

# Academic Resources Management Information System (ARMIS) dengan Arsitektur Model View Control (MVC)

Novi Safriadi

Prodi Teknik Informatika, Fakultas Teknik Universitas Tanjungpura  
Jl. Ahmad Yani, Pontianak – 78124  
e-mail: safriadi@untan.web.id

**Abstract**– Dengan teknologi informasi saat ini memungkinkan berkembangnya teknik pemrograman perangkat lunak, salah satunya adalah perangkat lunak *resources manajemen project (RMP)*. *Resource ruangan* merupakan salah satu yang dapat dikembangkan dengan RMP. Dengan menggunakan teknologi framework dan dukungan arsitektur MVC, ARMIS merupakan RMP yang berbasis pada alokasi dan pemantauan ruangan pada suatu gedung. ARMIS yang diimplementasikan pada gedung Labtek V ITB dapat mengelola ruangan, mengalokasi, dan memantau ruangan secara online.

**Kata kunci**- *Resources Management Project, Framework, MVC, Use Case*

## 1. Pendahuluan

Perkembangan teknologi informasi berdampak baik dalam peningkatan kinerja dalam sistem suatu lembaga. Penerapan teknologi informasi berupa pembangunan perangkat lunak yang diimplementasikan untuk mendukung kinerja di dalam sistem. Produk perangkat lunak memiliki pengertian perangkat lunak yang ditambahkan dengan semua item dan pelayanan pendukung yang secara keseluruhan dapat memenuhi kebutuhan pemakai [1] kebutuhan tersebut biasanya diimplementasikan dalam suatu sistem informasi.

Sistem informasi merupakan sistem buatan manusia yang berisi himpunan terintegrasi dari komponen-komponen manual dan komponen-komponen terkomputerisasi yang bertujuan untuk mengumpulkan data, menyimpan data, memproses data, dan menghasilkan informasi untuk dipakai [2]. Salah satu penerapan sistem informasi adalah pada *Resources Management Project*.

*Resources Management Project (RMP)* merupakan sebuah proyek yang bertujuan untuk memudahkan pengguna dalam melakukan pengalokasian suatu *resource* atau sumber daya. Dalam penelitian ini akan menerapkan RMP untuk alokasi dan pemantauan ruangan di suatu gedung. RMP ini mengintegrasikan fungsi-fungsi pengalokasian dan pemantauan ruangan dan sub ruang. Dengan adanya RMP ini diharapkan dapat meningkatkan pelayanan dari pihak-pihak yang terlibat dalam pengalokasian dan penggunaan ruangan.

## 2. Framework dan MVC

Framework adalah sekumpulan perintah atau fungsi dasar yang dapat membantu dalam menyelesaikan

proses-proses yang lebih kompleks. Framework juga dapat diartikan sebagai kumpulan script (terutama class dan function) yang dapat membantu programmer dalam menangani berbagai masalah-masalah dalam pemrograman seperti koneksi ke database, pemanggilan variabel, file, dll sehingga developer lebih fokus dan lebih cepat membangun aplikasi.

Framework merupakan komponen pemrograman yang siap *re-use* kapan saja. sehingga programmer tidak harus membuat skrip yang sama untuk tugas yang sama.

Secara sederhana bisa dijelaskan bahwa framework adalah kumpulan fungsi (libraries), maka seorang programmer tidak perlu lagi membuat fungsi-fungsi (biasanya disebut kumpulan library) dari awal, programmer tinggal memanggil kumpulan library atau fungsi yang sudah ada didalam framework, tentunya cara menggunakan fungsi-fungsi itu sudah ditentukan oleh framework.

Secara umum, framework menggunakan struktur MVC (Model, View, Controller). MVC adalah *desain pattern* atau arsitektur yang digunakan dalam rekayasa perangkat lunak, dimana terjadi pemisahan yang jelas antara data (model) dengan user interface (view) [3]. Salah satu framework PHP yang menggunakan arsitektur MVC adalah CI (Code Igniter). Secara ilustrasi, arsitektur MVC berupa: Input > Processing > Output = Controller > Model > View

Model mencakup semua proses yang terkait dengan pemanggilan struktur data baik berupa pemanggilan fungsi, input processing atau mencetak output ke dalam browser.

Controller mencakup semua proses yang terkait dengan pemanggilan database dan kapsulisasi proses-proses utama. Jadi semisal di bagian ini ada file bernama *member.php*, maka semua proses yang terkait dengan *member* akan dikapsulisasi/dikelompokan dalam file ini. View mencakup semua proses yang terkait layout output. Bisa dibayangkan untuk menaruh template interface website atau aplikasi.

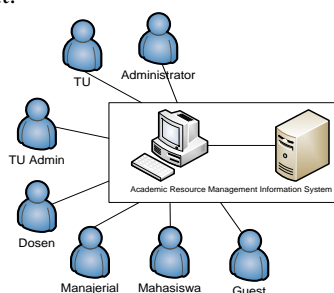
## 3. Desain ARMIS

ARMIS merupakan *resources management project* yang dikembangkan dengan framework CI. ARMIS ini dibangun diatas empat sub sistem yang masing-masing akan menangani manajemen resources, pengalokasian, *appointment* dan *utilization* ruangan. Keempat sub sistem tersebut dapat saling berhubungan dan saling

bertukar informasi dalam membentuk suatu aplikasi. ARMIS merupakan realisasi salah satu dari kedua sub sistem tersebut. ARMIS membantu pengguna dalam mengelola ruangan, mengalokasi, dan memantau ruangan. Dalam penelitian ini, ARMIS diimplementasikan pada Gedung LABTEK V Institute Teknologi Bandung.

### 3.1 Deskripsi Umum Sistem

*Academic Resource Management Information System* (ARMIS) dibangun untuk memantau dan mengalokasikan ruangan yang terdapat pada gedung Labtek V Institut Teknologi Bandung (ITB). Produk ini ditujukan bagi civitas akademika di lingkungan ITB maupun masyarakat umum yang berkepentingan. Perspektif perangkat lunak ARMIS ditampilkan seperti gambar berikut:

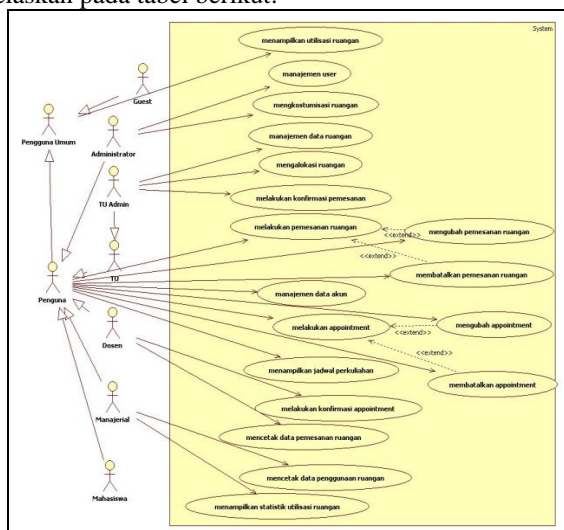


Gambar 1. Perspektif Perangkat Lunak ARMIS

### 3.2 Model Use Case

Model Use Case dirancang untuk menjelaskan kegiatan yang dilakukan oleh actor dan sistem. Use case class digunakan untuk memodelkan dan menyatakan unit fungsi/layanan yang disediakan oleh sistem (atau bagian sistem: subsistem atau class) ke pemakai. Gambar 2 merupakan diagram use case untuk ARMIS.

*Actor* mempresentasikan seseorang atau sesuatu (seperti perangkat, sistem lain) yang berinteraksi dalam ARMIS. Sebuah *actor* mungkin hanya memberikan informasi inputan pada sistem, hanya menerima informasi dari sistem atau keduanya menerima, dan memberi informasi pada sistem. *Actor* hanya berinteraksi dengan *use case*, tetapi tidak memiliki kontrol atas *use case*. Definisi actor dalam ARMIS dijelaskan pada tabel berikut:



Gambar 2. Diagram Use Case ARMIS

Tabel 1. Definisi Actor

No	Actor	Deskripsi
AC-01	Guest	<b>Guest</b> memiliki hak untuk menampilkan utilisasi ruangan.
AC-02	Pengguna	<b>Pengguna</b> memiliki hak untuk menampilkan utilisasi ruangan, melakukan pemesanan, manajemen data akun, melakukan appointment, dan menampilkan jadwal perkuliahan.
AC-03	Administrator	<b>Administrator</b> memiliki hak yang sama seperti <b>pengguna</b> dan memiliki kewenangan untuk manajemen user.
AC-04	TU Admin	<b>TU Admin</b> memiliki hak yang sama seperti <b>pengguna</b> dan memiliki kewenangan untuk manajemen data ruangan, mengalokasikan ruangan, mengkonfirmasi pemesanan ruangan dan mencetak data pemesanan ruangan.
AC-05	TU	<b>TU Admin</b> memiliki hak yang sama seperti <b>pengguna</b> .
AC-06	Dosen	<b>Dosen</b> memiliki hak yang sama seperti <b>pengguna</b> dan memiliki kewenangan untuk melakukan konfirmasi appointment.
AC-07	Manajerial	<b>Manajerial</b> memiliki hak yang sama seperti <b>pengguna</b> dan memiliki kewenangan untuk mencetak data penggunaan ruangan dan mencetak statistik utilisasi ruangan.
AC-08	Mahasiswa	<b>TU Admin</b> memiliki hak yang sama seperti <b>pengguna</b> .
AC-09	Pengguna Umum	<b>Guest</b> memiliki hak untuk menampilkan utilisasi ruangan.

*Use case* adalah gambaran fungsionalitas dari suatu sistem, sehingga *customer* atau pengguna sistem paham dan mengerti mengenai kegunaan sistem yang akan dibangun. Definisi use case pada ARMIS dijelaskan pada tabel berikut:

Tabel 2. Definisi Use Case

No	Use Case	Deskripsi
UC-01	Mengkostumi sasi ruangan	User melakukan perubahan yang berkaitan dengan identitas suatu ruangan.
UC-02	Manajemen user	User melakukan penambahan, penghapusan, dan pemblokiran akun user.
UC-03	Manajemen data akun	User melakukan perubahan terhadap data miliknya, misalnya password, alamat, telepon, email, dsb.
UC-04	Manajemen data ruangan	User melakukan manajemen ruangan yang meliputi penambahan, penghapusan, dan perubahan ruangan beserta atributnya (misal: deskripsi, penanggung jawab, dsb).
UC-05	Menampilkan utilisasi	User melihat ruangan mana saja yang sedang dipakai, digunakan

	ruangan	untuk apa dan siapa yang sedang menggunakannya, melihat ruangan yang sudah dipesan serta ruangan yang masih kosong pada hari dan waktu kapan saja.
UC-06	Mengalokasi ruangan	User mengalokasi ruangan berdasar jadwal kegiatan reguler, misalnya jadwal perkuliahan, jadwal ujian, dsb.
UC-07	Melakukan pemesanan ruangan	User melakukan pemesanan terhadap suatu ruangan untuk waktu tertentu.
UC-08	Mengubah pemesanan ruangan	User melakukan perubahan terhadap data pemesanan ruangan yang dibuatnya.
UC-09	Membatalkan pemesanan ruangan	User membatalkan pemesanan yang dilakukannya.
UC-10	Melakukan konfirmasi pemesanan ruangan	User melakukan konfirmasi bahwa pemesanan terhadap ruangan disetujui atau ditolak.
UC-11	Mencetak data pemesanan ruangan	User mencetak data pemesanan ruangan untuk periode waktu tertentu.
UC-12	Mencetak data penggunaan ruangan	User mencetak data penggunaan ruangan untuk periode waktu tertentu.
UC-13	Menampilkan statistik utilisasi ruangan	User melihat statistik utilisasi ruangan untuk periode waktu tertentu.
UC-14	Melakukan appointment	User melakukan appointment dengan user lainnya, disertai dengan pemesanan ruangan.
UC-15	Mengubah appointment	User melakukan perubahan terhadap data appointment yang dibuatnya.
UC-16	Membatalkan appointment	User membatalkan appointment yang dibuatnya.
UC-17	Melakukan konfirmasi appointment	User melakukan konfirmasi terhadap appointment yang ditujukan padanya.
UC-18	Menampilkan jadwal perkuliahan	User membaca dan mengunduh jadwal perkuliahan berdasarkan ruangan, dosen, dan mahasiswa.

Setiap Use Case memiliki skenario yang merupakan aksi dari *actor* dan reaksi dari sistem. Berikut adalah skenario untuk use case melakukan pemesanan ruangan (UC-07).

Tabel 3.1 SC-07-01

Aksi Actor	Reaksi Sistem
Skenario Normal	
1. Mengklik tombol "Book"	2. Menampilkan form pemesanan class/room
3. Mengisi form pemesanan class/room	
4. Mengklik tombol "Book"	5. Memvalidasi masukan tanggal dari pemesan
	6. Memeriksa apakah

	class/room yang dipesan tersedia
7.	Mencatat pemesanan class/room pada basisdata
8.	Mengirimkan surat pemesanan ke email pemesan
9.	Menampilkan pesan berhasil "Booking is success. Please wait confirmation from administrator."

Tabel 3.2 SC-07-02

Aksi Actor	Reaksi Sistem
Skenario Alternatif 1	
1. Mengklik tombol "Book"	2. Menampilkan form pemesanan class/room
3. Mengisi form pemesanan class/room	
4. Mengklik tombol "Book"	5. Memvalidasi masukan tanggal dari pemesan
	6. Memeriksa apakah class/room yang dipesan tersedia
	7. Sub Room yang dipesan tidak tersedia
	8. Menampilkan pesan berhasil "Sub Room that you select is full."

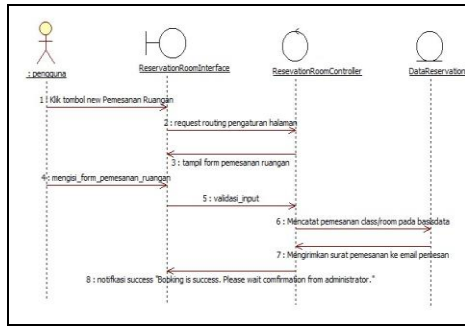
### 3.3 Realisasi Use Case Tahap Analisis

Realisasi Use Case tahap analisis adalah identifikasi kelas analisis dan mendefinisikan *diagram sequence*. Diagram *sequence* merupakan salah satu yang menjelaskan bagaimana suatu operasi itu dilakukan; *message* (pesan) apa yang dikirim dan kapan pelaksanaannya. Diagram ini diatur berdasarkan waktu. Obyek-obyek yang berkaitan dengan proses berjalannya operasi diurutkan dari kiri ke kanan berdasarkan waktu terjadinya dalam pesan yang terurut. Identifikasi kelas analisis adalah sebagai berikut:

Tabel 4 Identifikasi Kelas Analisis UC-07

No	Nama Kelas	Jenis Kelas (Interface, Control, Entity)
1	ReservationRoomInterface	Interface
2	ReservationRoomControl	Control
3	DataReservationRoom	Entity

Diagram sequence yang dihasilkan diperlihatkan pada gambar berikut :



Gambar 3 Sequence Diagram UC-07-01

**3.3 Kelas Analisis**

Daftar kelas analisis beserta jenisnyadijelaskan pada tabel berikut:

Tabel 5 Daftar Kelas Analisis

No	Nama Kelas	Jenis
1	MainPage	Interface
2	UserProfileInterface	Interface
3	UserManagementControl	Control
4	DataPengguna	Entity
5	AllocationRoomInterface	Interface
6	AllocationRoomControl	Control
7	DataAlokasiRuangan	Entity
8	AppointmentInterface	Interface
9	AppointmentControl	Control
10	DataAppointment	Entity
11	JadwalInterface	Interface
12	JadwalControl	Control
13	DataJadwal	Entity

**3.4 Paket Analisis**

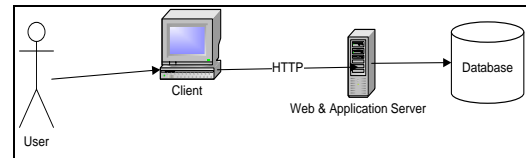
Paker analisis fungsi pada armis dijelaskan pada tabel berikut:

Tabel 6 Paket Analisis

No	Nama Paket	Use Case Terkait
1.	resources managem ent	1. Mengkostumisasi ruangan (UC-01) 2. manajemen user (UC-02) 3. manajemen data akun (UC-03) 4. manajemen data ruangan (UC-04)
2.	allocation	1. mengalokasi ruangan (UC-06) 2. melakukan pemesanan ruangan (UC-07) 3. mengubah pemesanan ruangan (UC-08) 4. membatalkan pemesanan ruangan (UC-09) 5. melakukan konfirmasi pemesanan ruangan (UC-10) 6. mencetak data pemesanan ruangan (UC-11) 7. mencetak data penggunaan ruangan (UC-12)
3.	appointme nt	1. melakukan appointment (UC-14) 2. mengubah appointment (UC-15) 3. membatalkan appointment (UC-16) 4. melakukan konfirmasi appointment (UC-17)
4.	utilization	1. menampilkan utilisasi ruangan (UC-05) 2. menampilkan statistik utilisasi ruangan (UC-13) 3. menampilkan jadwal perkuliahan (UC-18)

**3.5 Deskripsi Arsitektur**

ARMIS dikembangkan menggunakan arsitektur client server dengan teknologi berbasis web. User berinteraksi dengan ARMIS melalui web browser yang terinstall pada client. Client berinteraksi dengan web server melalui protokol HTTP. Web server berperan sebagai perantara antara client dengan application server. Logik – lojik utama ARMIS terletak pada application server, sedangkan data – data yang dibutuhkan ARMIS tersimpan dalam database. Secara umum, deskripsi arsitektur ARMIS dapat dilihat pada gambar di bawah ini.



Gambar 4 Arsitektur ARMIS

**4 Model Perancangan**

Model perancangan berisi identifikasi Elemen WAE, identifikasi kelas perancangan, sequence diagram dan diagram kelas perancangan. Pada tulisan ini, model perancangan hanya untukse UC-07

- Identifikasi Elemen WAE - Logical View

Tabel 7 Identifikasi Elemen WAE-Logical View UC-07

No	Nama Elemen WAE	Lojik	Stereotype	Nama Kelas Analisis Terkait
1	MainPageClient	Client Page	Client Page	MainPage
2	ReservationRoomClient	Client Page	Client Page	ReservationRoomInterface
3	ReservationRoomForm	Html Form	Html Form	ReservationRoomInterface
4	ReservationRoomServer	Server Pages	Server Pages	ReservationRoomController

- Identifikasi Kelas Perancangan

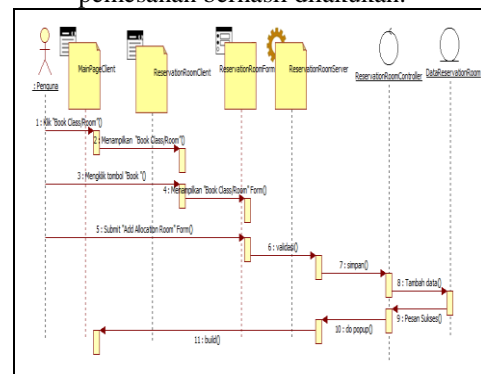
Tabel 8 Identifikasi Kelas Perancangan UC-07

No	Nama Kelas Perancangan	Nama Kelas Analisis Terkait
1	ReservationRoomController	ReservationRoomControl
2	DataReservationRoom	DataReservationRoom

- Sequence Diagram Skenario Normal

ID : SD-P-07-01

Deskripsi : Sequence ini menggambarkan aksi pemesanan saat melakukan pemesanan ruangan dan pemesanan berhasil dilakukan.



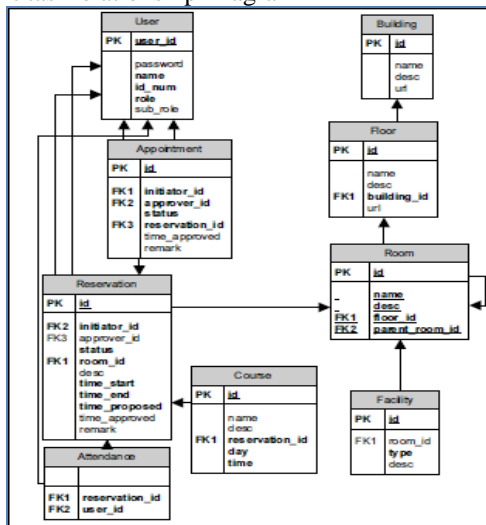
Gambar 5 SD-P-07-01

- Daftar Kelas Perancangan

**Tabel 9 Daftar kelas perancangan**

No	Nama Kelas Perancangan	Nama Kelas Analisis Terkait
1	RoomCustomizationController	RoomCustomizationControl
2	UserManagementController	RoleManagementControl
3	RuanganController	RuanganControl
4	UtilisasiRuanganController	UtilisasiRuanganControl
5	AllocationRoomController	AllocationRoomControl
6	ReservationRoomController	ReservationRoomControl
7	UtilityRoomController	UtilityRoomControl
8	AppointmentController	AppointmentControl
9	JadwalController	JadwalControl

- Entitas Relationship Diagram



Gambar 6 ERD ARMIS

**5. Implementasi**

**5.1 Implementasi Hardware**

ARMIS diimplementasikan pada lingkungan hardware berikut:

1. Dua buah komputer
  - a. Server
    - Ram : 2 GB
    - Processor : Intel Core2 Duo 2.0 GHz
  - b. Klien
    - Ram : 512 MB
    - Processor : 2.53 GHz

**2. Jaringan/LAN**

**5.2 Implementasi Software**

ARMIS diimplementasikan pada lingkungan software berikut:

1. Webserver: Apache Web Server
2. Database Server : MySQL
3. Web Browser : Firefox, IE, Safari

**5.3 Implementasi Kelas**

Berikut adalah kelas-kelas yang telah diimplementasikan.

Tabel 10 Kelas-kelas yang diimplementasikan

No	Nama Kelas	Nama File Fisik	Nama File Executable
1	Allocation	Allocation.php	Allocation.php
2	Appointment	Appointment.php	Appointment.php
3	Mainpage	MainPage.php	MainPage.php
4	Profil	Profil.php	Profil.php
5	RoomManagement	RoomManagement.php	RoomManagement.php
6	RoomUtilization	RoomUtilization.php	RoomUtilization.php
7	Schedule	Schedule.php	Schedule.php
8	UserManagement	UserManagement.php	UserManagement.php

**5.4 Implementasi WAE**

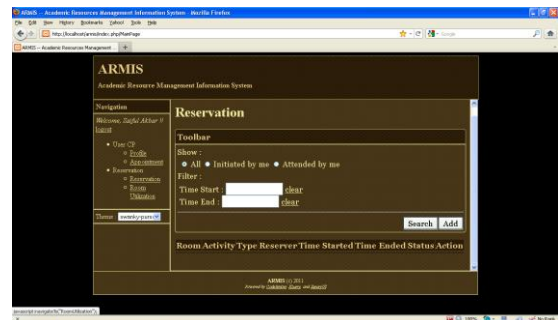
Elemen WAE-Componen View yang merealisasikan Use Case adalah sebagai berikut:

Tabel 11 Realisasi WAE-Componen View terhadap Use Case

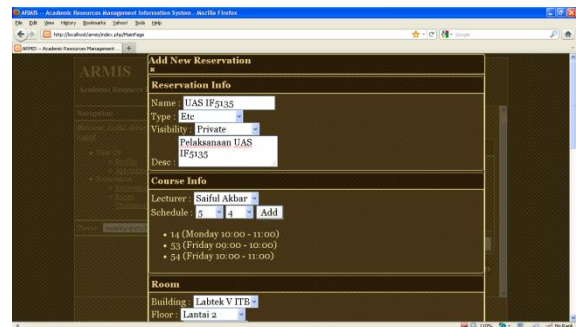
No	WAE-Componen View	Use case
1	Allocation	UC-06,UC-07,UC-08,UC-09,UC-10
2	Appointment	UC-14,UC-15,UC-16,UC17
3	CourseSchedule	UC-18
4	Profile	UC-02
5	RoomManagement	UC-04
6	RoomUtilization	UC-05
7	UserManagement	UC-03

**5.5 Implementasi Antar Muka**

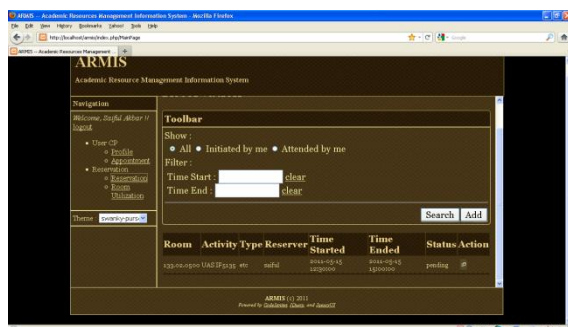
Implementasi antarmuka ARMIS ditampilkan pada gambar-gambar berikut:



Gambar 7 Halaman Pemilihan Reservasi Ruang



Gambar 8 Halaman Pengisian Reservasi Ruang



Gambar 9 Halaman Filtering Alokasi Ruangan

## 6. Kesimpulan

1. RMP dapat diterapkan pada alokasi dan pemantauan ruangan
2. Dengan kelebihan dari arsitektur MVC, RMP dapat diimplementasikan dengan mudah pada resource apapun.

## Referensi

- [1] Bahra, Al, 2006. *Rekayasa Perangkat Lunak*. Yogyakarta: Graha Ilmu.
- [2] Lani, Sidharta, 1995. *Sistem Informasi Manajemen: Pengantar Sistem Informasi Bisnis*. Jakarta: Elex Media Komputindi.
- [3] Widiyanto, Nur, 2010. *Membangun aplikasi Java Enterprise dengan ARSITEKTUR MODEL VIEW CONTROLLER (MVC)*. Yogyakarta: Andi..
- [4] Conallen Jim, 2002. *Building Web Applications with UML Second Edition..* Addison Wesley.
- [5] Booch, G., Rumbaugh, J., Jacobson, I.. 1998. *The Unified Modelling Language User Guide;* Addison Wesley
- [6] Widodo, Prabowo Pudjo, 2011 *Menggunakan UML*. Bandung: Informatika

## Biografi

**Novi Safriadi**, lahir di Pontianak, Indonesia, 3 November 1984. Memperoleh gelar Sarjana Teknik Informatika dari Universitas Tanjungpura, Pontianak, Indonesia, 2007, dan gelar Magister Teknik dari Institut Teknologi Bandung, Indonesia, 2011. Sejak tahun 2008 menjadi dosen di Program Studi Teknik Informatika Fakultas Teknik Universitas Tanjungpura Pontianak. Bidang penelitian saat ini adalah Natural Language Processing.