambarkan struktur dari *database* aplikasi SISERU dapat dilihat pada Gambar 7.



Gambar 7. *Database Diagram* Aplikasi SISERU

3 Hasil dan Analisis

3.1 Implementasi dan Pengujian Database SISERU dengan Operasi CRUD (Create, Read, Update, Delete)

Berdasarkan hasil dari tahap perancangan sistem seperti desain basis data, bisnis proses, *use case*, dan *activity*, langkah selanjutnya adalah melakukan implementasi dan pengujian operasi CRUD (*Create, Read, Update, Delete*) pada basis data untuk memastikan

fungsionalitas penyimpanan dan pengelolaan data dapat berjalan dengan baik.

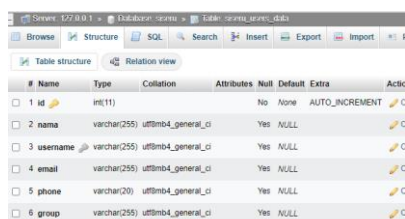
Operasi CRUD merupakan operasi dasar dalam pengolahan basis data. *Create* berfungsi untuk menambahkan data baru ke dalam basis data menggunakan perintah *INSERT*. *Read* berfungsi untuk membaca atau menampilkan data yang tersimpan di basis data menggunakan perintah *SELECT*. *Update* berfungsi untuk mengubah data yang sudah ada di basis data menggunakan

perintah *UPDATE*. Sedangkan *Delete* berfungsi untuk menghapus data dari basis data menggunakan perintah *DELETE*.

Pengujian CRUD dilakukan dengan mensimulasikan setiap operasi tersebut melalui antarmuka *website* SISERU yang telah diimplementasikan. Pengujian ini mencakup memasukkan data baru, menampilkan data, mengubah data yang sudah ada, serta menghapus data. Setiap operasi diuji untuk memastikan berfungsi dengan benar dan tidak terjadi kesalahan atau kehilangan data.

3.1.1 Implementasi dan Pengujian Database Tabel *siseru_users_data*

Pada *database* tabel *siseru_users_data* ini merupakan tabel untuk menampung data seluruh pengguna yang menggunakan aplikasi SISERU. Kolom *nama* dan *username* diambil dari *database* SITITA dimana kolom *username* bersifat *unique*, kolom *email* dan *phone* diisikan oleh pengguna, kolom *group* berisi *role* dari pengguna tersebut. Tampilan struktur *database* SISERU tabel *siseru_users_data* dapat dilihat pada Gambar 8. Hasil pengujian untuk *query* INSERT, SELECT, dan UPDATE dapat dilihat pada Tabel 1.



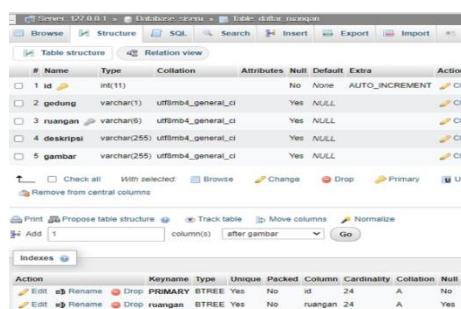
Gambar 8. Tampilan Struktur Database SISERU tabel *siseru_users_data*

Tabel 1. Hasil Pengujian Query pada Database SISERU tabel *siseru_users_data*

Pengujian	Hasil
INSERT	√
SELECT	√
UPDATE	√

3.1.2 Implementasi dan Pengujian Database Tabel *daftar_ruangan*

Pada *database* tabel *daftar_ruangan* ini merupakan tabel untuk menampung data seluruh ruangan yang bisa digunakan reservasi. Kolom *gedung*, *ruangan*, *deskripsi*, *gambar* diisikan oleh admin dimana kolom *ruangan* bersifat *unique*. Tampilan struktur *database* SISERU tabel *daftar_ruangan* dapat dilihat pada Gambar 9. Hasil pengujian untuk *query* INSERT, SELECT, UPDATE, dan DELETE dapat dilihat pada Tabel 2.



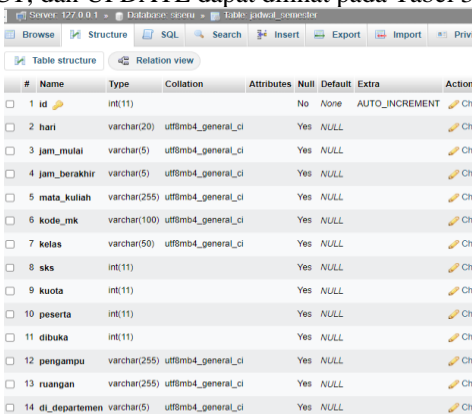
Gambar 9. Tampilan struktur *database* SISERU tabel *daftar_ruangan*

Tabel 2. Hasil Pengujian Query pada Database SISERU tabel *daftar_ruangan*

Pengujian	Hasil
INSERT	√
SELECT	√
UPDATE	√
DELETE	√

3.1.3 Implementasi dan Pengujian Database Tabel *jadwal_semester*

Pada *database* tabel *jadwal_semester* ini merupakan tabel untuk menampung data seluruh jadwal mata kuliah semester berjalan yang memakai ruangan-ruangan kelas. Kolom *hari*, *jam_mulai*, *jam_berakhir*, *mata_kuliah*, *kode_mk*, *kelas*, *sks*, *kuota*, *peserta*, *dibuka*, *pengampu*, dan *ruangan* diisikan oleh admin dengan cara menggunakan fitur *upload file excel* yang didapat dari SIAP Undip, kolom *di_departemen* akan diisi otomatis oleh sistem SISERU dimana sistem akan mendeteksi kode ruangan yang dipakai pada jadwal semester. Tampilan struktur *database* SISERU tabel *daftar_ruangan* dapat dilihat pada Gambar 10. Hasil pengujian untuk *query* INSERT, SELECT, dan UPDATE dapat dilihat pada Tabel 3.



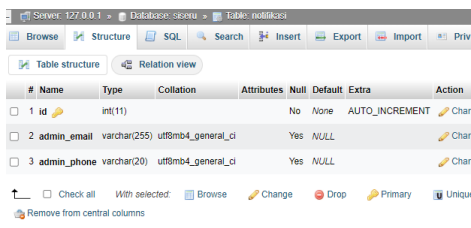
Gambar 10. Tampilan struktur *database* SISERU tabel *jadwal_semester*

Tabel 3. Hasil Pengujian Query pada Database SISERU tabel *jadwal_semester*

Pengujian	Hasil
INSERT	√
SELECT	√
UPDATE	√
DELETE	√

3.1.4 Implementasi dan Pengujian Database Tabel notifikasi

Pada database tabel notifikasi ini merupakan tabel untuk menampung data notifikasi untuk admin. Kolom *admin_email*, dan *admin_phone* diisikan oleh admin. Tampilan struktur database SISERU tabel notifikasi dapat dilihat pada Gambar 11. Hasil pengujian untuk query INSERT, SELECT, dan UPDATE dapat dilihat pada Tabel 4.



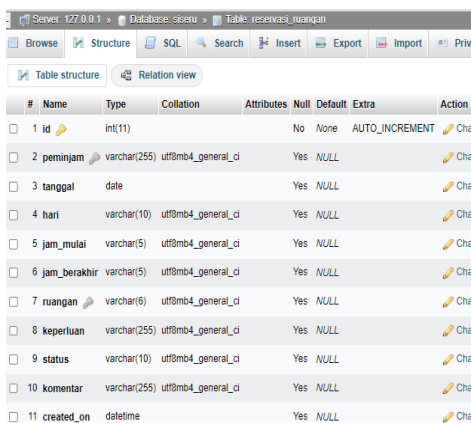
Gambar 11. Tampilan struktur database SISERU tabel notifikasi

Tabel 4. Hasil Pengujian Query pada Database SISERU tabel notifikasi

Pengujian	Hasil
INSERT	√
SELECT	√
UPDATE	√

3.1.5 Implementasi dan Pengujian Database Tabel reservasi_ruangan

Pada database tabel reservasi_ruangan ini merupakan tabel untuk menampung data seluruh reservasi ruangan yang dibuat oleh pengguna. Kolom *peminjam* dan *created_on* akan diisi otomatis oleh sistem SISERU dimana sistem akan mendeteksi *session* yang melakukan reservasi serta waktu saat melakukan reservasi, kolom *tanggal*, *hari*, *jam_mulai*, *jam_berakhir*, *ruangan*, *keperluan*, *status*, dan *komentar* diisi oleh pengguna. Tampilan struktur database SISERU Tabel reservasi_ruangan dapat dilihat pada Gambar 12. Hasil pengujian untuk query INSERT, SELECT, UPDATE, dan DELETE dapat dilihat pada Tabel 5.



Gambar 12. Tampilan struktur database SISERU tabel reservasi_ruangan

Tabel 5. Hasil Pengujian Query pada Database SISERU tabel reservasi_ruangan

Pengujian	Hasil
INSERT	√
SELECT	√
UPDATE	√
DELETE	√

3.2 Implementasi dan Pengujian Aplikasi Website SISERU Menggunakan Black Box

Pengujian *black box* merupakan teknik pengujian perangkat lunak yang dapat menguji fungsionalitas tanpa memahami struktur kode internal, detail implementasi, atau jalur internal. Pengujian ini hanya memeriksa *input* dan *output* aplikasi. Tabel 6 merupakan hasil implementasi perancangan sistem dan pengujian *black box* aplikasi website SISERU.

Tabel 6. Pengujian Blackbox pada Aplikasi SISERU

Pengujian	Hasil
Login	√
Dashboard	√
Kalender jadwal data ruangan reservasi yang tersedia	√
Buat reservasi ruangan	√
List reservasi ruangan	√
Upload template jadwal	√
List data jadwal	√
List pengguna	√
Daftar ruangan	√
List daftar ruangan	√
Manajemen notifikasi	√
Edit akun	√

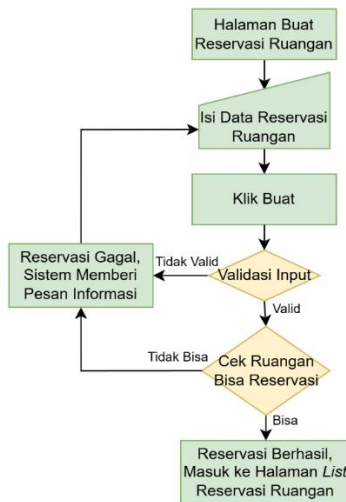
Dapat dilihat pada Tabel 6, semua pengujian *blackbox* pada aplikasi SISERU semua memiliki hasil baik dimana aplikasi dapat digunakan sesuai dengan yang diharapkan.

3.3 Pengujian Fungsi Utama Aplikasi Website SISERU Menggunakan White Box

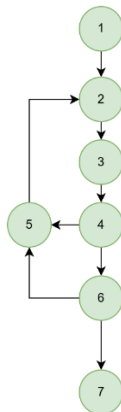
Pengujian *white box* merupakan metode pengujian perangkat lunak yang berfokus pada pengujian struktur internal, desain, dan kode program perangkat lunak. Berbeda dengan pengujian *black box* yang berfokus pada fungsionalitas, pengujian *white box* memerlukan pemahaman mendalam tentang kode sumber perangkat lunak. Grafik alir (*flowgraph*) berperan sebagai representasi visual sederhana yang menggambarkan aliran kontrol dalam struktur program. Dalam *flowgraph*, anak panah atau sisi (*edge*, E) merepresentasikan aliran kontrol, sedangkan lingkaran atau simpul (*node*, N) menggambarkan satu atau lebih aksi atau pernyataan logis. Area yang dibatasi oleh sisi dan simpul disebut sebagai region (R). Simpul yang memiliki lebih dari satu sisi keluar disebut sebagai *predicate node* (P) karena merepresentasikan sebuah keputusan dalam program. Untuk mengukur kompleksitas suatu program dan menentukan jumlah jalur independen yang perlu diuji,

digunakan metrik yang disebut *Cyclomatic Complexity*. Metrik ini memberikan representasi kuantitatif dari kompleksitas logika program.

Pada fungsi utama buat reservasi ruangan ini merupakan bagian untuk menangani dan memproses data reservasi yang dimasukkan seperti tanggal, jam mulai, jam terakhir, ruangan, keperluan, dan file surat pernyataan reservasi ruangan. *Flowchart* dari fungsi ini dapat dilihat pada gambar Gambar 13. dan *flowgraph* dari fungsi ini dapat dilihat pada gambar Gambar 14.



Gambar 13. *Flowchart* Fungsi Utama Buat Reservasi Ruangan



Gambar 14. *Flowgraph* Fungsi Utama Buat Reservasi Ruangan

Dari Gambar 14, dapat diketahui sebagai berikut, $node = 7$; $edge = 8$; $region = 3$; $predicate = 2$. Sehingga dapat diperoleh perhitungan *cyclomatic complexity* sebagai berikut.

- $V(G) = \text{jumlah area } (R)$
 $V(G) = 3$
- $V(G) = E - N + 2$
 $V(G) = 8 - 7 + 2$
 $V(G) = 3$
- $V(G) = P + 1$
 $V(G) = 2 + 1$
 $V(G) = 3$

Dari hasil perhitungan di atas, diperoleh jalur independen sebagai berikut.

- Jalur 1 = 1-2-3-4-6-7
- Jalur 2 = 1-2-3-4-5-2-3-4-6-7
- Jalur 3 = 1-2-3-4-6-5-2-3-4-6-7

Setelah diketahui jumlah jalur independennya, maka akan dilakukan perbandingan menggunakan tabel hubungan antara *cyclomatic complexity* dan risiko dapat dilihat pada Tabel 7. berikut.

Tabel 7. Hubungan *Cyclomatic Complexity* dengan Risiko

Nilai CC	Tipe Prosedur	Tingkat Risiko
1 - 4	Prosedur Sederhana	Rendah
5 - 10	Prosedur yang terstruktur dengan baik dan stabil	Rendah
11 - 20	Prosedur yang lebih kompleks	Menengah
21 - 50	Prosedur yang kompleks dan kritis	Tinggi
> 50	Rentan kesalahan, sangat mengganggu, prosedur tidak dapat diuji	Sangat Tinggi

Menurut Tabel 7. untuk fungsi buat reservasi ruangan ini memiliki risiko yang rendah dengan tingkat prosedur yang sederhana karena memiliki jalur independen berjumlah 3. Kemudian, setelah diketahui jalur independennya maka langkah selanjutnya adalah *test case*. *Test case* dari fungsi buat reservasi ini dapat dilihat pada Tabel 8. berikut:

Tabel 8. Hasil *Test Case* pada Fungsi Utama Buat Reservasi Ruangan

Pengujian	Hasil
Mengisi data reservasi ruangan yang valid dan ruangan yang dipilih bisa dilakukan reservasi	✓
Mengisi data reservasi ruangan yang tidak valid seperti tanggal sudah terlewat kemudian mengklik tombol buat, lalu mengisi kembali data yang valid dan ruangan yang dipilih bisa dilakukan reservasi	✓
Mengisi data reservasi ruangan yang valid kemudian mengklik tombol buat namun ruangan yang dipilih tidak bisa dilakukan reservasi, lalu pengguna mengulangi mengisi data reservasi dan memilih ruangan yang bisa direservasi	✓

Pada Tabel 8. *test case* yang dibuat berdasarkan jalur independen yang telah dibuat dan telah didapatkan hasil valid untuk ketiga pengujian yang dilakukan yang berarti tidak ditemukan *error* pada fungsi utama buat reservasi ruangan yang diuji.

3.4 Pengujian Performa Aplikasi Website SISERU

Pada penelitian ini, bentuk pengujian dilakukan dengan pengukuran waktu *respond* sistem saat melakukan pemrosesan data menggunakan fungsi *benchmark* pada *framework codeigniter 3*. Fungsi *benchmark* ini terdapat pada *library benchmarking class*. Dalam mengukur total waktu eksekusi (*total execution time*) dengan cara memanggil *method elapsed_time()* pada setiap halaman yang akan diukur. Kemudian, dalam mengukur konsumsi memori (*memory consumption*) dengan cara memanggil *method memory_usage()* pada setiap halaman yang akan diukur. Hasil pengujian performa sistem pada setiap halaman dapat dilihat pada Tabel 9.

Tabel 9. Hasil Pengujian Performa Website SISERU

No.	Halaman	Total Waktu Eksekusi	Konsumsi Memori
1.	Dashboard	0,0145 detik	0,67 MB
2.	Buat Reservasi Ruang	0,0148 detik	0,64 MB
3.	List Reservasi Ruang	0,0134 detik	0,64MB
4.	Upload Template Jadwal	0,0091 detik	0,63MB
5.	List Data Jadwal	0,0090 detik	0,63MB
6.	List Pengguna	0,0115 detik	0,62MB
7.	Daftar Ruang	0,0105 detik	0,63MB
8.	Manajemen Notifikasi	0,2781 detik	0,62MB
9.	Edit Akun	0,0086 detik	0,62MB

3.5 Pengujian Aplikasi Website SISERU Menggunakan User Acceptance Test

Pengujian UAT (*User Acceptance Testing*) merupakan tahap pengujian yang dilakukan oleh pengguna akhir untuk menilai sejauh mana sebuah sistem atau perangkat lunak memenuhi kebutuhan bisnis dan persyaratan pengguna. Pengujian ini melibatkan pihak mahasiswa dari Universitas Diponegoro. Pengujian dilakukan dengan memberikan halaman kuisioner menggunakan *Google Form* kepada beberapa pihak mahasiswa. Bobot penilaian yang dapat diberikan dijabarkan pada Tabel 10.

Tabel 10. Pengujian UAT Aplikasi Website SISERU

No.	Pernyataan	A	B	C	D	E
1.	Tampilan aplikasi website mudah dipahami	14	7			
2.	Menu aplikasi website mudah dipahami dan digunakan	13	8			
3.	Aplikasi website menampilkan jumlah reservasi ruangan secara <i>real-time</i>	14	6	1		

No.	Pernyataan	A	B	C	D	E
4.	Aplikasi website menampilkan grafik statistik reservasi ruangan secara <i>real-time</i>	15	6			
5.	Aplikasi website menampilkan detail reservasi ruangan secara <i>real-time</i>	14	6	1		
6.	Ruangan, Tanggal dan waktu reservasi ruangan ditampilkan dengan jelas pada <i>list</i> reservasi	14	7			
7.	Fitur-fitur pada aplikasi website sesuai dengan harapan	10	7	4		
8.	Aplikasi website dapat membantu pihak mahasiswa, dosen, dan <i>staff</i> untuk reservasi ruangan	15	5	1		
9.	Aplikasi website dapat digunakan oleh pihak mahasiswa, dosen, dan <i>staff</i> di kemudian hari	15	5	1		
10.	Secara keseluruhan, seberapa baik aplikasi website ini?	14	7			
Total		138	64	8		

Berdasarkan data pengguna yang didapatkan pada Tabel 8, dapat disimpulkan bahwa sebanyak 66% responden memilih nilai A (Sangat : Mudah/Bagus/Sesuai/Jelas), sebanyak 30% responden memilih nilai B (Mudah/Bagus/Sesuai/Jelas), dan sebanyak 4% responden memilih nilai C (Netral).

4. Kesimpulan

Aplikasi reservasi ruangan berbasis web, SISERU, telah dirancang dengan baik dan memenuhi kebutuhan yang diharapkan. Hal ini dibuktikan dengan berbagai pengujian yang telah dilakukan sebelumnya. Pengujian aplikasi SISERU menggunakan metode *black box* berhasil mendapatkan hasil aktual (*actual outcome*) sesuai dengan hasil harapan (*expected outcome*) secara keseluruhan. Kedua, pada pengujian mengakses aplikasi *website* Sistem Informasi Reservasi Ruangan (SISERU) dari segi performa sistem menunjukkan bahwa respon tetap konsisten dibawah 1 detik dengan pemakaian memori antara 0,62 MB hingga 0,67 MB, maka dapat dinilai sistem memiliki kinerja yang baik dalam melakukan pemrosesan data. Terakhir, Pada pengujian aplikasi *website* Sistem Informasi Reservasi Ruangan (SISERU) berdasarkan metode *User Acceptance Testing* (UAT) dapat diketahui bahwa dari 10 pernyataan yang diajukan kepada 21 mahasiswa Departemen Teknik Elektro Universitas Diponegoro didapat sebanyak 66% responden menilai bahwa aplikasi sangat mudah/bagus/sesuai/jelas (nilai A), 30% responden menilai bahwa aplikasi mudah/bagus/sesuai/jelas (nilai B), dan 4% responden menilai netral (nilai C). Hal ini menyimpulkan bahwa aplikasi penggunaan aplikasi diterima dengan baik oleh pengguna.

Referensi

- [1] C. Bai, P. Dallasega, G. Orzes, dan J. Sarkis, "Industry 4.0 technologies assessment: A sustainability perspective," *Int J Prod Econ*, vol. 229, hlm. 107776, Nov 2020, doi: 10.1016/j.ijpe.2020.107776.
- [2] J. Frohm, V. Lindström, M. Winroth, dan J. Stahre, "THE INDUSTRY'S VIEW ON AUTOMATION IN MANUFACTURING," *IFAC Proceedings Volumes*, vol. 39, no. 4, hlm. 453–458, 2006, doi: 10.3182/20060522-3-FR-2904.00073.
- [3] M. A. Arifin dan F. P. Tajudeen, "Impact of human resources information systems in the military environment," *Asia Pacific Management Review*, vol. 25, no. 4, hlm. 198–206, Des 2020, doi: 10.1016/j.apmr.2020.02.001.
- [4] N. A. Razak, H. Ab. Jalil, S. E. Krauss, dan N. A. Ahmad, "Successful implementation of information and communication technology integration in Malaysian public schools: An activity systems analysis approach," *Studies in Educational Evaluation*, vol. 58, hlm. 17–29, Sep 2018, doi: 10.1016/j.stueduc.2018.05.003.
- [5] K. Demir, "School Management Information Systems in Primary Schools.," *Turkish Online Journal of Educational Technology*, vol. 5, hlm. 32–45, 2006, [Daring]. Tersedia pada: <https://api.semanticscholar.org/CorpusID:56536121>
- [6] J. Martins dkk., "Assessing the success behind the use of education management information systems in higher education," *Telematics and Informatics*, vol. 38, hlm. 182–193, Mei 2019, doi: 10.1016/j.tele.2018.10.001.
- [7] F. Sanchez-Puchol, J. A. Pastor-Collado, dan B. Borrell, "Towards an Unified Information Systems Reference Model for Higher Education Institutions," *Procedia Comput Sci*, vol. 121, hlm. 542–553, 2017, doi: 10.1016/j.procs.2017.11.072.
- [8] C. Castillo, G. N. Rouskas, dan K. Harfoush, "Online algorithms for advance resource reservations," *J Parallel Distrib Comput*, vol. 71, no. 7, hlm. 963–973, Jul 2011, doi: 10.1016/j.jpdc.2011.01.003.
- [9] J. D. German, D. C. G. Yap, dan G. O. Binoya, "Design and Development of an Integrated Room Reservation System for Higher Education Institutions," dalam *2021 IEEE 8th International Conference on Industrial Engineering and Applications (ICIEA)*, IEEE, Apr 2021, hlm. 231–236. doi: 10.1109/ICIEA52957.2021.9436766.
- [10] D. Oktaviana dan M. A. Hamid, "Room Borrowing Information System Website Based using PHP and MySQL with Waterfall Model," *Journal of Vocational Education Studies*, vol. 4, no. 2, hlm. 210–218, Des 2021, doi: 10.12928/joves.v4i2.4400.
- [11] Y. Hendra, R. Tarigan, dan B. Badru, "RANCANG BANGUN APLIKASI RESERVASI HOTEL DEWIZA RESTO & CONVENTION HALL MENGGUNAKAN FRAMEWORK CODEIGNITER.," *Journal of Innovation And Future Technology (IFTECH)*, vol. 5, no. 2, hlm. 34–42, Agu 2023, doi: 10.47080/iftech.v5i2.2580.
- [12] A. Wirapraja, R. Widiatoro, dan J. Jason, "PERANCANGAN DAN SIMULASI SISTEM INFORMASI MANAJEMEN RESERVASI HOTEL BERBASIS WEB DENGAN METODE PROTOTYPING.," *JURNAL EKSEKUTIF*, vol. 19, no. 01, hlm. 1–9, Jul 2022, doi: 10.60031/jeksekutif.v19i01.301.
- [13] A. F. D. Utami, M. Ridwan, A. Khakim, dan E. D. Grafvera, "PENGEMBANGAN SISTEM INFORMASI RESERVASI HOMEYKU DI KALIURANG BERBASIS WEBSITE MENGGUNAKAN METODE WATERFALL.," *The Indonesian Journal of Computer Science Research*, vol. 2, no. 2, hlm. 1–7, Agu 2023, doi: 10.59095/ijcsr.v2i2.61.
- [14] M. Hills, "Evolution of dynamic feature usage in PHP," dalam *2015 IEEE 22nd International Conference on Software Analysis, Evolution, and Reengineering (SANER)*, IEEE, Mar 2015, hlm. 525–529. doi: 10.1109/SANER.2015.7081870.
- [15] F. Aulia dan Yahfizham, "Mengenal Bahasa Pemrograman pada Algoritma Pemrograman," *Journal Of Informatics And Busines (JIBS)*, vol. 1, no. 4, hlm. 223–228, 2024.
- [16] A. E. Prasetiadi, "WEB 3.0: TEKNOLOGI WEB MASA DEPAN.," *INDEPT*, vol. 1, no. 3, hlm. 1–6, 2011.
- [17] F. A. Pratama, S. P. D. Kristiana, dan C. M. Lukito, "The Implementation of SST as the System of Room Reservation for Students' Organization at Atma Jaya BSD Campus," *SISFORMA*, vol. 8, no. 2, hlm. 44–50, Jan 2022, doi: 10.24167/sisforma.v8i2.3434.