

## PENGEMBANGAN FITUR E-MATUR DENGAN V-MODEL SEBAGAI ALAT PENGADUAN PUBLIK UNTUK WEBSITE BADAN KEPEGAWAIAN NEGARA

Admaja Dwi Herlambang<sup>1</sup>, Aditya Rachmadi<sup>2</sup>, Kartika Utami<sup>3</sup>, Rahmana Ilmi Hakim<sup>4</sup>, Nurur Rohmah<sup>5</sup>

<sup>1,2,3,4</sup>Jurusan Sistem Informasi, Fakultas Ilmu Komputer, Universitas Brawijaya (10pt)  
Email: <sup>1</sup>herlambang@ub.ac.id, <sup>2</sup>rachmadi.aditya@ub.ac.id, <sup>3</sup>kartika.libra@gmail.com,  
<sup>4</sup>ilmi.rahmana@gmail.com, <sup>5</sup>nururrohmah23@gmail.com

(Naskah masuk: 01 Desember 2018, diterima untuk diterbitkan: 07 Januari 2019)

### Abstrak

Bagian Informasi Kepegawaian dan Komunikasi merupakan sebuah divisi yang ada pada Kantor Regional II Badan Kepegawaian Negara Surabaya. Layanan pengaduan publik oleh Bagian Informasi Kepegawaian dan Komunikasi melalui website memerlukan fitur pengelolaan. Penelitian pengembangan ini menggunakan metode V-Model. Proses pengujian yang dilakukan meliputi *unit testing*, *integration testing*, *system testing*, dan *acceptance testing*. *Unit testing* dilakukan dengan basis *path testing* dan menghasilkan 4 *independent path* dari perhitungan *cyclomatic complexity* pada kasus uji unit *addPIC*. *Integration testing* pada kasus uji fungsi *addPIC* dari *class controller* *admController* dengan fungsi *insertPIC* dari *class model* *admModel* menunjukkan hasil valid. *System testing* dilakukan terhadap 28 fungsi pada sistem, yang menunjukkan hasil valid. *Acceptance testing* dilakukan terhadap *stakeholder* atau *user* menggunakan *black box testing* menunjukkan hasil bahwa *stakeholder* menerima fungsi dalam sistem yang dibuat dan telah sesuai dengan kebutuhan fungsional pada dokumen elisitasi kebutuhan sistem.

**Kata kunci:** layanan pengaduan publik, pengembangan, v-model, website

## E-MATUR DEVELOPMENT USING V-MODEL AS PUBLIC COMPLAINT FEATURE FOR STATE STAFFING AGENCY WEBSITE

### Abstract

*Staff Information and Communication Division is one of division in State Staffing Agency Regional Office II Surabaya. Public complaint service handled by Staff Information and Communication Division through website require management features. This development research used V-Model method. The testing process includes unit testing, integration testing, system testing, and acceptance testing. Unit testing used the base path testing that produces 4 independent paths from the calculation of cyclomatic complexity in the addPIC unit test case. Integration testing in the test case of the addPIC function from controller class admController with the insertPIC function from model class admModel shows valid results. System testing performed on the 28 system functions show a valid result. Acceptance testing conducted against stakeholders or users using black box testing indicates that the stakeholders have received the system functions and had been in accordance with the functional requirements that exist in the document elicitation system requirements.*

**Keywords:** public complaint service, development, v-model, website

### 1. PENDAHULUAN

*E-Government* menjadi sangat populer sejalan dengan perkembangan teknologi informasi dan komunikasi. Berbagai lembaga pemerintahan berlomba untuk mengimplementasikan *E-Government* dengan strategi yang disesuaikan dengan kondisi masing-masing. Kantor Regional II Badan Kepegawaian Negara Surabaya merupakan sebuah instansi pemerintahan yang mengelola administrasi dalam bidang kepegawaian.

Penerapan *E-Government* di dalam Kantor Regional II Badan Kepegawaian Negara Surabaya dalam pengembangannya dengan sistem pelayanan publik belum dapat dilakukan secara maksimal. Pada *website* tersebut terdapat fitur konsultasi yang menunjukkan jumlah pegawai yang telah melakukan konsultasi atau keluhan hanya dibagikan dan ditampilkan dalam halaman *website* namun konsultasi atau keluhan yang telah disampaikan belum tentu diproses oleh pihak pelayanan informasi Kantor Regional II Badan Kepegawaian Negara Surabaya.

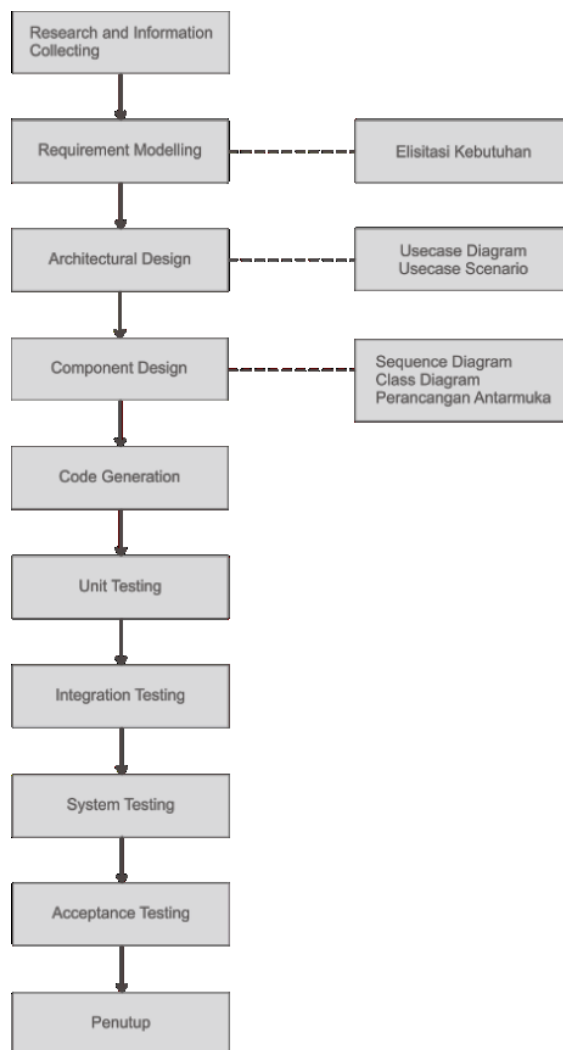
Kelebihan penggunaan sistem informasi dalam konsultasi diantaranya yaitu dapat memberikan dampak kemudahan bagi kepala BKBH untuk dapat melakukan pemantauan terhadap data konsultasi yang dilakukan oleh pengguna secara *online* (Handoko, Lestariningsih, & Ardianto, 2014).

*System Development Life Cycle* (SDLC) atau SDLC adalah sebuah kerangka kerja terstruktur yang terdiri dari proses sekuensial untuk pengembangan sistem informasi. SDLC juga dapat diartikan sebagai rangkaian aktivitas yang dilakukan oleh profesional dan pengguna sistem untuk mengembangkan dan mengimplementasikan sistem (Turban & Volonino, 2011). V-Model merupakan perluasan dari model *waterfall* dalam proses pengembangan perangkat lunak. V-Model menggambarkan hubungan dari aksi-aksi jaminan kualitas dengan komunikasi, pemodelan, dan aktivitas-aktivitas konstruksi pada tahap awal. Fase pada sisi kiri V-Model bergerak ke bawah untuk melakukan spesifikasi kebutuhan, menyaring dasar masalah untuk menghasilkan solusi dengan progresif, semakin detail dan teknis. Saat kode program telah dibuat, maka fase bergerak ke atas menuju sisi kanan V-Model (Pressman, 2010).

Terdapat 9 fase dalam V-Model. Fase pertama adalah fase *Requirement Modelling* yaitu tahap mendefinisikan rencana sistem atau perangkat lunak. *Architectural Design* digunakan untuk memahami dan merancang desain arsitektur sistem. *Component Design*, desain sistem dibagi menjadi beberapa modul yang lebih kecil. *Code Generation* mengimplementasikan desain penyimpanan sistem dan setiap modul yang telah dibangun dalam pengkodean. *Executable Software (Implementation)* dilakukan pengimplementasian terhadap keseluruhan sistem setelah pengkodean (*coding*) selesai dilakukan. *Unit Testing*, pengujian dilakukan untuk mencari kesalahan akibat salah tulis atau kesalahan pemrograman yang disebut dengan *debugging*. *Integration Testing* melakukan pengujian terhadap kumpulan modul yang telah diintegrasikan menjadi beberapa subsistem. *System Testing* dilakukan pengujian terhadap seluruh sistem atau perangkat lunak apakah integrasi pada antar modul sistem telah berjalan dengan baik. *Acceptance Testing* bertujuan untuk menguji apakah *user* telah menerima sistem yang dibangun dilihat dari *requirement* yang telah dipenuhi dan kemudahan penggunaan sistem. Jika terdapat kekurangan pada sistem, maka dapat dilakukan iterasi pada siklus hidup sistem (Nurbaity, 2010).

## 2. METODOLOGI

Tahapan penelitian disajikan pada Gambar 1. Tahap penelitian disesuaikan dengan tahapan yang dilalui dalam V-Model.



Gambar 1. Tahapan Penelitian

### 2.1. Research and Information Collecting

Tahap ini bertujuan untuk mengumpulkan informasi mengenai metode yang digunakan pada penelitian baik itu melalui buku, jurnal, *website* maupun sumber lainnya. Referensi pada penelitian ini yaitu mengenai V-Model yang berperan sebagai metode dalam pengembangan sistem informasi, dan *Research & Development* sebagai metode dalam penulisan laporan penelitian. Terdapat jurnal yang menjadi referensi mengenai metode pada penelitian ini yang berjudul *Research and Development (R&D) Sebagai Salah Satu Model Penelitian Dalam Bidang Pendidikan* oleh Wingate (2015). Wingate (2015) membahas secara singkat tentang metode penelitian dan pengembangan (R&D) serta tahapan yang harus dilakukan dan juga model yang dihasilkan pada setiap tahap. *Research and information collecting* meliputi studi literatur yang berkaitan dengan permasalahan yang dikaji, pengukuran kebutuhan, penelitian dalam skala kecil, dan persiapan untuk melakukan perumusan kerangka kerja penelitian (Wingate, 2015).

## 2.2. Requirement Modelling

Proses identifikasi masalah dibutuhkan untuk mendukung data dalam penelitian. Data ini merupakan data dari hasil wawancara dengan bagian informasi kepegawaian di Kantor Regional II Badan Kepegawaian Negara yang berperan sebagai narasumber atau stakeholder dari sistem yang dibangun. Wawancara ini dilakukan untuk mengidentifikasi kebutuhan fungsional sistem, kemudian hasilnya akan disusun menjadi dokumen elisitasi kebutuhan sistem dimana berisi kebutuhan sistem sesuai keinginan stakeholder dan kebutuhan sistem yang telah disarankan oleh peneliti sebagai solusi dari permasalahan.

## 2.3. Architectural Design

Hasil dari elisitasi kebutuhan sistem akan diterjemahkan ke dalam *use case diagram* yang bertujuan untuk memberikan gambaran bagaimana pengguna sistem dapat berinteraksi dengan sistem. Setelah *use case* dibuat, selanjutnya akan dibuat *use case scenario* yang digunakan untuk menjabarkan atau menjelaskan alur kerja dari setiap *use case* pada *use case diagram*.

## 2.4. Component Design

Hasil dari transformasi *use case diagram* menjadi *use case scenario* akan digunakan untuk membuat *sequence diagram* yang menjelaskan bagaimana alur dari fungsi sistem yang akan dibangun. *Sequence diagram* ini menunjukkan bagaimana pesan yang dikirimkan pada setiap objek untuk menyelesaikan fungsi sistem. Selain itu, pada tahap ini juga dijelaskan mengenai desain arsitektur *database* yang akan didefinisikan dalam bentuk *class diagram*.

## 2.5. Code Generation

Sistem atau perangkat lunak akan dibangun berbasis website dan berdasarkan pada desain arsitektur yang telah didefinisikan sebelumnya. Pada tahap implementasi sistem dilakukan menggunakan pemrograman berorientasi objek dengan bahasa pemrograman PHP dan manajemen basis data sistem MySQL.

## 2.6. Unit Testing

Pada sistem yang telah diimplementasikan, selanjutnya akan dilakukan *unit testing* dengan *basis path testing* yang bertujuan untuk menguji batasan dan jalur independen pada setiap unit program menggunakan logika pemrograman.

## 2.8. Integration Testing

Pengujian dilakukan terhadap setiap modul yang telah diintegrasikan menjadi subsistem-subsistem. Pengujian integrasi dikonsentrasikan untuk mendeteksi kesalahan atau ketidaksesuaian antarmuka atau keluaran dari setiap modul yang telah diintegrasikan ke dalam sistem.

## 2.9. System Testing

Tahap ini pengujian dilakukan terhadap sistem atau perangkat lunak yang telah dibangun menggunakan *validation testing* yang bertujuan untuk memastikan setiap fungsi sistem telah sesuai dengan spesifikasi kebutuhan sistem.

## 2.10. Acceptance Testing

Tahap ini pengujian dilakukan terhadap sistem oleh stakeholder atau penanggung jawab sistem yaitu bagian Informasi Kepegawaian Kantor Regional II Badan Kepegawaian Negara Surabaya untuk mengetahui apakah sistem yang dibuat telah sesuai dengan kebutuhan pengguna pada tahap *requirement modelling*. Apabila sistem yang dibangun telah lolos *acceptance testing* ini, maka pengembangan sistem yang dilakukan telah dinyatakan selesai.

## 3. ANALISIS DAN PERANCANGAN

### 3.1. Analisis Kebutuhan

Dari hasil proses elisitasi kebutuhan perangkat lunak melalui proses wawancara dengan pihak Kantor Regional II Badan Kepegawaian Negara dihasilkan *final draft* elisitasi kebutuhan perangkat lunak sebagai dasar pada pembuatan sistem, selain itu juga terdefinisi 4 aktor yang memiliki peran dalam sistem yaitu admin, PIC, agen, dan klien.

Kebutuhan perangkat lunak yang terdefinisi dari hasil analisis dan spesifikasi kebutuhan yaitu terdapat 28 kebutuhan fungsional, 1 kebutuhan non-fungsional, dan 28 fitur sistem. Pada Tabel 1 disajikan daftar fungsi-fungsi yang harus ada dan telah disesuaikan dengan kebutuhan sistem yang sudah diidentifikasi sebelumnya.

Gambar 2 merupakan *use case diagram* dari hasil analisis kebutuhan sistem E-matur. *Use case diagram* di atas menggambarkan fungsional dari sistem yang dibangun dan identifikasi aktor yang berinteraksi menjalankan sistem. Setiap *use case* yang telah didefinisikan, memiliki *scenario* atau penjelasan mengenai kondisi yang harus dipenuhi sebelum atau sedang menjalankan *use case*, alur kerja aktor dengan sistem, dan kondisi setelah *use case* selesai dilakukan. Berikut merupakan *use case scenario* menambah data PIC yang dituangkan dalam Tabel 2.

### 3.2 Perancangan Sistem

Perancangan sistem bertujuan agar sistem yang dibangun sesuai dengan kebutuhan pengguna. Perancangan sistem ini menggunakan diagram UML

(*Unified Modeling Language*). Gambar 3 merupakan pemodelan *class diagram* sistem E-matur yang berisi penggambaran relasi antar kelas *admController*, *admModel*, *admView*, *picController*, *picModel*, *picView*, *agnController*, *agnModel*, dan *agnView*.

Tabel 1. Kebutuhan dari Sistem

Kode Fungsi	Nama Fungsi
EM-F-001	Melihat <i>Dashboard</i>
EM-F-002	Melihat Data Konsultasi
EM-F-003	Melihat Detail Data Konsultasi
EM-F-004	Menambah Data Konsultasi
EM-F-005	Menghapus Data Konsultasi
EM-F-006	Melihat Data <i>Personal In Charge</i>
EM-F-007	Menambah Data <i>Personal In Charge</i>
EM-F-008	Menghapus Data <i>Personal In Charge</i>
EM-F-009	Memperbarui Data <i>Personal In Charge</i>
EM-F-010	Melihat Data Agen
EM-F-011	Menambah Data Agen
EM-F-012	Menghapus Data Agen
EM-F-013	Memperbarui Data Agen
EM-F-014	Melihat Data Rekap Kinerja Agen
EM-F-015	Menghapus Data Rekap Kinerja Agen
M-F-016	Mencetak Data Rekap Kinerja Agen
EM-F-017	Melihat Data Rekap Kinerja <i>Personal In Charge</i>
EM-F-018	Menghapus Data Rekap Kinerja <i>Personal In Charge</i>
EM-F-019	Mencetak Data Rekap Kinerja <i>Personal In Charge</i>
EM-F-020	Melihat Data Rekap Kinerja Admin
EM-F-021	Mencetak Data Rekap Kinerja Admin
EM-F-022	Memperbarui Profil Admin
EM-F-023	Memperbarui Profil Agen
EM-F-024	Memperbarui Profil <i>Personal In Charge</i>
EM-F-025	Melakukan disposisi Pesan Konsultasi
EM-F-026	Menjawab Pesan Konsultasi
EM-F-027	Mengubah Disposisi Pesan Konsultasi
EM-F-028	Mencari Data Konsultasi

Tabel 2 Use Case Scenario Menambah Data PIC

Item	Deskripsi
Aktor	Admin
Use Case	Menambah Data PIC
Deskripsi	Admin dapat menambahkan data PIC ( <i>Personal In Charge</i> ) ke dalam <i>database</i> .
Pra-Kondisi	Admin telah <i>login</i> ke sistem
Tindakan	<ol style="list-style-type: none"> <li>Admin memilih menu Pengaturan PIC</li> <li>Sistem melakukan permintaan halaman data PIC (<i>Personal In Charge</i>)</li> <li>Admin memilih menu Tambah PIC</li> <li>Sistem melakukan permintaan ke formulir tambah PIC</li> <li>Admin mengisi formulir berisi Nama Lengkap, NIP, Jabatan, Kata Sandi, dan Konfirmasi Kata Sandi</li> <li>Admin memilih tombol Tambah</li> <li>Sistem melakukan permintaan untuk menyimpan data PIC baru ke dalam <i>database</i></li> <li>Sistem menampilkan halaman data PIC (<i>Personal In Charge</i>) dan memuat data PIC yang baru ditambahkan</li> </ol>
Post-Kondisi	Sistem menampilkan halaman data PIC ( <i>Personal In Charge</i> ) dan memuat data PIC yang baru ditambahkan
Alternatif	-

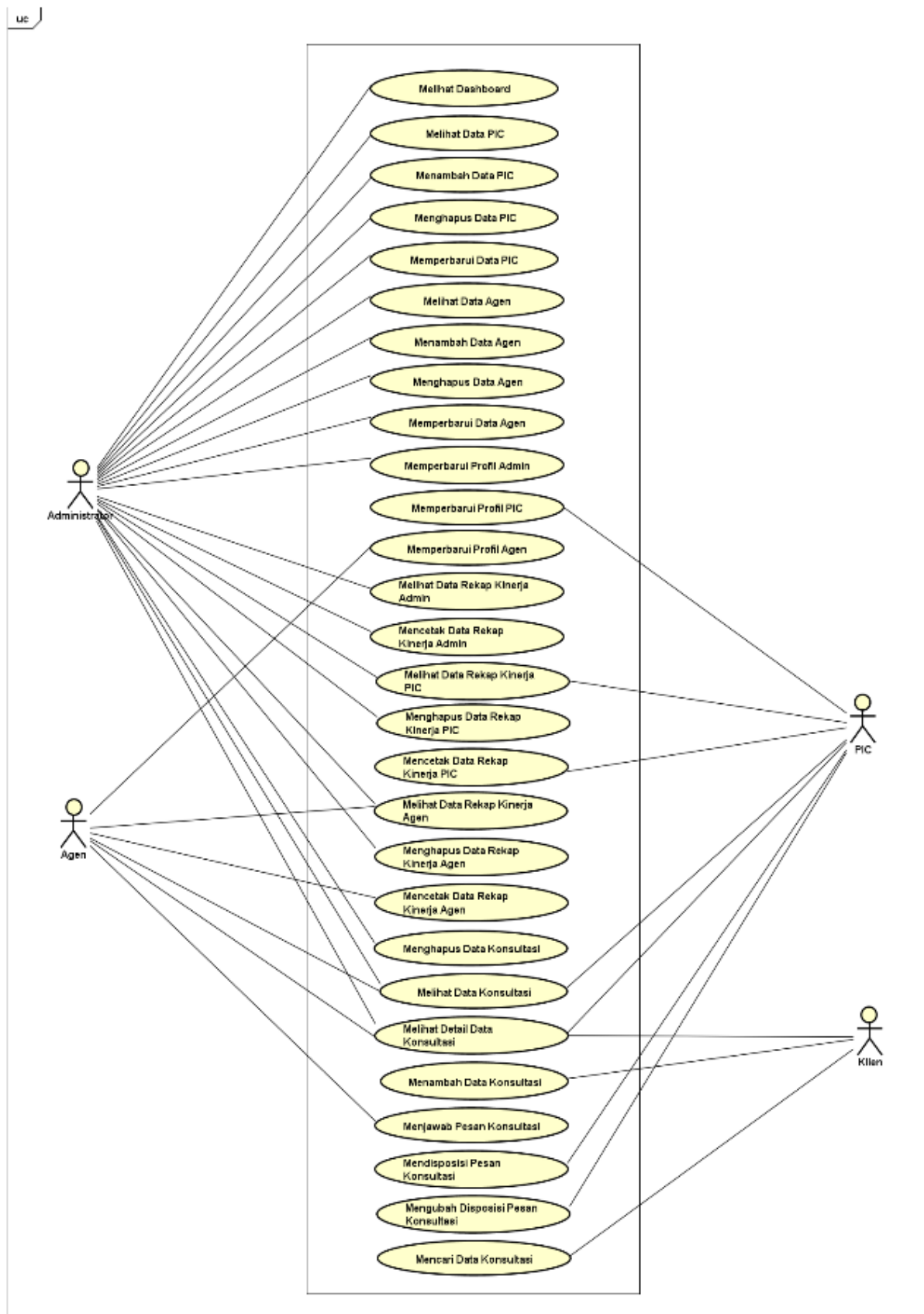


Gambar 4. Physical Data Diagram

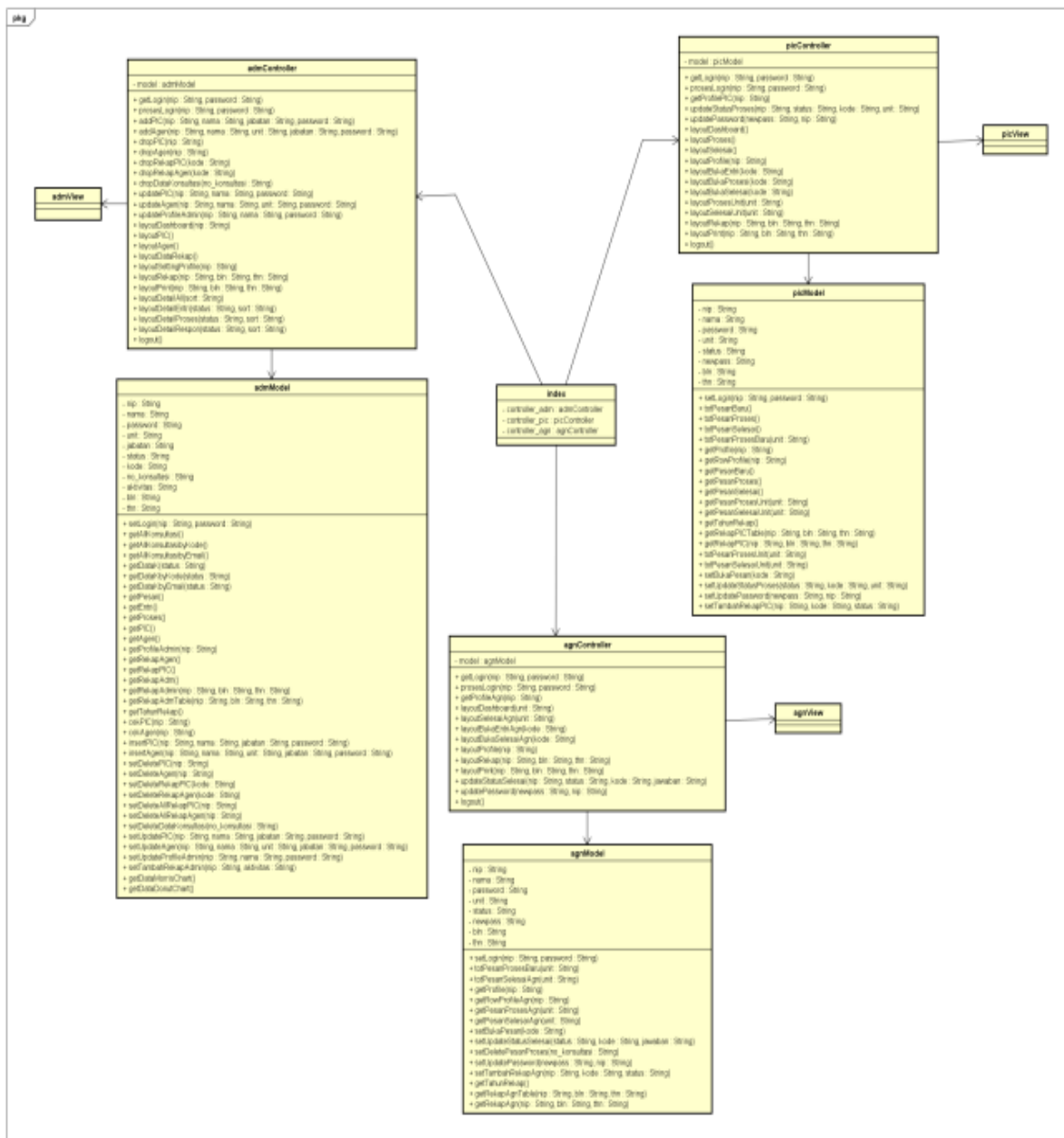
Gambar 4 merupakan diagram perancangan data yang menunjukkan relasi relasi antar tabel. Terdapat 7 tabel antara lain konsultasi, admin, rekap\_admin, pic, rekap\_pic, agen, dan rekap\_agen. Pada relasi tabel ini, tabel admin dan rekap\_admin dihubungkan oleh nip admin, tabel agen dan rekap\_agen dihubungkan oleh nip agen, tabel pic dan rekap\_pic dihubungkan oleh nip pic, tabel konsultasi dan rekap\_agen dihubungkan oleh kode konsultasi, tabel konsultasi dan rekap\_pic dihubungkan oleh kode konsultasi.

Gambar 5 menunjukkan interaksi antar objek ketika sistem menjalankan proses untuk menambah data PIC yang digambarkan ke dalam *Sequence Diagram*. Dimulai dari admin sebagai *user* sistem mengklik menu Pengaturan PIC untuk meminta halaman data PIC (*Personal In Charge*) pada *view*. Kemudian *user* mengklik tombol Tambah PIC untuk meminta halaman tambah data PIC (*Personal In Charge*) pada *view*. Selanjutnya admin mengisi Nama, NIP, Jabatan, Kata Sandi, dan Konfirmasi Kata Sandi, serta mengklik tombol Tambah untuk menyimpan data PIC pada *database*. Kemudian *controller* menjalankan fungsi *addPIC* dengan parameter nip, nama, jabatan dan pass yang didalamnya memanggil fungsi *insertPIC* dengan parameter nip, nama, jabatan dan password pada *model* untuk menyimpan data PIC (*Personal In Charge*) yang ditambahkan. Setelah *model* menyimpan data PIC (*Personal In Charge*) pada *database*, maka seluruh data PIC (*Personal In Charge*) ditampilkan kembali ke *view* melalui *controller*.

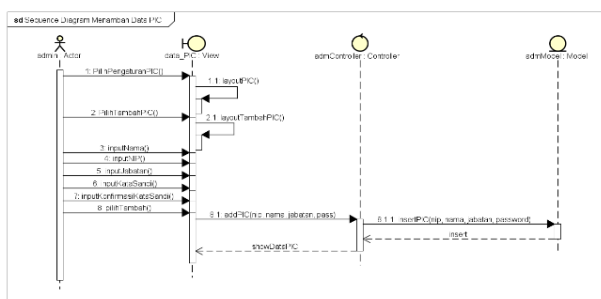
Gambar 6 merupakan implementasi tampilan sistem dalam melakukan penambahan data PIC oleh admin. Terdapat sebuah *form* data PIC yang dibutuhkan diantaranya nama lengkap, NIP, unit kerja, jabatan, kata sandi, dan konfirmasi kata sandi yang ditampilkan dengan model *pop up* di atas halaman daftar data PIC yang sudah terdaftar.



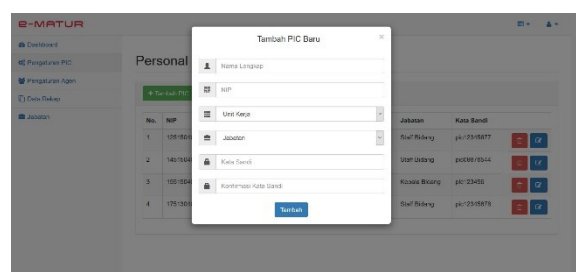
Gambar 2. Use Case Diagram Sistem E-matur



Gambar 3. *Class Diagram* Sistem E-matur



Gambar 5. Sequence Diagram Menambah Data PIC



Gambar 6. Antar muka penambahan data PIC



## 4. HASIL DAN PEMBAHASAN

### 4.1. Implementasi

Tabel 3. Kode Program

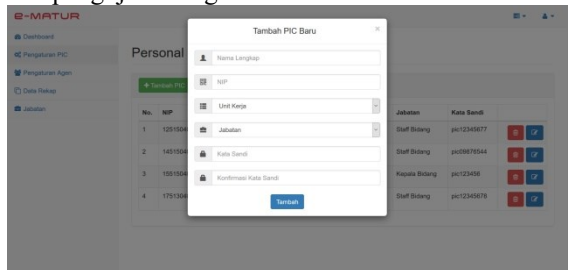
```
public function insertPIC($nip, $nama,
    $jabatan, $password) {
    $this->nip=$nip;
    $this->nama=$nama;
    $this->password=$password;
    $this->jabatan=$jabatan;
    $insert = $this->mysql->
    query("INSERT INTO pic (nip, nama,
    jabatan, password) VALUES ('$this->nip',
    '$this->nama', '$this->jabatan', '$this->
    password')");
    return $insert;
}
```

Hasil perancangan sistem diterjemahkan ke dalam kode program untuk melakukan implementasi sistem E-Matur. Berdasarkan perancangan sistem yang menggunakan OOAD maka implementasi sistem menggunakan pendekatan *Object Oriented Programming* (OOP). Hasil implementasi terdiri dari 3 *subclass controller* yang menggambarkan kemampuan fungsional sistem serta 3 *subclass model* untuk menangani basis data. Cuplikan kode untuk fungsi menambah PIC disajikan pada Tabel 3. Gambar 7 merupakan antarmuka dari implementasi sistem dalam melakukan penambahan PIC. Terdapat 6 kolom yaitu kolom Nama Lengkap, NIP, Unit Kerja, Jabatan, Kata Sandi, dan Konfirmasi Kata Sandi.

### 4.2. Pengujian Sistem

#### 4.2.1. Pengujian White Box

Pengujian *white box* dilakukan menggunakan *basis path testing* untuk mengetahui kompleksitas logika (*cyclomatic complexity*) pada sistem (Pressman, 2010). Dalam penelitian ini, pengujian *white box* yang dilakukan terdiri atas pengujian unit dan pengujian integrasi.



Gambar 7. Antar muka penambahan PIC

Pengujian unit dilakukan pada 1 fungsi yaitu fungsi untuk menambahkan *Personal In Charge* (PIC). Berdasarkan kasus uji yang dilakukan, maka didapatkan jumlah *Region* = 2, *Node* = 11, *Edge* = 13, dan *Predicate Node* = 3. Sehingga nilai *cycomatic complexity* yang didapatkan adalah 4. Hal ini menunjukkan bahwa tingkat resiko terjadinya kesalahan program adalah rendah. Sedangkan untuk

pengujian integrasi juga dilakukan pada fungsi untuk menambahkan *Personal In Charge* (PIC) dan didapatkan hasil pengujian yang valid.

#### 4.2.2. Pengujian Black Box

Pengujian *black box* bertujuan untuk mengetahui apakah sistem informasi yang telah diimplementasikan sesuai dengan kebutuhan yang telah diidentifikasi sebelumnya pada fase analisis kebutuhan. Pengujian *black box* dilakukan pada seluruh *use case* sistem E-Matur yang telah diidentifikasi pada fase analisis kebutuhan, sehingga terdapat 28 kasus uji yang dilakukan. Berdasarkan pada 28 kasus uji, hasil pengujian menunjukkan hasil "Valid" pada seluruh kasus uji. Berikut merupakan hasil pengujian *black box* pada kasus penambahan *Personal In Charge* (PIC) yang dituangkan dalam Tabel 4.

Tabel 4. Hasil Pengujian Menambahkan Data PIC

Nama Kasus Uji	Menambahkan Data <i>Personal In Charge</i> (PIC)
Nomor Kebutuhan	EM-F-007
Tujuan Pengujian	Pengujian dilakukan untuk memastikan sistem dapat menambahkan data <i>Personal In Charge</i> (PIC) baru
Prosedur Pengujian	<ol style="list-style-type: none"> <li>Admin memilih menu Pengaturan PIC</li> <li>Admin menekan tombol Tambah PIC</li> <li>Admin memasukkan data PIC baru</li> <li>Admin menekan tombol Tambah</li> </ol>
Hasil Yang Diharapkan	Sistem dapat menambahkan data <i>Personal In Charge</i> (PIC) baru dan menampilkan halaman data PIC setelah dilakukan penambahan data.
Hasil Pengujian	Sistem menambahkan data <i>Personal In Charge</i> (PIC) baru dan menampilkan halaman data PIC setelah dilakukan penambahan data.
Status Validitas	Valid

## 5. KESIMPULAN

Berdasarkan hasil penelitian, dapat diambil beberapa simpulan, yaitu:

- Proses perancangan sistem menghasilkan diagram *use case* berdasarkan kebutuhan fungsional sistem yang dilakukan pada fase *architectural design*. Terdapat 28 *use case* pada perancangan diagram *use case* dan setiap alur kerja sederhana dari masing-masing *use case* dijelaskan dalam tabel *use case scenario*. Alur data dalam sistem digambarkan dengan *sequence diagram* dan *class diagram* pada fase *component design*. Pada *Class diagram* terdapat 3 kelas *controller* dan 3 kelas *model*. Selanjutnya terdapat perancangan *database* yang digambarkan dalam diagram *physical data model* pada fase *component design* ini.
- Proses implementasi sistem pada fase *code generation* menggunakan PHP sebagai bahasa pemrograman dan MySQL sebagai manajemen

basis data. Implementasi sistem menghasilkan penjelasan deskriptif mengenai fungsi dan potongan program yang diimplementasikan beserta *screenshot* tampilan dari setiap halaman sistem yang menggambarkan kode program yang diimplementasikan.

3. Proses pengujian yang dilakukan meliputi *unit testing*, *integration testing*, *system testing*, dan *acceptance testing*. *Unit testing* dilakukan menggunakan *basis path testing* dan menghasilkan 4 *independent path* melalui perhitungan *cyclomatic complexity* pada kasus uji unit addPIC. *Integration testing* pada kasus uji fungsi addPIC dari class *controller* admController dengan fungsi insertPIC dari class *model* admModel menunjukkan hasil valid. *System testing* dilakukan terhadap 28 fungsi sistem dan menunjukkan hasil yang valid. *Acceptance testing* yang dilakukan terhadap stakeholder atau *user* dengan menggunakan *black box testing* menunjukkan bahwa stakeholder telah menerima fungsi yang dibangun dan telah sesuai dengan kebutuhan fungsional pada dokumen elisitasi kebutuhan sistem.

## DAFTAR PUSTAKA

- HANDOKO, W.T., LESTARININGSIH, E., & ARDHIANTO, E, 2014. Rancang Bangun Sistem Informasi Data Konsultasi Perkara Klien Pada Biro Konsultasi Dan Bantuan Hukum (BKBH) Universitas Stikubank (UNISBANK) Semarang. *Jurnal Dinamika Informatika*, 6(1), p.17-26.
- NURBAITY, S., 2010. *Pengembangan Sistem Informasi Kepegawaian (SIMPEG) Berbasis Web (Studi Kasus: Subbag Useristrasi Kepegawaian Pusat UIN Syarif Hidayatullah Jakarta)*. Universitas Islam Negeri Syarif Hidayatullah. PRESSMAN, R.S. 2010. *Software Engineering A Practitioner's Approach, Seventh Edition*. New York: McGraw-Hill.
- TURBAN, E. & VOLONINO, L., 2011. *Information Technology for Management Improving Strategic and Operational Performance*. New York: John Wiley & Sons, Inc.
- WINGATE, L.M., 2015. *Project Management for Research and Development: Guiding Innovation for positive R&D Outcomes*. Norwegia: CRC Press.