

PERANCANGAN DAN IMPLEMENTASI KONSEP E-MONEY UNTUK SISTEM PEMBAYARAN RETRIBUSI NON TUNAI PADA RUSUNAWA PEMDA D.I. YOGYAKARTA

Rachmad Sanuri, M.Eng

Program Studi Sistem Informasi STMIK EL RAHMA Yogyakarta

e-mail: nicesanuri@gmail.com

Abstract

Rumah Susun Sederhana Sewa atau biasa disebut Rusunawa dibangun dengan menggunakan dana Anggaran Pendapatan dan Belanja Pemerintah Daerah. Dengan fungsi utamanya sebagai hunian dalam membantu masyarakat berpenghasilan rendah untuk mendapatkan hunian layak yang terjangkau. Penghasilan dari pembayaran sewa tersebut adalah sebagai retribusi yang akan menjadi salah satu sumber pendapatan daerah.

Penggunaan e-money akan memberikan sejumlah manfaat dibandingkan dengan menggunakan uang tunai. Metode pembayaran dengan e-money dinilai lebih memberikan keuntungan kenyamanan, keamanan serta mengurangi kecurangan. Penerapan pembayaran non tunai dalam pengelolaan penerimaan retribusi Rusunawa akan memberi keuntungan bagi pengguna sistem, baik dari penghuni rusun dan pemiliknya dalam hal ini Pemda D.I Yogyakarta. Disisi pemilik akan mendapat tambahan manfaat untuk peningkatan kualitas sistem dan tata kelola pendapatan daerah.

Penerapan sistem pembayaran non tunai dengan metode e-money meningkatkan transparansi dan akuntabilitas sebagai salah satu pilar pelaksanaan good governance terkait dengan pelayanan dan peningkatan tata kelola pemerintahan. Dimana akan memberi transparansi tentang aliran retribusi daerah yang akan meningkatkan kepercayaan masyarakat kepada pemerintah sehingga akan memotivasi mereka agar semakin taat dalam pembayaran retribusi.

Keyword : e-money, integrasi sistem, e-retribusi, software enggining

PENDAHULUAN

Penggunaan teknologi dalam kehidupan sehari-hari menjadi sebuah fenomena kehidupan yang terus berkembang saat ini. Secara tidak langsung keadaan ini akan menuntut perusahaan untuk lebih efektif dalam mengelola perusahaan dengan memanfaatkan sumber daya yang dimiliki serta penggunaan teknologi termutakhir yang dapat meningkatkan efisiensi kinerja.

Sistem pembayaran elektronik menjadi salah satu teknologi yang banyak digunakan belakangan ini, sudah banyak masyarakat yang mengandalkan *e-money* sebagai sarana pembayaran dalam bertransaksi. Penggunaan uang tunai yang menimbulkan masalah adanya ancaman keamanan dan rentan terhadap kecurangan, apabila jumlahnya besar sangat sulit untuk dibawa kemana-mana, memicu masyarakat untuk menggunakan *e-money* yang lebih praktis dalam hal waktu maupun tempat.

Penerapan pembayaran non tunai dengan e-money dalam sistem pembayaran retribusi rusunawa akan memberi keuntungan bagi pengguna sistem baik dari penghuni rusun dan pemiliknya dalam hal ini Pemda D.I Yogyakarta. Disisi pemilik akan mendapat tambahan manfaat untuk peningkatan kualitas sistem dan tata kelola pendapatan daerah. Penerapan sistem pembayaran non tunai dengan metode *e-money* meningkatkan transparansi dan akuntabilitas sebagai salah satu pilar pelaksanaan *good governance* terkait dengan pelayanan dan peningkatan tata kelola pemerintahan

Berdasarkan latar belakang tersebut, penelitian ini dilakukan untuk membangun suatu Sistem Pembayaran Non Tunai dengan konsep e-money pada Rusunawa Pemda D.I. Yogyakarta

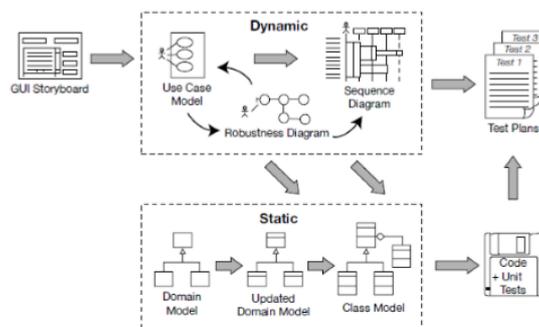
METODE PENELITIAN

1. Metode Pengumpulan Data

- Observasi, yaitu pengumpulan data dengan melakukan pengamatan langsung terhadap proses bisnis objek penelitian untuk memperoleh data yang diperlukan.
- Interview, yaitu pengumpulan data dengan melakukan wawancara atau tanya jawab secara langsung dengan pihak terkait.
- Literatur, yaitu pengumpulan data dengan melakukan studi pustaka mencakup peraturan perundang – undangan, buku-buku teks, diktat, makalah dan artikel.

2. Metode Pengembangan Sistem

Pengembangan sistem meliputi beberapa kegiatan, yaitu analisis kebutuhan perangkat lunak dan perancangan bisnis process perangkat lunak serta perancangan antar muka pengguna (user interface). Proses Analisa dan Desain dilakukan dengan *Use Case Driven Object Modeling* menggunakan UML dengan metode *ICONIX Process*.



Gambar 1 Skema Use Case Driven Object Modeling

Keunggulan utama ICONIX (dan membedakannya dengan metodologi lain) adalah pemanfaatan *robustness diagram* untuk melakukan analisa kehandalan. Dengan analisa kehandalan, akan ditemukan objek-objek baru yang sebelumnya tidak teridentifikasi. Untuk menentukan model dinamis dari sistem yang dibangun, ICONIX menitik beratkan pada penggunaan *sequence diagram*. Dari *sequence diagram* yang dihasilkan akan ditemukan operasi operasi yang harus dimiliki suatu kelas.

Metode ini akan membagi proses menjadi dua aliran kerja yaitu *Static* dan *Dynamic*. Kedua aliran kerja saling berhubungan terus-menerus dan beriterasi. Bagian yang static berhubungan dengan struktur sedangkan dynamic berhubungan dengan behaviour.

HASIL DAN PEMBAHASAN

1. Pendefinisian Kebutuhan Pengembangan

Kegiatan pengumpulan data untuk diolah sesuai dengan kebutuhan fungsional yang nantinya diperlukan di dalam analisa dan perancangan perangkat lunak.

a) Kebutuhan Fungsional

Kebutuhan fungsional sistem ini terdiri atas beberapa fungsi utama yang saling berhubungan dan mendukung satu sama lain, berikut ini fungsi-fungsi dari skenario sistem hasil analisa yang bisa kita definisikan.

Sistem ini harus bisa menangani tatakelola data yang berkaitan dengan rusun milik Pemda DIY. Yaitu data - data yang berkaitan dengan data rusun - rusun milik pemda, data berkaitan dengan kamar yang disewa oleh penghuni rusun, sekaligus data dari semua penghuni yang tinggal disemua rusun.

Sistem juga harus mampu menangani tata kelola data berkaitan dengan administrasi sewa yaitu data harga sewa masing - masing dari semua rusun sekaligus administrasi kontrak sewa yang dilakukan oleh penghuni.

Kemudian yang paling utama sistem harus mampu menangani administrasi data tagihan dan pembayaran kamar semua penghuni. Dimana data tersebut akan terintegrasi dengan sistem keuangan daerah yang dikelola oleh SKPD lain.

Sistem juga harus bisa terintegrasi dengan sistem lain yaitu *core banking system* milik Bank BPD DIY dan juga terintegrasi dengan mesin - mesin anjungan guna melakukan pembayaran sewa.

b) Kebutuhan Non Fungsional

Dalam sistem informasi ini, kebutuhan yang mendukung kelancaran fungsi-fungsi utama dapat didefinisikan pada Tabel

Tabel 1 Kebutuhan Non Fungsional

Parameter	Requirement
<i>Availability</i>	24 jam nonstop, kecuali ada maintenance / perbaikan sistem.
<i>Reliability</i>	Kegagalan yang ditolerir sekitar 3%
<i>Ergonomy</i>	Antar muka dari sistem informasi ini harus user friendly
<i>Portability</i>	Aplikasi ini berjalan pada platform atau sistem operasi apa saja yang mendukung aplikasi berbasis web
<i>Response Time</i>	Tidak lebih dari 3 detik
<i>Safety</i>	Menggunakan secure socket layer dgn sertifikasi.
<i>Security</i>	Login (manajemen user) dan validasi data sangat penting karena menyangkut pembayaran secara online.

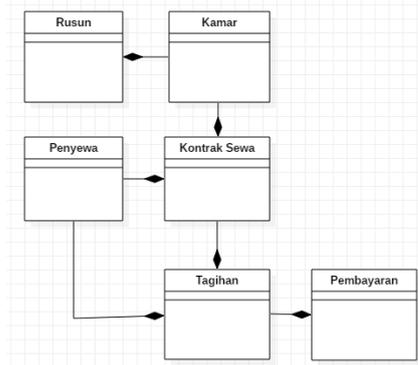
2. Domain Modeling

Domain model akan menjelaskan gambaran besar dari sistem yang akan dibuat. Pada domain model akan muncul obyek – obyek dan bagaimana mereka berelasi satu dengan yang lain. Dari objek–objek tersebut dapat dianalisa lebih lanjut dengan melakukan *generalization* untuk memunculkan *Class*. Berikut hasil penyaringan dari kata - kata unik dengan menghilangkan ambiguitas dari dokumen *functional requirement* dapat dijadikan *class candidate*.

Tabel 2 Kata – Kata Unik Untuk Class Candidate

Rusun	Kamar	Penyewa
Kontrak Sewa	Tagihan	Pembayaran
	Pengguna	

Diagram candidate class dari daftar kata tersebut diatas adalah sebagai berikut.



Gambar 2 First Domain Model

3. GUI prototypes

Tahap *prototypes* atau purwarupa digunakan metode GUI Storyboard. Dengan storyboard membantu designer menghubungkan *actor*, *user stories* dan beberapa penemuan penelitian (*research findings*) untuk mengembangkan *requirements* produk. Kombinasi dari gambar dan kata membuat ide yang kompleks menjadi lebih jelas.

Storyboard juga berguna untuk mengkomunikasikan keputusan desain ketika prototype dibangun oleh beberapa orang dalam waktu singkat, seperti dalam *event design sprint* dan *hackaton*.

Analisa kebutuhan yang didapatkan dari hasil analisa awal pada tahap *requirement* dan tahap *domain modeling* dan dikombinasikan dengan hasil konsultasi dan diskusi dengan pihak - pihak terkait, untuk kemudian disimpulkan alur *storyboard frame* atau modul dari perangkat lunak yang akan dibuat adalah sebagai berikut.



Gambar 3 Frame storyboard

4. Use Case Modeling

Pemodelan use case diperlukan untuk menjawab pertanyaan dasar dari pengembangan yaitu: “Apa yang akan dilakukan oleh pengguna sistem?” Pemodelan use case diperlukan untuk menangkap kebutuhan-kebutuhan pengguna terhadap sistem yang dibangun dengan menggambarkan secara detil seluruh *scenario* yang akan dilakukan pengguna terhadap sistem dan tanggapan yang akan diberikan oleh sistem. Dengan demikian *use case* yang dibuat akan bisa menghubungkan class-class yang akan dirancang dengan *use case*.

a) Refining Use Case

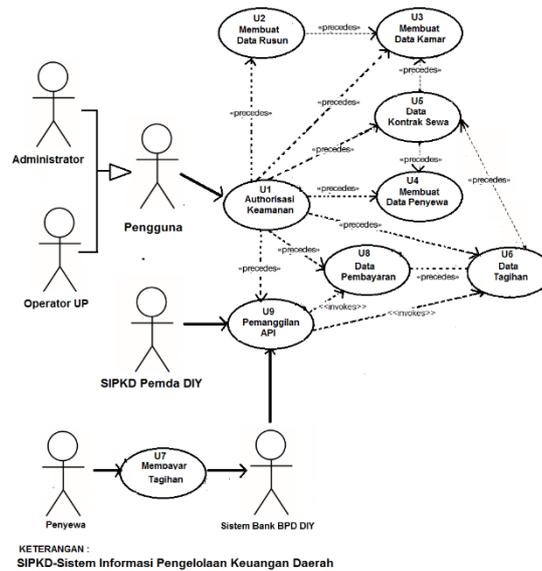
Sebuah use case hampir mirip seperti dokumentasi sistem. Kita perlu menspesifikasikan secara *detail requirement*.

Tabel 3 Use Case Pengguna

Kode Use Case	Kode Requirement	Requirement
U1	R2	Pengguna melakukan autentifikasi sehingga hanya pengguna yang berhak saja yang akan diperbolehkan untuk mengakses sistem.
U2	R3	Pengguna melalui sistem menangani Data Rusun, yaitu mampu melakukan administrasi data rusun yang dimiliki Pemda.
U3	R4	Pengguna menggunakan sistem mengolah data – data kamar yang akan atau telah disewakan dari masing - masing rusun yang dimiliki dan dapat menyajikan laporan yang terkait dengan data kamar
U4	R5	Pengguna menggunakan sistem mengolah data berkaitan dengan data – data semua penghuni kamar dari tiap rusun, penghuni ini adalah merupakan penyewa kamar.
U5	R6	Pengguna menggunakan sistem pengolahan data berkaitan dengan administrasi data kontrak sewa yang diambil dokumen kontrak sewa antara penyewa dengan Pemda DIY.
U6	R7	Pengguna melalui sistem melakukan administrasi tagihan dari semua penyewa. Data sewa yang harus dibayar berdasarkan tagihan milik masing penyewa yang tercantum dalam kontrak sewa.
U8	R9	Pengguna melalui sistem menangani data – data berkaitan dengan administrasi data pembayaran dari para penyewa.

b) Use Case Diagram

Sebuah *use case* hampir mirip seperti dokumentasi sistem. Kita perlu menspesifikasikan seperti apa cara pakai sistem (termasuk respon sistem) sebelum sebuah sistem dibuat. Berikut ini adalah use case diagram berdasarkan *functional requirement* dan memakai *domain model* dari langkah sebelumnya. Dari analisa *requirement* pada langkah sebelumnya didapatkan use case diagram secara keseluruhan sebagai berikut.



Gambar 4 Use Case Diagram

c) Analisa Use Case

Use case mendefinisikan behavioral requirements berdasarkan *functional requirement* dan sumber lainnya. Yaitu untuk menjawab pertanyaan dasar dari pengembangan yaitu: “Apa yang akan dilakukan oleh pengguna sistem?”.

Tabel 4 Use Case Data Pembayaran

Use-case field	Description		
Use case name	Membuat Data Pembayaran	Kode	U8
Actors	Sistem Bank BPD DIY		
Basic Scenario	Infomasi pembayaran adala berasal dari data – data penyewa yang sudah berhasil melakukan pembayaran secara mandiri. Daftar pembayaran ditampilkan berdasar penyaringan berdasar tanggal pada periode tertentu. Dibutuhkan Input data tanggal awal dan tanggal akhir untuk kemudian memilih tombol filter.		
Alternate Scenario	Input tanggal awal harus lebih kecil dari tanggal akhir, jika terjadi sebaliknya maka sistem akan menolak.		

5. Requirements Review

Pada langkah ini untuk memastikan kembali bahwa *use case* dan *domain model* telah dibuat dengan baik. Pengguna sistem perlu dilibatkan untuk memastikan bahwa *use case* (*behavioral requirement*) dan *functional requirement* sesuai dengan yang diharapkan. Bagian terpenting dari sebuah sistem bukanlah seberapa bagus design pattern yang diterapkan di class diagram, tetapi sejauh mana sistem tersebut memberikan keuntungan bagi penggunanya (memenuhi requirements).

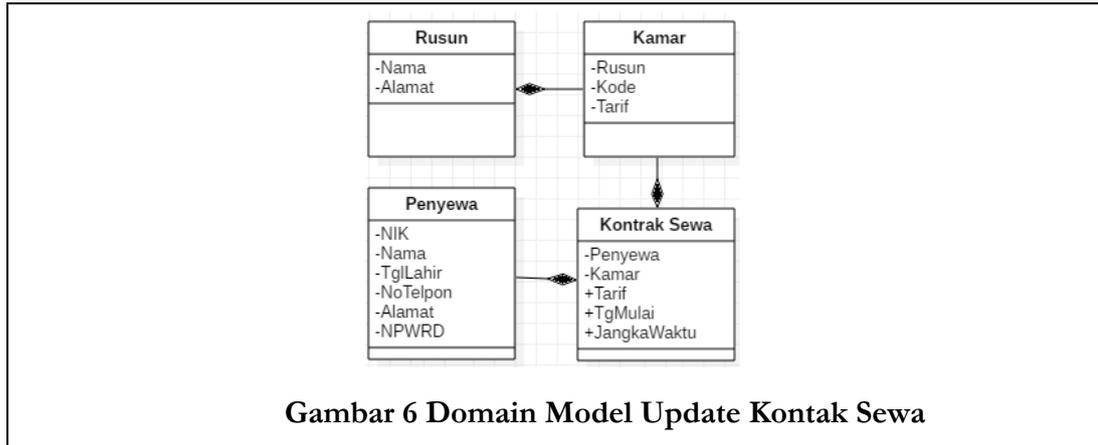
Dalam pembuatan sistem ini dipastikan bahwa semua analisa yang telah dibuat pada langkah – langkah sebelumnya sudah memenuhi kebutuhan dan pengguna.

6. Robustness Analysis

Analisa kehandalan (*robustness analysis*) digunakan untuk menjembatani analisa dan perancangan sistem atau perangkat lunak. *Robustness analysis* diterapkan pada setiap use case yang ada. Pada *Enterprise Architect* tahap *robustness analysis* dapat digambarkan dengan menggunakan Analysis Diagrams. Berikut ini diagram analisa dari tiap – tiap use case.

Tabel 5. Robustness Diagram Sewa Kamar

Kode Robustness Diagram : R05	
Sumber Kode Prototipe : F06	Sumber Kode Use Case: U05
Robustness Diagram	
Gambar 5. Robustness Diagram Data Kontrak Sewa	
Basic Scenario	
<p>Data Kontrak Sewa dikelola oleh pengguna dengan mode administrator dan pengguna mode UP. Prosedur autentifikasi Pengguna harus dilalui untuk masuk kedalam sistem. Input dan update data menggunakan form data kontrak. Data baru harus memilih tombol ‘Data Baru’. Untuk update data didahului dengan memilih data yang akan diupdate melalui data kamar yang tampil pada Daftar Kontrak Sewa. Untuk menyimpan dalam database dipilih tombol “Simpan”. Jika ingin menghapus/menonaktifkan pilih tombol “Hapus”.</p> <p>Entitas data yang dibutuhkan adalah entitas Data Kontrak Sewa. Data yang dibutuhkan adalah data Identitas Penyewa, Kode Kamar berikut Tarif Sewa tiap kamar, Jangka Waktu sewa serta tanggal mulai sewa. Entitas pendukung adalah Data Kamar dan Data Penyewa.</p>	
Alternate Scenario	
<p>Data Kamar dan Data Penyewa harus diinputkan terlebih dahulu jika pilihan Kamar dan Penyewa tidak ada</p>	
Domain Model Update	



7. Preliminary Design Review

Tahapan ini untuk mereview dan memastikan bahwa semua yang telah dibuat sesuai dengan requirement. Tahapan adalah langkah terakhir dimana stackholder terlibat dikarenakan langkah berikutnya melibatkan proses technical.

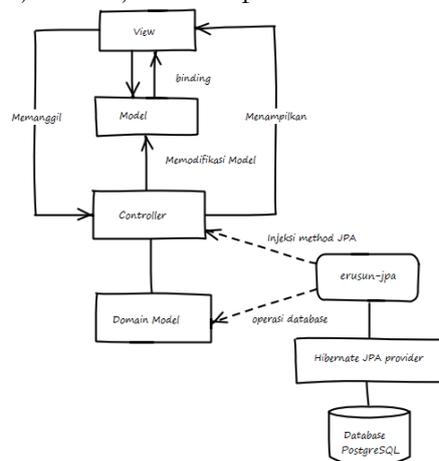
Akan merepotkan bila membiarkan stackholder yang non-technical atau semi-technical mengambil keputusan untuk hal-hal yang bersifat teknis (misalnya framework atau database yang dipakai). Walaupun demikian client masih bisa memberikan usulan saran mengenai tampilan.

8. Technical Architecture

Tahap ini adalah tahap untuk memutuskan berkaitan dengan penentuan *Framework* yang akan digunakan dalam pengembangan aplikasi.

Framework yang digunakan adalah konsep model *Model-View-Controller (MVC)* yang merupa-kan pola umum dalam aplikasi web. Prinsipnya adalah menyusun aplikasi menjadi tiga lapis yaitu.

- **Lapisan Model.** Lapisan yang berkaitan lapisan abstraksi antara algoritma sistem objek – objek data
- **Lapisan View.** Lapisan implementasi terkait dengan antar muka pengguna.
- **Lapisan Controller.** Lapisan yang memiliki fungsi untuk melakukan orkestrasi berkaitan dengan objek – objek dari lapisan Model dan lapisan View.

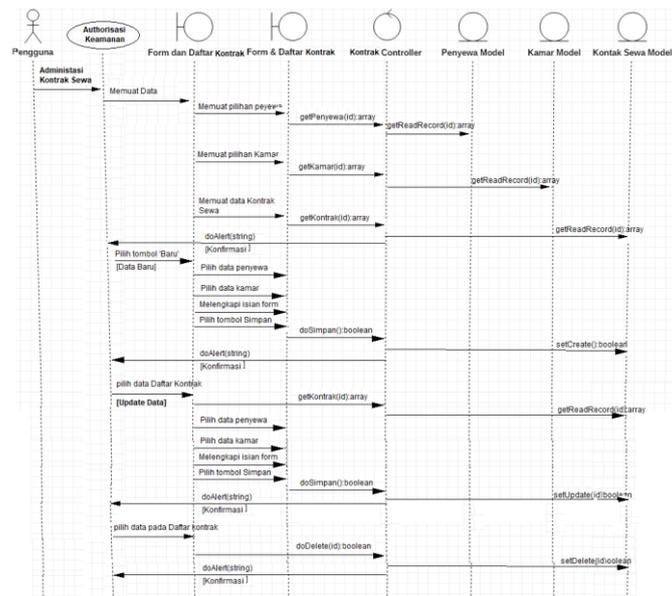


Framework MVC yang digunakan dalam pengembangan aplikasi e-Rusun ini adalah Spring MVC. Dengan metode MVC dapat menciptakan kondisi dimana antara kode dan tampilan merupakan lapisan-lapisan yang terpisah. Hal ini akan membantu desainer grafis dan programmer untuk bekerja bersama-sama tanpa saling mengganggu. Desainer grafis dapat berkonsentrasi pada penciptaan tampilan halaman web yang memikat, sedangkan programmer berkonsentrasi pada penciptaan paket-paket aplikasi yang lengkap.

9. Sequence Diagrams

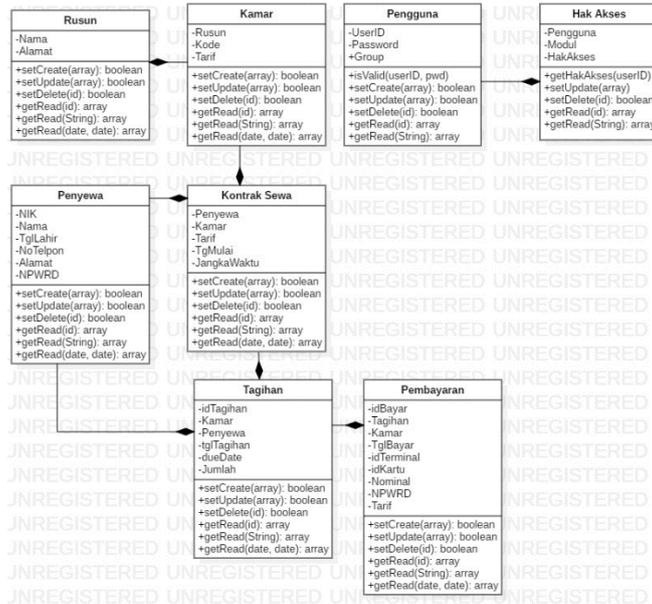
Object oriented pada dasarnya adalah menggabungkan antara data dan operasi ke dalam sebuah entitas. Pada tahap sebelumnya domain model baru berisi data. Oleh sebab itu, dibutuhkan sebuah upaya untuk menemukan operasi untuk domain model. Cara yang digunakan adalah dengan memakai sequence diagram.

Sequence diagram dibuat untuk setiap use case yang ada, berdasarkan hasil robustness analysis. Selama membuat sequence diagram tujuan utamanya adalah menemukan operasi (*behavior*) untuk setiap class yang ada, bukan menunjukkan *step-by-step* operasi secara rinci. Untuk menjaga agar tidak terjebak dalam untuk membuat *flow-chart* yang detail.



Gambar 8. Sequences Diagram Kontrak Sewa

Sequence digram yang didefinisikan memunculkan behavior dari domain model yang terkait dengan proses bisnis e-Rusun demikian pula menimbulkan penambahan operasi pada tiap – tiap class yang ada yang digambarkan dalam dalam gambar berikut

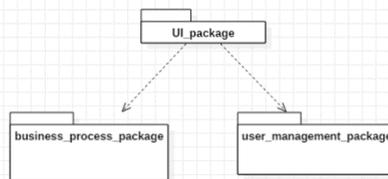


Gambar 9. Update Domain Model

10. Class Package Analysis

Package adalah sarana/cara pengelompokkan dan pengorganisasian kelas-kelas dan interface yang sekelompok menjadi suatu unit tunggal dalam library. Package juga mempengaruhi mekanisme hak akses ke kelas-kelas di dalamnya.

Kelas – kelas dikelompok dalam sebuah package berdasarkan fungsinya. Dengan package maka semua kelas yang ada dalam sistem akan lebih terstruktur dan mudah dalam pemeliharanya. Berikut ini adalah rancangan dari class package dalam aplikasi e-Rusun yang akan dibangun.



Gambar 10. Desain Package

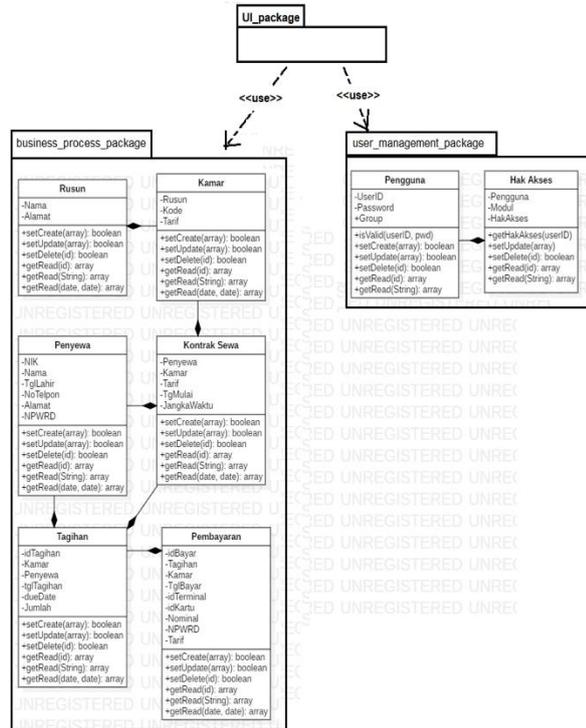
Berdasarkan analisa dan fungsi dari masing – masing kelas didapat 3 (tiga) kelompok fungsi.

1. **UI Package.** Untuk menangani hal – hal yang berhubungan dengan antar muka pengguna.
2. **Business Process Package.** Untuk menangani hal – hal yang berhubungan dengan proses bisnis dari tatakelola Rusunawa.
3. **User Management Package.** Untuk menangani hal – hal yang berhubungan dengan manajemen pengguna.

11. Critical Design Review

Tahap ini adalah review untuk memastikan bahwa tidak ada yang kurang pada sequence diagram. Memastikan pula bahwa setiap class yang ada telah memiliki atribut dan operasi yang didefinisikan secara lengkap (memiliki nama, tipe data, parameter, dsb).

Sekaligus juga mengimplentasikan rancangan package dari tahap sebelumnya. Berikut ini adalah sebagian hasil domain model yang telah dilengkapi.



Gambar 11. Desain Package

12. Implementasi Pengembangan

Implementasi dari rancangan yang telah dibuat sebelumnya. Secara garis besar terbagi atas dua. Yaitu yang pertama adalah implementasi perancangan berorientasi objek (OOD) kedalam lingkungan database relasional. Kemudian yang kedua ada implementasi hasil rancangan kedalam lingkungan pemrograman berorientasi objek.

Implementasi dari rancangan yang telah dibuat sebelumnya. Secara garis besar terbagi atas dua. Yaitu yang pertama adalah implementasi perancangan berorientasi objek (OOD) kedalam lingkungan database relasional. Kemudian yang kedua ada implementasi hasil rancangan kedalam lingkungan pemrograman berorientasi objek.

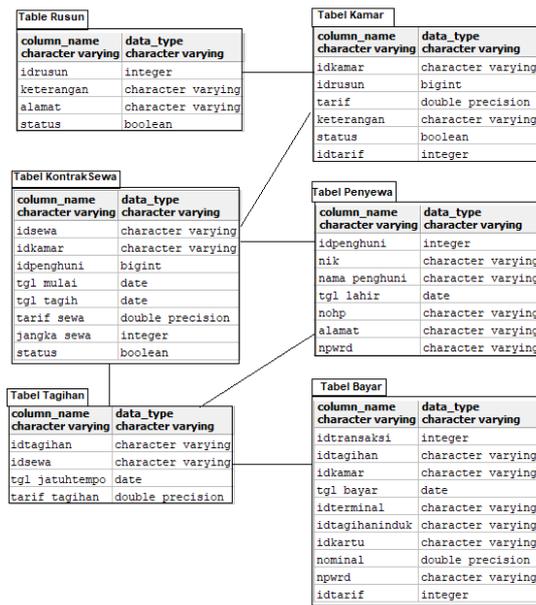
a) Hasil Object Relational Mapping

Hasil implentasi perancangan dalam lingkungan perancangan berorientasi objek (OOD) adalah implementasi **ORM (Object Relational Mapping)** adalah metode pemrograman yang digunakan untuk mengkonversi data dari lingkungan bahasa pemrograman berorientasi objek (OOP) kedalam lingkungan database relasional.

Perangkat lunak ORM yang digunakan dalam pembuatan aplikasi ini adalah JPA. Java Persistence API (**JPA**) adalah bagian dari Java sejak tahun 2006. Versi terbaru **JPA** adalah versi 2.0 (Desember 2009). **JPA** merupakan sebuah spesifikasi yang terdiri atas API untuk menyimpan object (disebut sebagai entity) ke dalam database relasional, termasuk juga spesifikasi *Java Persistence Query Language (JPQL)*.

Software RDBMS yang digunakan dalam aplikasi e-Rusun adalah database PostgreSQL versi 9.6. RDBMS ini adalah software gratis yang sudah terkenal akan kehandalannya terutama untuk menangani data dengan volume besar.

Dari hasil generate JPA ORM dengan menggunakan RDBMS PostgreSQL didapatkan hasil struktur tabel dengan relasi antar tabel sebagai berikut.

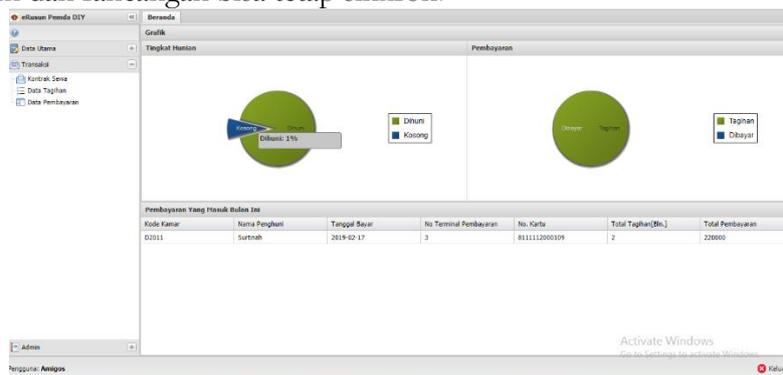


Gambar 12. Relasi Antar Tabel Database

3) Pemrograman dan Hasil

Pada bagian kedua adalah implementasi hasil, berikut ini adalah Implementasi dari rancangan yang telah dibuat sebelumnya kedalam lingkungan Pemrograman Berorientasi Object. Secara garis besar implementasi sistem e-Rusun digunakan administrasi pencatatan berkaitan dengan pengelolaan Rusunawa.

Developer berperan mengubah rancangan (design) menjadi kode program. Karena semua telah direncanakan dan dipikirkan sebelumnya, maka proses coding dapat dianggap sebagai sebuah pembuktian (*test*) bahwa rancangan yang dibuat sudah benar. Terkadang terdapat beberapa hal yang lolos dari perancangan dan baru terungkap saat coding, pada kasus tersebut perubahan pada rancangan harus segera dilakukan sehingga kode program dan rancangan bisa tetap sinkron.



Gambar 13. Form Utama

13. Pengujian (*Integration and Scenario Testing*)

Metode pengujian perangkat lunak yang menguji fungsionalitas aplikasi yang bertentangan dengan struktur internal atau kerja (lihat pengujian white-box).

Pengetahuan khusus dari kode aplikasi / struktur internal dan pengetahuan pemrograman pada umumnya tidak diperlukan. Berikut ini dari daftar pengujian yang dilakukan potongan dari hasil pengujian.

Tabel 5.B Pengujian Sistem dengan Black Box Testing

Input Pengujian	Fungsi	Keluaran Yang Diharapkan
Mengakses aplikasi web e-Rusun	Login Aplikasi eRusun untuk mengakses beranda, modul Rusun, modul Kamar, modul Penyewa, modul Kontrak Sewa, modul Tagihan dan modul Pembayaran	Login berhasil, akses beranda berhasil, akses form Rusun berhasil, akses form Kamar berhasil, dan akses Pembayaran berhasil.

KESIMPULAN

Berdasarkan tahapan penelitian yang telah dilakukan berikut ini beberapa hal yang dapat disimpulkan

1. Implentasi konsep e-money dan pengembangan aplikasi e-Rusun sangat diperlukan karena dengan adanya kedua hal tersebut maka pengelolaan penerimaan retribusi dan pengelolaan Rusunawa menjadi lebih mudah, efektif, efisien dan aman.
2. Dengan fitur Web Service yang terdapat dalam aplikasi e-Rusun memungkinkan pengintegrasian antara Sistem – Sistem lain yang terkait dengan tatakelola administrasi Rusunawa walaupun masing – masing sistem memiliki independensi.

SARAN

Masih dimungkinkan penambahan fitur yang belum tersedia yang berkait dengan tata kelola Rusunawa Pemda DIY yang belum terakomodir dengan mengadopsi teknologi informasi terbaru yang mendukung. Sehingga kemudahan, efektifitas, efisiensi serta keamanan dalam sistem menjadi lebih baik

DAFTAR PUSTAKA

- [1] Anonim, 2018, *ICONIX Process Overview*, <https://iconixprocess.wordpress.com/iconix-process/>, diakses 10 oktober 2018
- Anonim, 2018, *Web Service*, https://en.wikipedia.org/wiki/Web_service, diakses 12 oktober 2018
- [2] Doug Rosenberg and Matt Stephens, 2007, *Use Case Driven Object Modeling with UML Theory and Practice*, second edition, Apress, New York Tim Inisiatif. (2006). *Working Paper* Bank Indonesia. Upaya Meningkatkan Penggunaan Alat
- [3] Pembayaran Non Tunai Melalui Pengembangan E-Money, https://www.bi.go.id/id/publikasi/sistempembayaran/riset/Pages/Kajian_e_money.aspx, diakses 15 oktober 2018