PENGEMBANGAN SISTEM MARKETPLACE TANAH DAN PROPERTI BERBASIS WEB DENGAN PENDEKATAN RAPID APPLICATION DEVELOPMENT (RAD)

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Ivan H. Primananda
NIM: 155150201111335

PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2019

BAB 1 PENDAHULUAN

1.1 Latar belakang

Bisnis properti di Indonesia merupakan salah satu bisnis yang mengalami perkembangan secara signifikan. Hal ini dibuktikan dengan banyaknya pembangunan perumahan baru dengan harga yang bersaing. Selain itu komponen penunjang kepemilikan rumah juga semakin mudah dan menjangkau seluruh lapisan masyarakat, contohnya saja dengan adanya KPR (Kredit Pemilikan Rumah) yang hampir seluruh bank besar di Indonesia menyediakan dana pinjaman dengan beragam variasi pembiayaan.

Tingkat persaingan didunia bisnis saat ini khususnya di Indonesia sangat ketat, tidak terkecuali pada sektor properti yang mengalami perkembangan dan tumbuh dinamis adalah sektor rumah tinggal. Dilihat dari luas lahan produktif, sektor pertanian terus mengalami penurunan dikarenakan meningkatnya alih fungsi lahan produktif menjadi perumahan. "Kalau mengacu pada data 2013, luas baku lahan kita (Indonesia) 7,75 juta hektare. Tetapi sesudah dilakukan pemetaan yang baru, ternyata ada pengurangan luas baku lahan menjadi 7,1 juta hektare." ujar Kepala BPS RI Suharyanto, dikutip Harian Neraca, Selasa (30/10/2018). Selain itu, berdasarkan data Kementerian Agraria dan Tata Ruang (ATR) menunjukkan dalam enam tahun terakhir, 2013-2018, luas baku sawah secara nasional menyusut cukup signifikan yaitu 8,32% atau sekitar 645 ribu hektare.

(Ha)

No.	Jenis Lahan/*Land Type*	Tahun/*Year*					Pertumbuhan/Growth (%)
		2012	2013	2014	2015	2016*)	2016 over 2015
1.	Sawah/*Wetland*	8.132.345,91	8.128.499,00	8.111.583,00	8.092.906,80	8.186.469,65	1,16
	a. Sawah Irigasi/Irrigated Wetland	4.417.581,92	4.817.170,00	4.763.341,00	4.755.054,10	4.781.494,65	0,56
	b. Sawah Non Irigasi/Non Irrigated Wetland	3.714.763,99	3.311.329,00	3.348.252,00	3.337.852,70	3.404.975,00	2,01
2.	Tegal/Kebun/*Dry Field/Garden*	11.947.956,00	11.838.770,00	12.033.776,00	11.861.675,90	11.546.655,70	-2,66
3.	Ladang/Huma/*Shifting Cultivation*	5.262.030,00	5.123.625,00	5.036.409,00	5.190.378,40	5.073.457,40	-2,25
4.	Lahan yang Sementara Tidak Diusahakan/*Temporarily Unused Land*	14.245.408,00	14.162.875,00	11.713.317,00	12.340.270,20	11.957.735,70	-3,10

Gambar 1. Luas lahan pertanian di Indonesia 2012-2016

Sumber : Buku statistika data lahan tahun 2012-2016

hal ini tentunya dipengaruhi oleh pertumbuhan jumlah penduduk pada suatu daerah. Hal seperti ini merupakan suatu peluang bisnis yang bisa dimanfaatkan untuk berinvestasi di bidang tanah dan properti. Selain itu,

pertumbuhan pada sektor properti juga disebabkan banyaknya cara alternatif dalam kepemilikannya yang semakin mudah. Saat ini untuk mempunyai sebuah tanah atau peroperti bangunan tidak harus selalu dibeli langsung secara tunai, akan tetapi dapat juga melalui proses pembiayaan kredit sehingga lebih meringankan calon pembeli untuk melakukan pelunasan pembayaran. Pesatnya bisnis properti ini didorong oleh kebutuhan pokok terhadap papan, disamping pangan serta sandang. Kebutuhan ini merupakan kebutuhan utama yang harus terpenuhi, maka sangatlah wajar bagi seseorang untuk berkeinginan memiliki sebuah hunian sendiri. Tidak dapat dipungkiri bahwa tanah dan properti juga menjadi salah satu alternatif untuk berinvestasi karena harganya yang akan terus meningkat dimasa yang akan datang.

Persaingan untuk mendapatkan harga properti yang murah sebagai investasi berlanjut bisa dibilang sangat ketat, sehingga tidak jarang para investor mulai melirik tanah dan properti dibeberapa daerah timur Indonesia yang dimana harga tanah masih tergolong murah sehingga dapat dijangkau pembiayaannya oleh beragam lapisan masyarakat. Kuatnya pertumbuhan kelas menengah dan daya beli konsumen domestik, mendorong peningkatan bisnis properti di beberapa wilayah Indonesia (Setiawan, et al, 2014). Walaupun berinvestasi di sektor tanah dan properti menjanjikan prospek yang bagus dimasa mendatang, namun pada kenyataannya masih terdapat beberapa lapisan masyarakat yang belum yakin untuk melakukan investasi tanah dan properti.

Kota Kupang adalah ibu kota dan pusat pemerintahan provinsi Nusa Tenggara Timur (NTT), Indonesia. Berdasarkan data dari Badan Pusat Statistik Provinsi NTT, peningkatan jumlah penduduk di NTT mengalami kenaikan sebesar 2,07%. (BPS NTT, 2018) sehingga kebutuhan rumah hunian pun diperkirakan akan terus mengalami peningkatan. Selain jumlah tanah dan hunian kosong yang tergolong banyak, harga properti di NTT juga tergolong murah sehingga dapat dijangkau oleh beragam tingkatan masyarakat. Selain itu, wisata alam pun ikut mempengaruhi ketertarikan investor dalam negeri maupun luar negeri yang ingin membangun apartemen ataupun rumah huni didaerah timur Indonesia khususnya NTT. Menurut Asisten Pemerintahan dan Kesehatan Rakyat Kabupaten Kupang, Nusa Tenggara Timur (NTT), Marthe Bekuliu dikutip dari liputan6.com mengungkapkan, pemerintah daerah (pemda) dan pemerintah pusat (pempus) telah menganggarkan dana pengadaan lahan bagi warga baru (pengungsi yang sudah menjadi Warga Negara Indonesia/WNI). Marthe mengakui, pengadaan lahan di wilayah Kupang (daerah tertinggal) cukup mudah karena harga jual tanah tidak semahal seperti kota-kota besar di Tanah Air yang mencapai jutaan per meternya. Namun pada kejadiannya, bukan saja investor yang ingin mendapatkan tanah dan properti yang murah. Umumnya, pasangan yang baru saja menikah ataupun penduduk yang baru saja mendapat kerjaan di daerah timur ingin mencari rumah tetap yang nyaman untuk ditinggali. Namun kebanyakan dari mereka

masih bingung untuk membeli rumah dengan memanfaatkan kredit pemilikan rumah (KPR) karena belum ada bayangan besaran dana yang harus dibayar setiap bulannya dan juga susahnya harus mensurvei keadaan rumah satu persatu membuat para calon pembeli kewalahan harus kesana kemari mencari alamat dari agen penjual rumah. Selain itu dikutip dari foxnews.com, beberapa masalah yang menyebabkan masyarakat enggan untuk melakukan pembelian dan investasi baik itu investasi tanah kavling maupun properti bangunan adalah yang pertama, ketakutan jika suatu saat nanti tidak mampu untuk membayar cicilan bulanan. Kedua, ketakukan karena tanah atau properti yang dibeli terlalu mahal. Ketiga, ketakutan jika suatu saat nanti terdapat properti yang lebih bagus. Keempat, ketakutan jika penjualan rumah akan menurun dimasa mendatang. Kelima, ketakutan akan subsidi dimasa depan terkait renovasi rumah.

Dengan adanya masalah-masalah tersebut, penerapan konsep marketplace dapat menjadi pilihan yang tepat dan minim resiko. Marketplace adalah suatu tempat di internet dimana banyak pihak berkumpul untuk melakukan proses transaksi jual beli, ada yang ingin mencari suatu barang dan ada pihak lain yang sedang ingin menjual barang. Secara konvensional, konsep marketplace bisa dianalogikan seperti pasar tradisional dimana banyak orang berkumpul di tempat tersebut untuk melakukan transaksi jual beli. Pihak penyedia marketplace bertindak sebagai fasilitator yang mewadahi pertemuan dan transaksi legal antara pihak penjual dan pihak pembeli. Sehingga dengan adanya pengembangan Sistem Marketplace Tanah dan Properti diharapkan dapat memudahkan para calon pembeli maupun investor dalam memilih, mensurvei dan mendapatkan informasi-informasi terkait properti yang akan dibeli. Selain itu, dengan adanya Sistem Marketplace Tanah dan Properti diharapkan semakin banyak masyarakat yang memanfaatkan bisnis investasi tanah tanpa khawatir terkait pembayaran karena dalam pengembanganya, sistem menyediakan fitur simulasi KPR dan perbandingan antar bank yang dapat memberi bayangan kepada calon pembeli maupun investor terkait biaya yang dikeluarkan setiap bulannya untuk melakukan pembayaran kredit properti yang ingin dibeli sehingga memudahkan calon pembeli dalam melakukan pendataan sebagai acuan dalam membeli tanah maupun properti.

1.2 Rumusan masalah

Berdasarkan latar belakang yang telah dipaparkan, dapat dirumuskan permasalahan sebagai berikut :

1. Bagaimana hasil analisis Sistem Marketplace Tanah dan Properti dengan Pendekatan Rapid Application Development (RAD)?
2. Bagaimana hasil rancangan Sistem Marketplace Tanah dan Properti sesuai dengan analisis kebutuhan sistem tersebut?
3. Bagaimana hasil implementasi Sistem Marketplace Tanah dan Properti?
4. Bagaimana hasil pengujian Sistem Marketplace Tanah dan Properti?

1.3 Tujuan

Tujuan yang diharapkan dari penelitian berikut ini adalah sebagai berikut :

1. Mengetahui hasil analisis kebutuhan Sistem Marketplace Tanah dan Properti.
2. Mengetahui hasil rancangan Sistem Marketplace Tanah dan Properti sesuai dengan hasil analisis sistem tersebut.
3. Mengetahui hasil implementasi Sistem Marketplace Tanah dan Properti yang sesuai dengan hasil perancangan sistem tersebut.
4. Menghasilkan Sistem Marketplace Tanah dan Properti berbasis web yang telah melewati tahap pengujian sehingga dapat digunakan sesuai kebutuhan yang telah terdefinisi.

1.4 Manfaat

Manfaat yang diharapkan dari penelitian ini adalah dapat menghasilkan Sistem Marketplace Tanah dan Properti berbasis Web yang memudahkan calon pembeli dan juga investor untuk mendapatkan informasi terkait properti yang ingin dibeli. Sehingga diharapkan peran penting investasi tanah dan properti dalam pembangunan ekonomi Indonesia. Selain itu, dengan adanya Sistem Marketplace Tanah dan Properti diharapkan agar masyarakat tidak perlu khawatir lagi soal pembiayaan terhadap properti karena sebelum membeli tanah dan properti, calon pembeli sudah terlebih dahulu mengetahui besaran biaya perbulan yang harus dibayar untuk melunasi pembiayaan tanah maupun properti.

1.5 Batasan masalah

Batasan dari penelitian ini adalah perangkat lunak yang dihasilkan berbasis website dan dapat diakses oleh perangkat dengan browser yang mendukung. Sistem Marketplace ini dibangun dengan menggunakan bahasa HTML, CSS, PHP dan JavaScript dengan bantuan *framework* Bootstrap. Data sampel dan pengumpulan kebutuhan diambil hanya sebatas tanah dan properti di daerah Nusa Tenggara Timur (NTT) khususnya kota Kupang.

1.6 Sistematika pembahasan

Sistematika penyusunan proposal ini ditujukan untuk memberikan gambaran dan uraian dari laporan proposal secara garis besar meliputi beberapa bab sebagai berikut :

Bab I Pendahuluan

Berisi tentang latar belakang dilaksanakannya penelitian, rumusan masalah yang akan diteliti, tujuan dan manfaat yang ingin dicapai, batasan masalah yang ditentukan dan sistematika penulisan dari Sistem Marketplace Tanah dan Properti.

Bab II Landasan Kepustakaan

Berisi tentang referensi dan dasar teori dilakukannya penelitian yang berhubungan dengan sistem marketplace. Serta metode yang dapat digunakan untuk mendukung Sistem Marketplace Tanah dan Properti.

Bab III Metodologi

Berisi metode yang digunakan dalam penelitian, seperti literatur, analisis kebutuhan, metode pengambilan data, perancangan system, implementasi, pengujian dan pengambilan keputusan dalam Sistem Marketplace Tanah dan Properti.

Bab IV Hasil

Hasil berfungsi untuk melaporkan hasil pelaksanaan metode/teknik penelitian dan menyajikan data yang mendukung hasil penelitian. Penyajian data dan penjelasannya dilakukan secara terurut dan logis menggunakan teks dan ilustrasi lainnya (misalnya, tabel dan gambar).

Bab V Pembahasan

Berfungsi untuk menerjemahkan makna dari hasil yang diperoleh untuk menjawab pertanyaan atau masalah penelitian. Fungsi lainnya adalah untuk menjelaskan pemahaman baru yang didapatkan dari hasil penelitian, yang diharapkan berguna dalam pengembangan keilmuan.

Bab VI Penutup

Bagian ini memuat kesimpulan dan saran terhadap penelitian Sistem Marketplace Tanah dan Properti.

BAB 2 LANDASAN KEPUSTAKAAN

2.1 Rekayasa Perangkat Lunak

Menurut Roger S. Pressman, rekayasa perangkat lunak adalah teknologi yang berlapis. Semua pendekatan rekayasa (termasuk rekayasa perangkat lunak) harus bergantung pada komitmen untuk meningkatkan kualitas sebuah produk. Manajemen peningkatan kualitas diupayakan dapat mendorong peningkatan proses yang berkesinambungan, dan proses yang berkesinambungan tersebut berdampak pada pendekatan yang semakin matang dan dapat mendukung peningkatan kualitas dari sebuah perangkat lunak (Pressman, 2001).

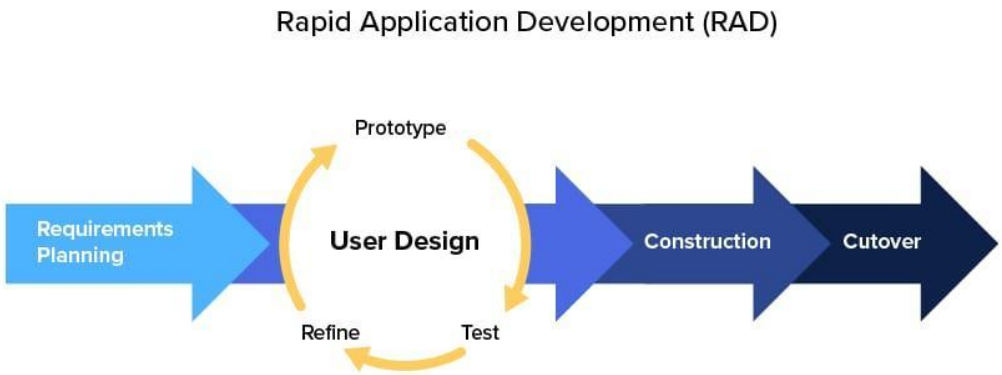
Gambar 2.1 Rekayasa Perangkat Lunak
(Sumber : Pressman, 2001)

2.2 Pengembangan Perangkat Lunak

Dalam pengembangan sebuah perangkat lunak terdapat berbagai macam jenis model yang digunakan. Model-model tersebut mencakup metodologi konvensional dan inovatif, contohnya prototyping, waterfall, incremental, rapid application development (RAD), spiral development, dan extreme programming. Sebuah model proses untuk suatu perangkat lunak dipilih berdasarkan karakteristik project atau aplikasi tersebut (Pressman, 2001).

2.2.1 Rapid Application Development

Rapid Application Development (RAD) adalah sebuah proses perkembangan perangkat lunak sekuensial linier tergolong dalam teknik incremental (bertingkat) yang menekankan siklus perkembangan dalam waktu yang singkat. (Safrian Aswati, 2016)



Gambar 2.2 Metode Rapid Application Development

Sesuai dengan gambar yang ditunjukan pada Gambar 2.2, menurut Riffat Naz and M. N. A. Khan Model RAD memiliki 3 tahapan sebagai berikut:

1. Rencana Kebutuhan (*Requirement Planning*): *User* dan *analyst* melakukan pertemuan untuk mengidentifikasi tujuan dari sistem dan kebutuhan informasi untuk mencapai tujuan. Pada tahap ini merupakan hal terpenting yaitu adanya keterlibatan dari kedua belah pihak.

2. Proses Desain Sistem (*Design System*): Pada tahap ini keaktifan *user* yang terlibat menentukan untuk mencapai tujuan karena pada proses ini melakukan proses desain dan melakukan perbaikan-perbaikan apabila masih terdapat ketidaksesuaian desain antara *user* dan *analyst*. Seorang *user* dapat langsung memberikan komentar apabila terdapat ketidaksesuaian pada desain, merancang sistem dengan mengacu pada dokumentasi kebutuhan *user* yang dibuat pada tahap sebelumnya. Keluaran dari tahapan ini adalah spesifikasi *software* yang meliputi organisasi sistem secara umum, struktur data dan yang lain.

3. Implementasi (*Implementation*): Tahapan ini adalah tahapan *programmer* yang mengembangkan desain suatu program yang telah disetujui oleh *user* dan *analyst*. Sebelum diaplikasikan pada suatu organisasi terlebih dahulu dilakukan proses pengujian terhadap program tersebut apakah ada kesalahan atau tidak. Pada tahap ini *user* biasa memberikan tanggapan akan sistem yang sudah dibuat serta mendapat persetujuan mengenai sistem tersebut. (Riffat Naz and M. N. A. Khan, 2015)

Tujuan dan fungsi dari Rapid Application Development adalah untuk memenuhi kebutuhan pengguna secara efektif dan membutuhkan biaya perawatan yang seminimal mungkin. Kualitas dari sistem didefinisikan sebagai sejauh mana sistem tersebut memenuhi kebutuhan bisnis (atau kebutuhan pengguna) pada saat *project* dimulai.

2.2.2 Unified Modeling Language

Unified Modeling Language (UML) adalah proses mendefinisikan komponen yang akan digunakan untuk membangun sebuah sistem dan untuk menggabungkan semua komponen-komponen interface yang ada (Pressman, 2001). Pemodelan UML terdiri dari beberapa diagram, seperti:

1. *Class Diagram*

   Class diagram digunakan ketika mengembangkan sebuah sistem yang *object oriented* untuk menunjukkan kelas-kelas yang ada dan hubungan antar kelas-kelas tersebut. Cara paling mudah dalam membuat sebuah *class diagram* adalah dengan menuliskan nama kelas ke dalam sebuah kotak. Bisa juga dengan memberikan hubungan antar kelas dengan memberikan garis di antara kelas-kelas tersebut. Contohnya seperti di Gambar 2.3. Sebuah class diagram yang simple menunjukkan ada 2 kelas yang bernama Patient dan Patient Record yang keduanya saling terhubung (Sommerville, 2011).

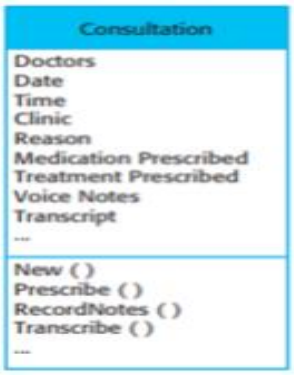
Gambar 2.3 Simple Class Diagram
(Sumber : Sommerville, 2011)

Di dalam level *class diagram* yang lebih detail, maka informasi yang ada di dalam kelas tersebut harus dimasukkan ke dalam kotak *class diagram*. Contohnya, *object* Patient memiliki atribut Address dan didalam *class diagram* harus memiliki operasi yang bernama ChangeAddress. Seperti dicontohkan di dalam Gambar 2.4 dimana:

1. Nama dari *object* diletakkan di bagian paling atas

2. Atribut dari *class* diletakkan di bagian tengah. Di bagian ini juga harus termasuk nama atribut, dan tipenya.

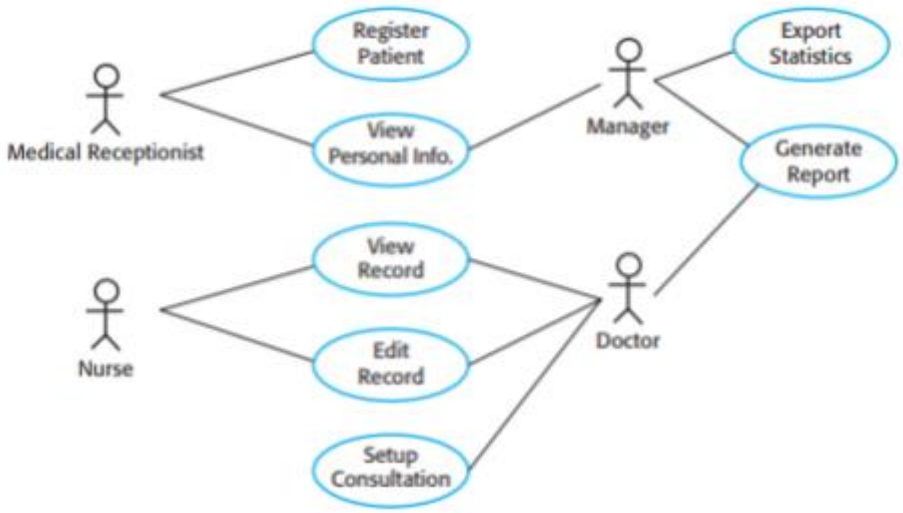
3. Operasi



Gambar 2.4 Class Diagram
(Sumber: Sommerville, 2011)

2. *Use Case Diagram*

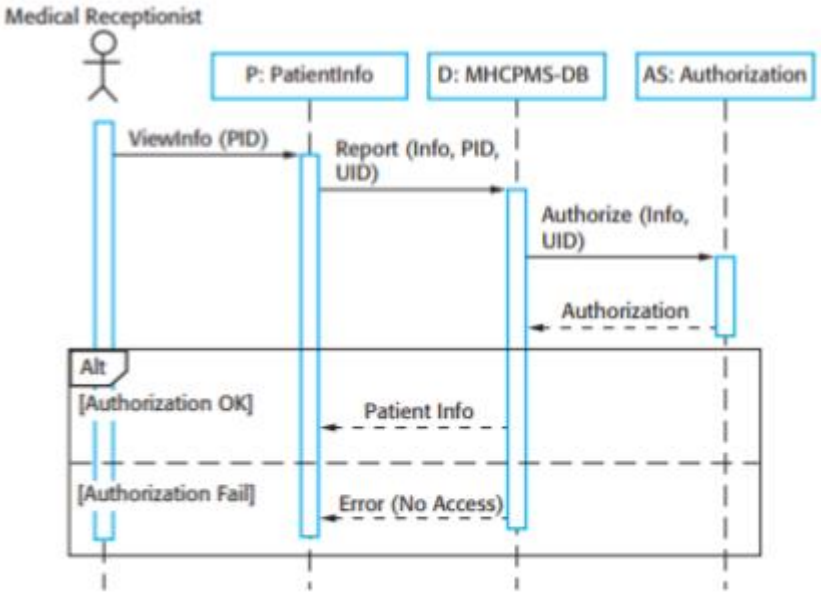
Use Case Diagram adalah sebuah teknik pengumpulan kebutuhan yang pertama kali diperkenalkan dalam objectory method (Jacobson et al., 1993). *Use Case* sekarang sudah menjadi fitur yang fundamental dalam UML. *Use Case* menunjukkan aktor yang terlibat dalam sebuah interaksi dan nama tipe dari interaksi tersebut. Lalu interaksi tersebut dijabarkan ke dalam sebuah informasi yang lebih detail mengenai hubungan antara aktor dengan sistem. Seperti yang ditunjukkan di Gambar 2.5 dimana menunjukkan sebuah *use case* untuk *patient information system* (Sommerville, 2011).



Gambar 2.5 Use Case Diagram
(Sumber : Sommerville, 2011)

3. *Sequence Diagram*

Di dalam UML *sequence diagram* utamanya digunakan untuk memodelkan interaksi antara aktor dengan objek di dalam sebuah sistem dan interaksi antara *object* dengan *object* yang ada dalam sistem tersebut. Seperti namanya *sequence diagram* menunjukkan interaksi yang ada, seperti yang ditunjukkan di Gambar 2.6 dimana pada gambar tersebut ditunjukkan notasi-notasi yang ada dalam *sequence diagram*. Model diagram seperti ini menunjukkan interaksi yang ada diantara View patient information use case, dimana medical receptionist bisa melihat informasi yang dimiliki oleh pasien (Sommerville, 2011).



Gambar 2.6 Sequence Diagram
(Sumber : Sommerville, 2011)

2.3 Pengujian Perangkat Lunak

Pengujian perangkat lunak dilakukan untuk menjamin kualitas perangkat lunak dan merepresentasikan kajian pokok dari spesifikasi, desain dan pengkodean. Pengujian perangkat lunak bertujuan untuk mengetahui sejauh mana perangkat lunak yang diuji telah memenuhi kebutuhan pengguna dan *stakeholder* yang menjadi pengguna sistem (Sommerville, 2011).

2.3.1 Pengujian *Black Box*

Pengujian *black box* adalah metode pengujian yang berfokus pada persyaratan fungsional perangkat lunak (Pressman, 2012). Pengujian ini berusaha menemukan kesalahan dalam kategori sebagai berikut:

1. Fungsi-fungsi yang tidak benar atau hilang.
2. Kesalahan *interface*.
3. Kesalahan dalam struktur data atau akses database eksternal.
4. Kesalahan kinerja.

2.3.2 Pengujian *White Box*

Pengujian *white box* adalah metode desain *test case* yang menggunakan struktur kontrol desain prosedural untuk memperoleh *test case*. Dengan menggunakan metode pengujian ini akan didapatkan *test case* yang:

1. Memberikan jaminan bahwa semua jalur independen pada suatu modul telah digunakan paling tidak satu kali.
2. Menggunakan semua keputusan logis pada sisi *true* dan *false*.

3. Mengeksekusi semua *looping* pada batasan tertentu.

4. Menggunakan struktur data internal yang menjamin validitasnya.

2.4 Marketplace

Marketplace merupakan media online berbasis internet (web-based) tempat melakukan kegiatan bisnis dan transaksi antara pembeli dan penjual. Pembeli dapat mencari supplier sebanyak mungkin dengan kriteria yang diinginkan, sehingga memperoleh sesuai harga pasar (Angga, K.P et al,2017).

Marketplace adalah wadah komunitas bisnis interaktif secara elektronik yang menyediakan pasar dimana perusahaan dapat ambil andil dalam B2B *e-Commerce* dan atau kegiatan *e-Business* lain. Dari beberapa definisi tersebut, dapat dikatakan bahwa marketplace merupakan sebuah wadah pemasaran produk secara elektronik yang mempertemukan banyak penjual dan pembeli untuk saling bertransaksi (Brunn Peter, Jensen Martin, Skovgaard Jakob. 2002).

Beberapa komponen yang menunjang sebuah *marketplace* itu sendiri, yaitu:

1. Pelanggan berasal dari seluruh dunia, yang *surf* melalui Web.

2. Penjual jutaan toko ada di Web, iklan dan menawarkan barang yang sangat bervariasi.

3. Barang dan jasa mempunyai tipe fisik dan digital. Digital produk ini adalah barang yang dibuah menjadi format digital dan di kirim melalui Internet.

4. Infrastruktur *Network, hardware, software* dan lainnya adalah infrastuktur yang harus disiapkan dalam menjalankan *marketplace*.

5. *Front-end* penjual dan pembeli berhubungan dalam *marketplace* melalui sebuah *front-end. Front-end* ini berisi portal penjual, catalog elektronik, *shopping cart*, mesin pencari, mesin lelang,

6. *Back-end* Aktivitas yang berhubungan dengan pemesanan dan pemenuhan pemesanan, manajemen persediaan, pembelian dari pemasok, akuntansi dan finansial, proses pembayaran, pengepakan, dan pengiriman dilakukan di *backend*.

7. *Intermediaries* pihak ke tiga yang mengoperasikan antara penjual dan pembeli. Kebanyakan dioperasikan secara komputerisasi.

2.5 Website

Website merupakan fasilitas internet yang menghubungkan dokumen dalam lingkup lokal maupun jarak jauh. Dokumen pada *website* disebut *web page* dan *link* dalam *website* memungkinkan pengguna bisa berpindah dari satu *page* ke *page* lain (*hypertext*), baik diantara *page* yang disimpan dalam server yang sama maupun server di seluruh dunia. *Pages* diakses dan dibaca melalui browser seperti Netscape Navigator, Internet Explorer, Mozilla Firefox, Google Chrome dan aplikasi browser lainnya (Hakim Lukmanul, 2004).

Website adalah kumpulan halaman web yang saling terhubung dan file-filenya saling terkait. Web terdiri dari *page* atau halaman, dan kumpulan halaman yang dinamakan *homepage*. *Homepage* berada pada posisi teratas, dengan halaman-halaman terkait berada di bawahnya. Biasanya setiap halaman di bawah homepage disebut *child page*, yang berisi *hyperlink* ke halaman lain dalam web (Gregorius, 2000:30).

2.6 Framework Laravel

Laravel merupakan frameword untuk mengembangkan aplikasi web yang menggunakan pola model-view-controller. Dalam pemakaiannya laravel meggunakan bahasa pemrograman web yaitu PHP. Tujuan utama laravel sendiri adalah untuk menunjang kualitas dari sebuah perangkat lunak yang berguna untuk mengurangi cost pengembangan dan perawatan perangkat lunak. Fitur-fitur yang dimiliki laravel juga sangat berperan dalam mempercepat waktu mengimplementasikannya. (McCool, 2012).

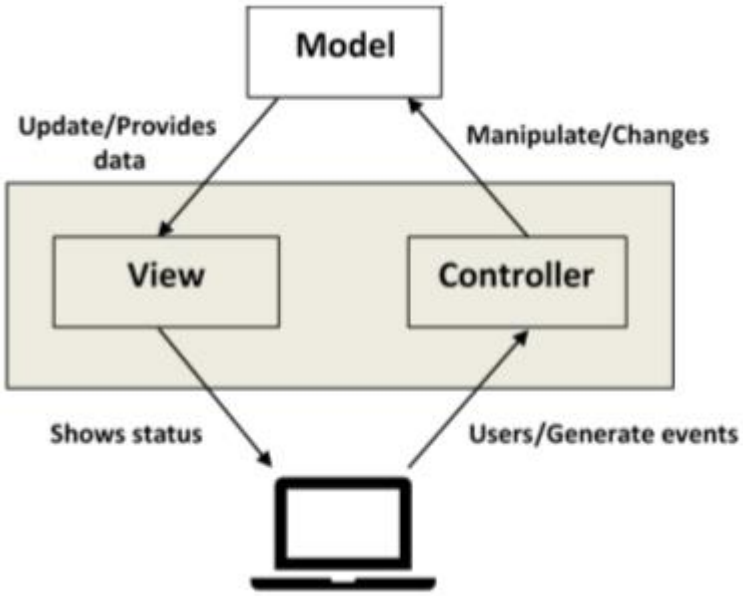
Karena pola yang digunakan pada laravel dalah MVC atau model-viewcontroller, maka dalam struktur laravel tersebut dilakukan pemisahan antara model, view, dan controller. Tujuan dari pemisahan model, view, dan controller adalah untuk menurunkan ikatan dari tiap komponen sehingga perubahanperubahan yang terjadi tidak memiliki dampak yang terlalu besar. (Caytiles and Lee, 2014) dalam penelitiannya menyatakan bahwa MVC merupakan pola yang mana hasil dari penggunaannya akan mendapat sebuah aplikasi perangkat yang terbagi menjadi tiga objek diantarnya:

Model berperan dalam menangkap aksi dari domain permasalahan maupun antarmuka. Digunakannya juga untuk melakukan manipulasi data atau mengelola data beserta logika data dan aturan-aturan aplikasi.

View memproyeksikan keluaran apapun dalam bentuk apapun ke antarmuka diantarnya teks, checkbox, grafik atau diagram, dan bentuk-bentuk lain yang serupa. Sehingga dalam satu view dapat terjadi penggabungan dari beberapa elemen.

Controller berperan sebagai penghubung antara model dan view. Dimana controller mengatur seluruh logika melalui inputan yang dilalukan user dan mengatur logika-logika dalam memanipulasi data.

Pemaparan model MVC dapat dilihat pada Gambar 2.7.



Gambar 2.7 Relasi Komponen MVC
Sumber: (Caytiles dan Lee, 2014)

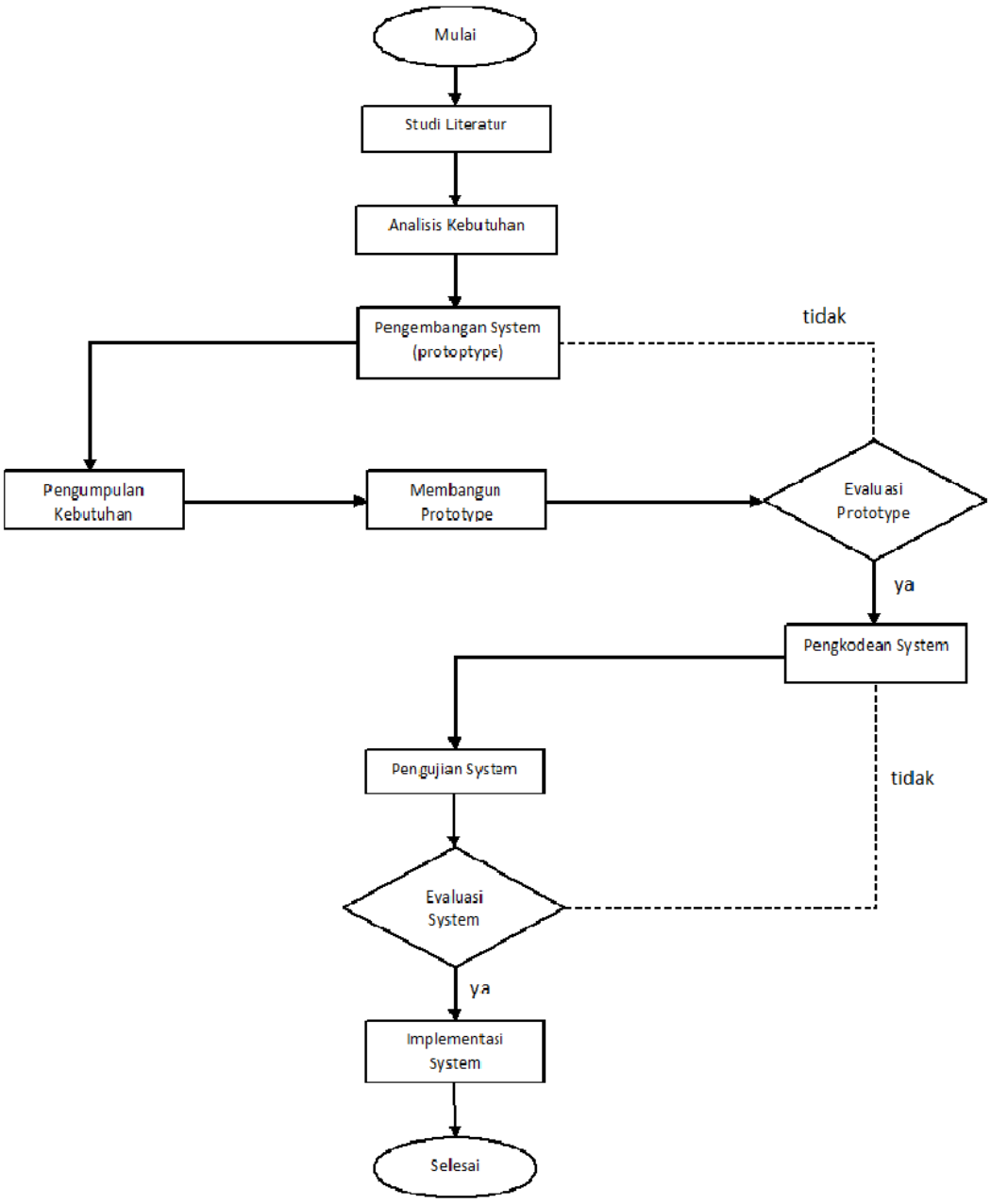
Framework merupakan perangkat lunak yang mulai menjadi pilihan untuk membuat suatu aplikasi (Andresta, 2008). Kemudahan-kemudahan yang diberikan menarik orang-orang untuk menggunakannya. Hal ini tidak terlepas dari tingkat efektifitas dan efisiensinya yang lebih baik dalam proses pengembangan suatu perangkat lunak.

Framework adalah sekumpulan perintah/fungsi dasar yang dapat membantu dalam menyelesaikan proses-proses yang lebih kompleks (Visikom, 2009).

Framework adalah suatu aplikasi yang dapat digunakan ulang untuk membuat bermacam-macam aplikasi (Jhonson, 2009). Framework merupakan kumpulan beberapa kelas abstrak pada domain tertentu sehingga pengembang yang menggunakan *framework* harus melengkapi kelas abstrak tersebut menjadi perangkat lunak yang diinginkan (Andresta, 2008).

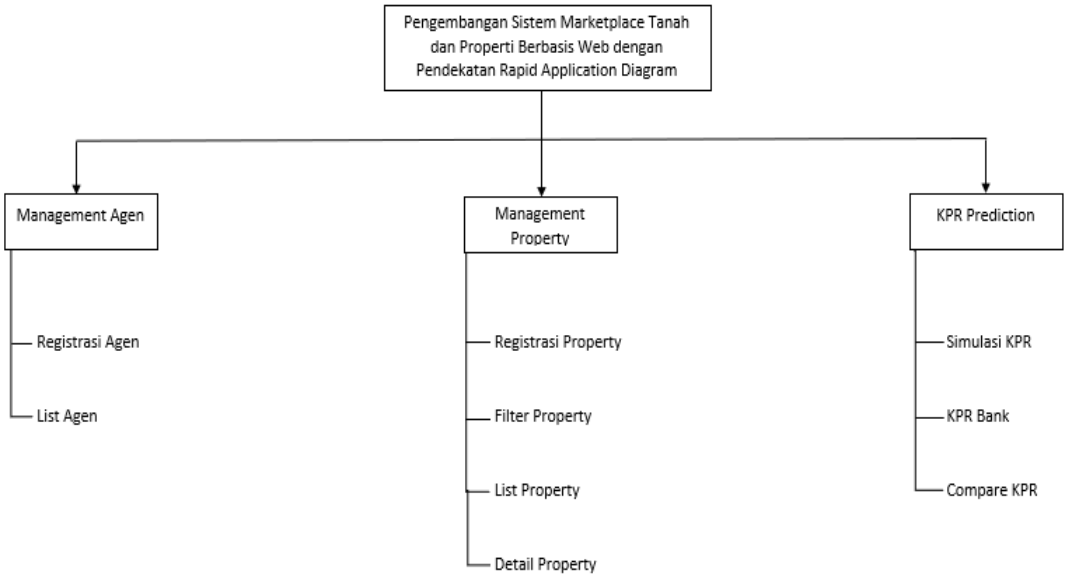
BAB 3 METODOLOGI

Pada bagian metodologi akan dibahas mengenai langlah-langkah atau metode yang digunakan dalam pembuatan Sistem Marketplace Tanah dan Properti. Diagram alir metodologi ditunjukkan pada Gambar 3.1.



Gambar 3.1 Diagram Alir Metodologi

Selain diagram alir diatas, terdapat *Work Breakdown Structure* (WBS) yang dibuat berdasarkan metode *Rapid Application Development* (RAD) yang digunakan sebagai acuan dari sistem yang dibangun dapat dilihat pada Gambar 3.2.



Gambar 3.2 Work Breakdown Structure (WBS) Pengembangan Sistem Marketplace Tanah dan Properti

3.1 Studi Literatur

Metode penelitian ini memerlukan tahapan studi literatur yang digunakan sebagai acuan untuk membuat proposal skripsi ini beserta pengembangan aplikasinya. Dasar teori yang berhubungan dalam pembuatan antara lain:

1. Rekayasa Perangkat Lunak
2. Pengembangan Perangkat Lunak
3. Pengujian Perangkat Lunak
4. *Marketplace*
5. *Framework Laravel*

Studi literatur diguakan untuk menunjang dan mendukung dalam pembuatan proposal skripsi ini. Sumber yang digunakan adalah jurnal, laporan ilmiah, serta bantuan search engine yang ada di internet agar memperkuat dasar teori terkait dengan pembuatan Sistem Marketplace Tanah dan Properti.

3.2 Analisis Kebutuhan

Tahap analisis kebutuhan pada penelitian ini menyesuaikan dengan tahap *requirements planning* pada metode *Rapid Application Development* (RAD). Pada tahap ini dilakukan proses analisis tujuan dan kebutuhan dari aplikasi atau sistem. Analisis kebutuhan dilakukan terhadap pengguna dan pengembang dengan

menggali masalah yang ada kemudian menentukan kebutuhan untuk sistem. Setelah kebutuhan sistem didapatkan, langkah berikutnya adalah memastikan kebutuhan sistem telah sesuai dengan persetujuan dari pengguna dan pengembang. Dengan memastikan analisis kebutuhan yang ada sudah sesuai untuk menghindari adanya perubahan dan miskomunikasi pada tahap berikutnya.

3.3 Perancangan Sistem

Tahap perancangan sistem pada penelitian ini menyesuaikan dengan tahap user design pada metode *Rapid Application Development* (RAD). Pada tahap ini dilakukan proses desain dan perbaikan jika terdapat ketidaksesuaian dengan analisis kebutuhan yang terdapat sebelumnya. Dengan mengadopsi teknik perancangan pada metode *Rapid Application Development* (RAD) yaitu melakukan pembuatan prototype dari aplikasi yang dibangun untuk menampilkan hasil dan fitur kepada pengguna. Melalui prototype yang disajikan, pengguna melakukan ujicoba (test), kemudian jika terdapat ketidaksesuaian dilakukan perbaikan terhadap aplikasi tersebut agar dapat dilanjutkan ke tahap berikutnya.

3.4 Implementasi

Tahap implementasi pada penelitian ini menyesuaikan dengan tahap construction pada metode *Rapid Application Development* (RAD). Pada tahap ini, *prototype* yang sebelumnya dibuat pada perancangan sistem dikembangkan lebih lanjut dengan perbaikan-perbaikan yang telah disesuaikan dengan kebutuhan sistem dan pengguna. Pengembangan prototype dilakukan per fitur sesuai dengan *Work Breakdown Structure* (WBS) dari sistem yang dibangun. Pengembangan *prototype* merupakan proses coding sistem yang diwujudkan dengan berbasis web.

3.5 Pengujian

Tahap pengujian pada penelitian ini menyesuaikan dengan tahap cutover pada metode *Rapid Application Development* (RAD). Tahap ini merupakan kelanjutan dari tahap implementasi sebelumnya, dimana sistem yang telah dikembangkan dilakukan pengujian untuk memastikan semua unit sistem bekerja dengan baik dan sistem yang dihasilkan sesuai ekspektasi serta memuaskan pengguna. Pengujian yang dilakukan terhadap sistem menggunakan pengujian *black box* dan *white box*.

3.6 Kesimpulan dan Saran

Penarikan kesimpulan dilakukan setelah semua tahapan sebelumnya selesai terlaksana. Kesimpulan akhir diambil dari hasil pengujian dan analisis metode yang digunakan, kemudian pemberian saran yang berguna sebagai rekomendasi untuk perbaikan kesalahan yang terjadi dan sebagai tolak ukur bagi penelitian di kemudian hari.

BAB 4 ANALISIS KEBUTUHAN

Analisis kebutuhan adalah tahap awal dalam proses pengembangan suatu perangkat lunak. Analisis ini digunakan untuk mengetahui kebutuhan-kebutuhan yang diperlukan oleh Sistem Marketplace Property dan Tanah. Teknik yang digunakan penulis untuk melakukan analisis kebutuhan dalam penelitian ini adalah teknik wawancara dan kuesioner. Pada analisis kebutuhan ini juga penulis akan menjelaskan aktor-aktor yang terlibat dalam sistem, apa saja yang bisa dilakukan oleh aktor-aktor tersebut, dan juga interaksi antar aktor di dalam sistem.

4.1 Deskripsi Sistem

Sistem Marketplace merupakan suatu sistem yang dibuat dengan tujuan untuk memudahkan para agent maupun penjual property dalam memasarkan property. Selain itu adanya sistem marketplace memudahkan para calon pembeli yang ingin berinvestasi ataupun membutuhkan sebuah bangunan siap huni untuk ditinggali. Sistem marketplace juga dibuat agar para agent dapat dengan mudah dalam melakukan registrasi tanpa harus melalui birokrasi yang dapat menyusahkan calon pendaftar. Simulasi KPR menjadi salah satu fitur yang ada dalam Sistem Marketplace Property dan Tanah, yang diharapkan dapat membantu calon pembeli dalam memperhitungkan estimasi biaya yang dibutuhkan untuk mendapatkan property yang diinginkan.

4.2 Identifikasi Aktor

Menjelaskan tentang peran dari aktor yang terlibat dalam sistem. Aktor-aktor ini didapatkan dari hasil analisis kebutuhan yang dilakukan dengan cara wawancara dan kuesioner. Aktor-aktor yang didapatkan adalah *Guest* dan Penjual. Identifikasi aktor dijelaskan dalam Tabel 4.1 yang berisi nomor, nama aktor, dan deskripsi dari setiap aktor yang ada.

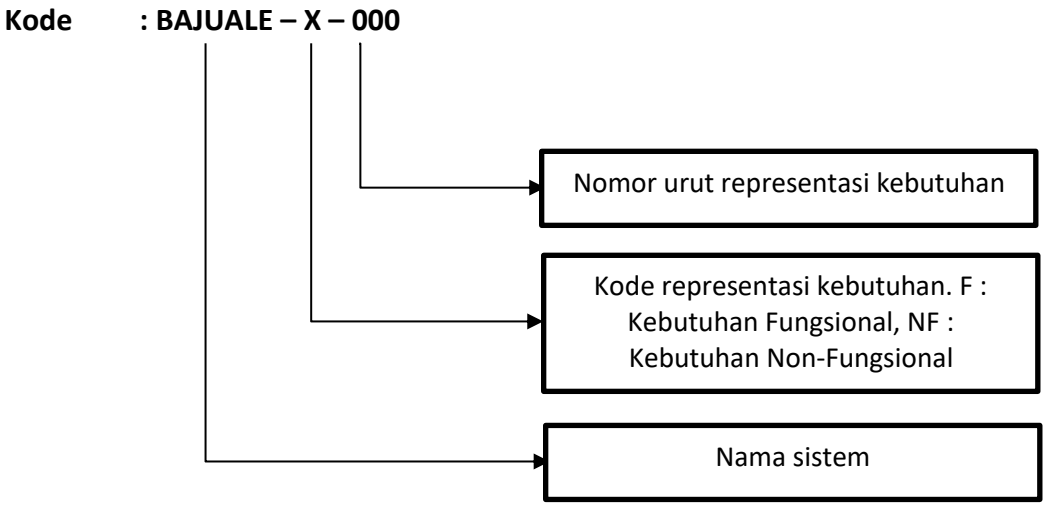
Tabel 4.1 Identifikasi Aktor

No	Nama Aktor	Deskripsi
1.	*Guest*	*Guest* adalah aktor yang dapat melihat isi dari *landing page,* dapat *login* atau *register* ke dalam sistem, dapat melihat list property, melihat detail property, melakukan simulasi kpr, menghubungi agent property melihat info perbandingan kpr antara bank.
2.	*Agent*	Agent adalah aktor yang dapat melakukan edit

		akun, edit info property, menjual property, melihat list property yang dijual, menerima notifikasi pesan email dari calon pembeli, dan mengirim pesan ke calon pembeli melalui email.

4.3 Kebutuhan Pengguna

4.3.1 Aturan Penomoran



Gambar 4.1 Aturan Penomoran

Berdasarkan pada gambar 4.1, aturan penomoran yang ditetapkan dengan kode BAJUALE-X-000 dengan penjelasan sebagai berikut :

- BAJUALE = Merupakan kode yang diterapkan untuk nama sistem
- X = Merupakan kode representasi kebutuhan dengan pengkodean F untuk Kebutuhan Fungsional, dan NF untuk kebutuhan Non- Fungsional.
- 000 = Merupakan nomor urut representasi kebutuhan

4.3.2 Kebutuhan Fungsional

A. Guest

Daftar kebutuhan fungsional yang dimiliki oleh *guest* dijelaskan pada Tabel 4.2 berikut ini.

Tabel 4.2 Kebutuhan Fungsional Aktor *Guest*

No.	Kode	Nama Fungsi	Deskripsi
1.	BAJUALE_F_1	Register Agent	Sistem menyediakan fungsi untuk mendaftarkan sebagai agent bagi guest.
2.	BAJUALE_F_2	Login	Sistem menyediakan fungsi login bagi guest.
3.	BAJUALE_F_3	Melihat List Property	Sistem menyediakan fungsi list property pada halaman landing page dan properties.
4.	BAJUALE_F_4	Melakukan Filter Property	Sistem menyediakan fungsi filter property.
5.	BAJUALE_F_5	Melihat Detail Property	Sistem menyediakan fungsi untuk menampilkan informasi detail dari property.
6.	BAJUALE_F_6	Melakukan Perhitungan Simulasi KPR Property	Sistem menyediakan fungsi untuk melakukan perhitungan simulasi KPR.
7.	BAJUALE_F_7	Melihat Perbandingan KPR Antara Bank	Sistem menyediakan informasi perbandingan KPR antara satu bank dengan bank yang lain.
8.	BAJUALE_F_8	Melihat Informasi KPR Setiap Bank	Sistem menyediakan informasi KPR dari setiap bank.
9.	BAJUALE_F_9	Menghubungi Agent	Sistem menyediakan fungsi mengirim pesan agar guest bisa berkomunikasi dengan agent.
10.	BAJUALE_F_10	Melihat Page Contact	Sistem menyediakan fungsi untuk melihat page contact dengan fungsi untuk menghubungi developer web.
11.	BAJUALE_F_11	Melihat List Agent	Sistem menyediakan fungsi list agent untuk melihat agent yang sudah terdaftar dalam web.
12.	BAJUALE_F_12	Melihat Page About	Sistem menyediakan fungsi untuk melihat page about yang berisi informasi tentang sistem

B. Agent

Daftar kebutuhan fungsional yang dimiliki oleh pembeli dijelaskan pada Tabel 4.3 berikut ini.

Tabel 4.3 Kebutuhan Fungsional Pembeli

No.	Kode	Nama Fungsi	Deskripsi
1.	BAJUALE_F_13	Login	Sistem menyediakan fungsi login untuk agent memasukan email dan password yang sesuai.
2.	BAJUALE_F_14	Mengedit Info Agent	Sistem menyediakan fungsi edit info data pribadi agent seperti nama, email, domisili, nomor telepon, dan juga cv agent.
3.	BAJUALE_F_15	Register Property	Sistem menyediakan fungsi untuk mendaftar property yang ingin dijual.
4.	BAJUALE_F_16	Mengedit Property	Sistem menyediakan fungsi edit untuk memperbarui informasi dari sebuah property yang ingin diubah infonya.
5.	BAJUALE_F_17	Menghapus Property	Sistem menyediakan fungsi untuk menghapus property yang ingin dihapus.
6.	BAJUALE_F_18	Menerima Notifikasi Email	Sistem menyediakan fungsi pemberitahuan bahwa ada pesan yang masuk di email agent.

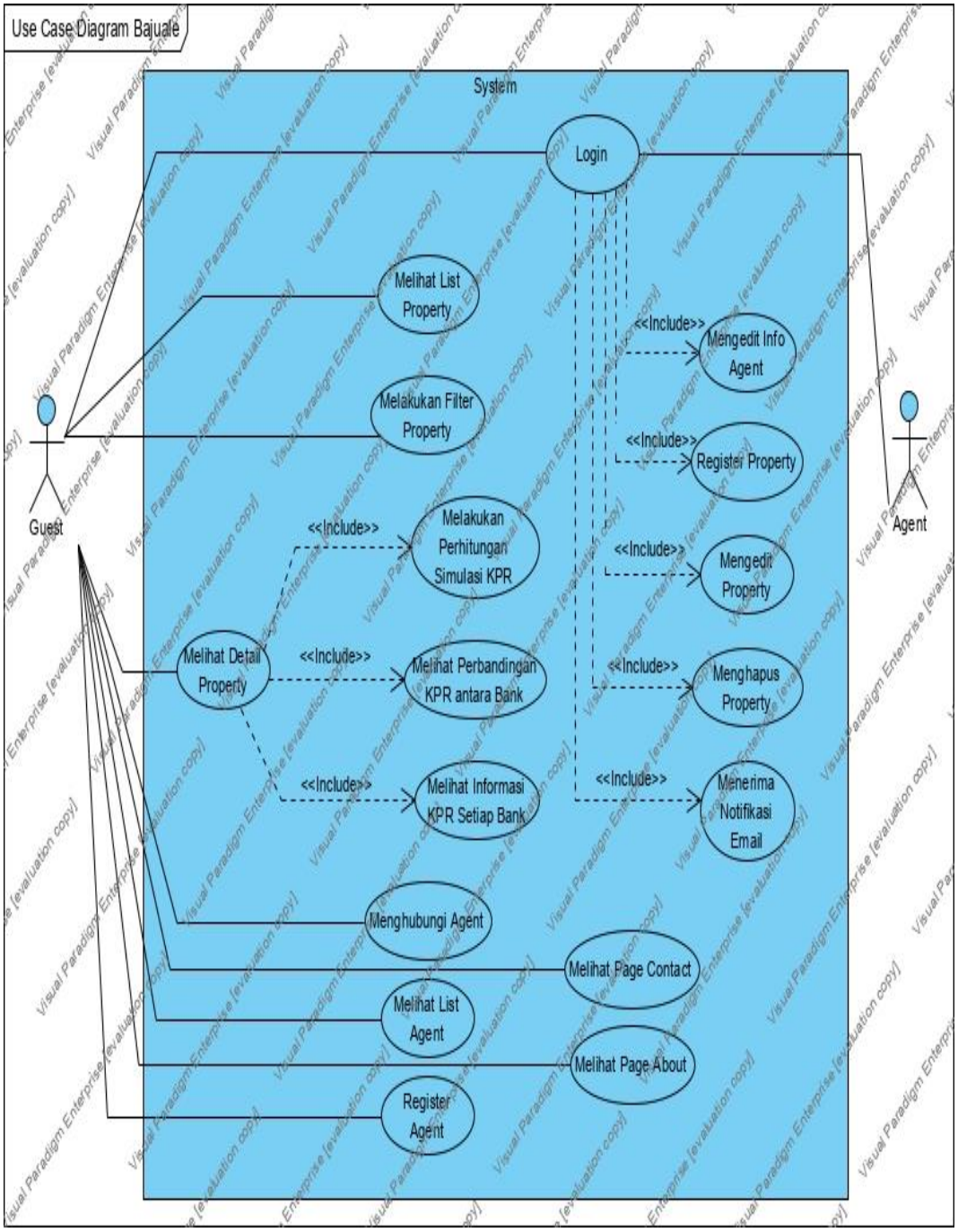
4.3.3 Kebutuhan Non-Fungsional

Tabel 4.4 Kebutuhan Non-Fungsional

No.	Kode	Nama Fungsi	Deskripsi
1.	BAJUALE_NF_1	*Readability*	Kemudahan memahami sistem bagi guest, pembeli, dan penjual.
2.	BAJUALE_NF_2	*Security*	*Member* dan admin login menggunakan email dan password.
3.	BAJUALE_NF_3	*Reliability*	Sistem dapat digunakan selama 24 jam secara terus menerus. Kecuali jika sedang maintenance.

4.4 Pemodelan Kebutuhan

4.4.1 Use Case Diagram



Gambar 4.2 Use Case Diagram

Pada gambar 4.2 terlihat bahwa aktor *guest* dapat melakukan fungsi login, Register *Agent*, Melihat *List Property*, Melakukan *Filter Property,* Melihat Detail *Property,* Melakukan Perhitungan Simulasi KPR, Melihat Perbandingan KPR antar Bank, Melihat Informasi KPR Setiap Bank, Menghubungi *Agent,* Malihat *Page*

Contact, Melihat *List Agent,* dan Melihat *Page About* . Sedangkan aktor *agent* dapat melakukan fungsi Login, Mengedit Info *Agent,* Register *Property,* Mengedit *Property,* Menghapus *Property,* Menerima Notifikasi Email.

4.4.2 Use Case Scenario

Berdasarkan *use case* yang telah dijelaskan pada gambar 4.2, maka sistem marketplace property dan tanah memiliki *use case scenario* yang dapat dilihat pada tabel 4.5 – 4.19 dibawah ini.

Tabel 4.5 Use Case Scenario *Login*

Use Case Scenario *Login*	
Tujuan	Memberikan akses kepada *guest* dan *agent* untuk dapat melakukan aksi sesuai otoritasnya
Aktor	*Guest, Agent*
Kondisi Awal	Aktor sudah mengakses halaman web sistem
Main Flow	1. Aktor memilih opsi *login*. 2. Sistem menampilkan form *login.* 3. Aktor mengisis field-field pada form *login* seperti email dan *password* kemudian memilih tombol login. 4. Sistem memberikan izin kepada *guest* untuk masuk ke halaman utama *agent*.
Skenario Alternatif	1. Jika aktor tidak mengisi field email dan *password* maka akan sistem akan menampilkan pesan bahwa field tidak boleh kososng. 2. Jika aktor mengisi email dan *password* yang tidak cocok, maka sistem akan menampilkan pesan *login* gagal.
Kondisi Akhir	*Guest* masuk sebagai *agent* dan mengakses fitur yang tersedia untuk *agent.*

Tabel 4.6 Use Case Scenario Melihat *List Property*

Use Case Scenario Melihat *List Property*	
Tujuan	*Guest* dapat melihat daftar *property* yang telah didaftarkan oleh *agent* sebagai property jualan ataupun sewaan.
Aktor	*Guest*

Kondisi Awal	Aktor sudah mengakses halaman web sistem
Main Flow	1. Sistem menampilkan landing page sebagai tampilan awal 2. User memilih tombol navigasi *properties* pada navbar 3. Sistem menampilkan Halaman *List Property*
Skenario Alternatif	-
Kondisi Akhir	Sistem menampilkan semua *property* yang diiklankan.

Tabel 4.7 Use Case Scenario Melakukan *Filter Property*

Use Case Scenario Melakukan *Filter Property*	
Tujuan	*Guest* dapat melihat daftar *property* yang telah didaftarkan oleh *agent* sesuai dengan *filter* pilihan yang dimasukan oleh *guest*.
Aktor	*Guest*
Kondisi Awal	Aktor berada pada halaman *properties*.
Main Flow	1. Sistem menyediakan form filter berupa daerah *property* berada, tipe property, dan tipe iklan. 2. User memilih tombol *search property.*
Skenario Alternatif	1. Jika *filter* yang diinginkan aktor tidak terdapat kecocokan pada iklan penjualan rumah, maka sistem menampilkan list kosong.
Kondisi Akhir	Sistem menampilkan semua *property* yang sesuai dengan *filter* yang diinginkan dari aktor.

Tabel 4.8 Use Case Scenario Melihat Detail *Property*

Use Case Scenario Melihat Detail *Property*	
Tujuan	*Guest* dapat melihat informasi detail *property* seperti harga, dp awal, luas lahan, sertifikat, jumlah kamar tidur, jumlah kamar mandi, dan lain lain.
Aktor	*Guest*
Kondisi Awal	Aktor berada pada halaman *properties*.

Main Flow	1. Aktor memilih tombol *more details* atau nama *property* yang ingin dilihat informasi detailnya. 2. Sistem menampilkan informasi terkait *property* berupa biaya *property,* deskripsi, data *property, property's room,* alamat *property dan agent of property.*
Skenario Alternatif	-
Kondisi Akhir	Sistem menampilkan semua informasi detail terkait *property* yang diinginkan.

Tabel 4.9 Use Case Scenario Melakukan Perhitungan Simulasi KPR

Use Case Scenario Melakukan Perhitungan Simulasi KPR	
Tujuan	*Guest* dapat melakukan perhitungan simulasi dari KPR yang ingin diambil sebagai acuan dalam menentukan biaya cicilan rumah perbulannya.
Aktor	*Guest*
Kondisi Awal	Aktor berada pada halaman detail *property*.
Main Flow	1. Aktor memilih tombol simulasi KPR. 2. Aktor mengisi form berupa jangka waktu pelunasan dan estimasi bunga per tahun.
Skenario Alternatif	-
Kondisi Akhir	Sistem menampilkan biaya angsuran bulanan, sesuai dengan perhitungan yang telah dimasukan oleh aktor.

Tabel 4.10 Use Case Scenario Melihat Perbandingan KPR antara Bank

Use Case Scenario Melihat Perbandingan KPR antara Bank	
Tujuan	*Guest* dapat melihat perbandingan informasi biaya antara bank yang ingin dibandingkan.
Aktor	*Guest*
Kondisi Awal	Aktor berada pada halaman detail *property*.
Main Flow	1. Aktor memilih tombol simulasi KPR. 2. Aktor memilih dua bank yang ingin dibandingkan biaya KPR nya.
Skenario Alternatif	-

Kondisi Akhir	Sistem menampilkan perbandingan informasi biaya antara dua bank yang telah dipilih oleh aktor.

Tabel 4.11 Use Case Scenario Melihat Informasi KPR setiap Bank

Use Case Scenario Melihat Informasi KPR setiap Bank	
Tujuan	*Guest* dapat melihat informasi KPR setiap bank mulai dari maksimal tahun pelunasan, minimal bunga, dan lain lain.
Aktor	*Guest*
Kondisi Awal	Aktor berada pada halaman detail *property*.
Main Flow	1. Aktor memilih tombol simulasi KPR. 2. Aktor memilih bank yang ingin dilihat informasi terkait KPR nya. 3. Aktor memeilih tombol lihat.
Skenario Alternatif	-
Kondisi Akhir	Sistem menampilkan informasi KPR bank yang dipilih oleh aktor.

Tabel 4.12 Use Case Scenario Menghubungi *Agent*

Use Case Scenario Menghubungi *Agent*	
Tujuan	*Guest* dapat mengirim pesan kepada *agent* yang selanjutnya akan masuk ke email *agent*.
Aktor	*Guest*
Kondisi Awal	Aktor berada pada halaman detail *property*.
Main Flow	1. Aktor mengisi field-field yang tersedia pada form seperti nama lengkap, nomor telepon, alamat email, dan pesan yang ingin disampaikan. 2. Aktor memilih tombol *submit request.*
Skenario Alternatif	1. Jika aktor tidak mengisi salah satu field yang tersedia maka sistem akan menampilkan pesan bahwa field tidak boleh kosong.
Kondisi Akhir	Sistem menampilkan pesan bahwa pesan berhasil terkirim.

Tabel 4.13 Use Case Scenario Melihat *Page Contact*

Use Case Scenario Melihat *Page Contact*	
Tujuan	Aktor dapat menghubungi develop sistem melalui halaman *contact.*
Aktor	*Guest*
Kondisi Awal	Aktor sudah berada dalam tampilan awal web sistem.
Main Flow	1. Sistem menampilkan landing page sebagai tampilan awal 2. *Guest* memilih tombol navigasi *Contact* pada navbar 3. Sistem menampilkan Halaman *Contact* 4. *Guest* mengisi email pada form 5. *Guest* mengisi keluhan/isi pesan pada form 6. *Guest* menekan tombol submit 7. Sistem memunculkan pop up pesan berhasil dikirim
Skenario Alternatif	1. Jika aktor tidak mengisi salah satu field yang tersedia maka sistem akan menampilkan pesan bahwa field tidak boleh kosong.
Kondisi Akhir	User berhasil mengirim pesan ke develop.

Tabel 4.14 Use Case Scenario Melihat *List Agent*

Use Case Scenario Melihat *List Agent*	
Tujuan	Aktor dapat melihat daftar *agent* yang telah terdaftar
Aktor	*Guest*
Kondisi Awal	Aktor sudah dalam halaman awal web sistem
Main Flow	1. Sistem menampilkan landing page sebagai tampilan awal 2. *Guest* memilih tombol navigasi *List agent* pada navbar 3. Sistem menampilkan Halaman *List agent* 4. *Guest* menekan tombol detail pada *agent* yang ingin dilihat informasinya 5. Sistem menampilkan informasi detail dari *agent* yang dipilih
Skenario Alternatif	-
Kondisi Akhir	*Guest* berhasil melihat informasi *agent*.

Tabel 4.15 Use Case Scenario Melihat *Page About*

Use Case Scenario Melihat *Page About*	
Tujuan	Aktor dapat masuk ke halaman *about* untuk melihat visi dan misi dari dikembangkannya web *marketplace* tanah dan properti
Aktor	*Guest*
Kondisi Awal	Aktor sudah dalam halaman awal web sistem
Main Flow	1. Sistem menampilkan *landing page* sebagai tampilan awal 2. *Guest* memilih tombol navigasi *About* pada navbar 3. Sistem menampilkan Halaman About
Skenario Alternatif	-
Kondisi Akhir	*Guest* berhasil masuk ke halaman *about*.

Tabel 4.16 Use Case Scenario Register *Agent*

Use Case Scenario Register *Agent*	
Tujuan	*Guest* dapat mendaftar sebagai penjual properti
Aktor	*Guest*
Kondisi Awal	Aktor sudah dalam halaman awal web sistem
Main Flow	1. Sistem menampilkan landing page sebagai tampilan awal 2. *Guest* memilih tombol navigasi *Sale* pada navbar 3. *Guest* menekan tombol Daftar sekarang! 4. *Guest* mengisi field yang sudah disediakan di dalam form seperti foto profil, nama lengkap, alamat email, password, nomor telepon, domisili, dan juga file cv 5. *Guest* menekan tombol daftar menjadi agen
Skenario Alternatif	1. Jika aktor tidak mengisi salah satu field yang tersedia maka sistem akan menampilkan pesan bahwa field tidak boleh kosong.
Kondisi Akhir	*Guest* berhasil terdaftar sebagai *agent*.

Tabel 4.17 Use Case Scenario Mengedit Info *Agent*

Use Case Scenario Mengedit Info *Agent*	
Tujuan	*Agent* dapat mengedit info pribadinya
Aktor	*Agent*
Kondisi Awal	Aktor berada pada halaman *dashboard agent*
Main Flow	1. *Agent* memilih fungsi dropdown yang terdapat pada foto profil *agent* 2. *Agent* menekan tombol pilihan *Account* 3. *Agent* mengisi data yang ingin di edit 4. *Agent* menekan tombol Edit Akun 5. Sistem melakukan *authentication* 6. Sistem menampilkan pop-up pesan Akun Berhasil di Edit
Skenario Alternatif	1. Jika aktor menghapus info pada salah satu field dan tidak diisi kembali maka sistem akan menampilkan pesan bahwa field tidak boleh kosong. 2. Jika pada prosesnya sistem menemukan email yang sama dengan akun lain pada *database* maka sistem akan menampilkan pesan bahwa email telah terpakai
Kondisi Akhir	*Agent* berhasil mengubah info pribadi

Tabel 4.18 Use Case Scenario Register *Property*

Use Case Scenario Register *Property*	
Tujuan	Aktor dapat mendaftar properti yang ingin dijual
Aktor	*Agent*
Kondisi Awal	Aktor berada pada halaman *dashboard agent*
Main Flow	1. *Agent* memilih navigasi jual properti pada bagian navbar *agent* 2. *Agent* mengisi field-field yang telah disediakan pada form seperti data umum properti, fasilitas, alamat, dan harga properti 3. *Agent* memilih tombol jual properti 4. Sistem menampilkan pesan bahwa properti berhasil terdaftar
Skenario Alternatif	1. Jika aktor tidak mengisi salah satu field yang tersedia maka sistem akan menampilkan pesan bahwa field tidak boleh kosong.

Kondisi Akhir	*Agent* berhasil mendaftar properti yang ingin dijual.

Tabel 4.19 Use Case Scenario Mengedit *Property*

Use Case Scenario Mengedit *Property*	
Tujuan	*Agent* dapat mengedit info properti yang sudah terdaftar
Aktor	*Agent*
Kondisi Awal	Aktor berada pada halaman *dashboard agent*
Main Flow	1. *Agent* memilih tombol edit pada card properti yang ingin diubah infonya 2. *Agent* mengisi data yang ingin di edit 3. *Agent* menekan tombol Edit Property 4. Sistem melakukan *authentication* 5. Sistem menampilkan pop-up pesan Property Berhasil di Edit
Skenario Alternatif	1. Jika aktor menghapus info pada salah satu field dan tidak diisi kembali maka sistem akan menampilkan pesan bahwa field tidak boleh kosong.
Kondisi Akhir	*Agent* berhasil mengubah info properti

Tabel 4.20 Use Case Scenario Menghapus *Property*

Use Case Scenario Menghapus *Property*	
Tujuan	*Agent* dapat menghapus properti yang sudah terdaftar
Aktor	*Agent*
Kondisi Awal	Aktor berada pada halaman *dashboard agent*
Main Flow	1. *Agent* memilih tombol hapus pada card properti yang ingin dihapus 2. Sistem menampilkan pilihan yakin untuk dihapus atau tidak 3. *Agent* memilih tombol yakin hapus 4. Sistem menampilkan pop-up pesan Property Berhasil dihapus
Skenario Alternatif	-
Kondisi Akhir	*Agent* berhasil menghapus properti

Tabel 4.21 Use Case Scenario Menerima Notifikasi Email

Use Case Scenario Menerima Notifikasi Email	
Tujuan	*Agent* mendapatkan notifikasi pesan di email
Aktor	*Agent*
Kondisi Awal	Aktor berada pada halaman *dashboard agent*
Main Flow	1. *Agent* memilih icon lonceng yang terdapat pada navbar *agent* 2. Sistem menampilkan pesan masuk 3. *Agent* memilih pesan yang ingin dilihat 4. Sistem akan men-*direct* ke halaman email pribadi *agent*
Skenario Alternatif	-
Kondisi Akhir	*Agent* berhasil mebaca pesan pada notifikasi yang masuk