

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Berdasarkan pada teori dari penelitian penulis saat ini, diperlukan data-data pendukung yang bisa untuk dijadikan bahan pertimbangan dalam pembuatan Sistem Informasi Rumah Susun Sederhana Sewa(Rusunawa) Berbasis *Website* dari penelitian terdahulu yang relevan serta mencakup pembahasan yang dibahas dalam penelitian ini. Sebagai bahan pertimbangan dalam penelitian ini penulis mencantumkan beberapa hasil penelitian terdahulu oleh beberapa peneliti yang penulis ketahui, diantaranya :

Dari hasil penelitian yang berjudul “Sistem Informasi Manajemen Rumah Susun Sumbawa Universitas Teknologi Sumbawa Berbasis Web” yang dilakukan oleh (Karisma et al., 2021). Penelitian ini merupakan salah satu penelitian yang menggunakan Metode *Waterfall* dengan bertujuan untuk memudahkan pembina dan pengurus RUSUNAWA UTS untuk mengawasi, menertibkan, memberitahukan dan mengolah data yang ada, sehingga dalam semua kegiatan yang ada dapat berjalan dengan efektif dan efisien serta dapat menyampaikan informasi dengan tepat waktu. Berdasarkan penelitian yang dilakukan dapat disimpulkan bahwa Sistem Informasi Manajemen Rumah Susun Sumbawa Berbasis Web dapat menjadi masukan bagi pengelola RUSUNAWA UTS untuk mempermudah dalam melakukan seluruh kegiatan yang ada di RUSUNAWA, memudahkan dalam melakukan pendataan penghuni dan mempermudah dalam melakukan izin di RUSUNAWA.

Kemudian ada penelitian yang berjudul “Aplikasi Rumah Susun Teluk Kelayan Kota Banjarmasin Berbasis Web” yang dilakukan oleh (Aminah et al., 2022) yang bertujuan untuk mempermudah calon penghuni mendapatkan informasi yang lengkap serta dapat memudahkan pengguna atau konsumen melakukan registrasi pada Rumah Susun Teluk Kelayan. Berdasarkan penelitian yang telah dilakukan ditarik kesimpulan bahwa dengan adanya aplikasi ini pengunjung, konsumen atau pegawai instansi khususnya yang ada diluar rumah susun teluk nelayan bisa melakukan penyewaan dan melakukan pemesanan secara

terstruktur dan interaktif antara masyarakat rumah susun Teluk Kelayan. Aplikasi ini juga mempermudah proses penginputan unit dan tipe unit yang terstruktur.

Selain itu, ada juga penelitian yang dilakukan oleh (Sarfini, 2021) yang berjudul “Sistem Informasi Pengelolaan Rumah Susun Sederhana Sewa(RUSUNAWA) Pada Dinas Perumahan Rakyat Dan Kawasan Permukiman Kota Pontianak”. Penelitian ini merupakan salah satu penelitian yang dilakukan di kota Pontianak dengan bertujuan untuk memilih pemohon yang berhak menghuni Rusunawa berdasarkan waktu pendaftaran dan kelengkapan persyaratannya, mengelola data pemohon dan penghuni secara komputerisasi, sistem juga dapat membantu memberikan kemudahan baik dari segi waktu maupun biaya dan menjaga keamanan data pemohon dan penghuni baik itu dari kerusakan maupun kehilangan. Berdasarkan penelitian yang telah dilakukan didapat kesimpulan bahwa sistem yang dibangun berhasil berjalan dan dikelola dengan baik serta memberikan kemudahan dalam segi waktu seperti kecepatan mengakses data pemohon dan penghuni, maupun biaya seperti pembelian buku dan alat tulis untuk mencatat data pemohon yang mendaftar.

Kemudian ada penelitian yang berjudul “Rancang Bangun Simawa (Sistem Informasi Rusunawa) Berbasis Web Application Menggunakan Framework Laravel (Studi Kasus : Universitas Negeri Surabaya)” yang dilakukan oleh (Fahriya & Nurhidayat, 2018). Penelitian ini bertujuan untuk memudahkan proses pendaftaran dan pengunduran diri serta memudahkan user melihat ketersediaan kamar kosong dan fasilitas yang ada di Rusunawa. Selain itu, aplikasi ini juga membantu pengelola dalam manajemen data-data dalam pemesanan dan juga pembayaran di Rusunawa. Hasil rata-rata dari setiap pernyataan kuisioner yang dilakukan dalam penelitian ini adalah 80%-100% sehingga dapat ditarik kesimpulan bahwa aplikasi beserta semua fitur yang ada dalam penelitian ini sesuai dengan yang diharapkan .

Terakhir ada penelitian yang berjudul “Fleksibilitas Sistem Informasi dari Perspektif Pengguna Dan Pengembang Sistem Informasi ” (Arafat, 2016). Penelitian ini bertujuan untuk mengetahui bagaimana pandangan dari pengguna dan pengembang sistem dan apa peran dari kedua pihak tersebut dalam kesuksesan

penerapan asas fleksibilitas pada sistem informasi. Pada penelitian ini dikatakan bahwa Sistem informasi (SI) ditujukan untuk mendukung pelaksanaan proses bisnis suatu unit organisasi. Untuk meningkatkan kinerja proses bisnis dapat didekati dengan meningkatkan efisiensi dan efektivitas dari sistem informasi pendukungnya. Salah satu faktor penting untuk meningkatkan kinerja sistem informasi adalah dengan menggunakan asas fleksibilitas pada sistem informasi tersebut. Dengan menghubungkan karakteristik proses bisnis dengan strategi penerapan fleksibilitas sistem informasi dapat ditarik kesimpulan pada penelitian ini bahwa dalam pembangunan sistem informasi yang menggunakan asas fleksibilitas, peran pengguna tidak dapat dipisahkan dari kesuksesan. Persiapan yang tidak matang dari kedua belah pihak, baik pengguna maupun pengembang dapat menyebabkan kurang maksimalnya hasil yang diraih.

Dari 5 (lima) penelitian terkait tersebut aplikasi yang dirancang akan mengakuisisi beberapa fitur di antaranya pengelolaan data penghuni, kamar, tagihan, pengumuman, kegiatan dan pendaftaran serta dapat mengadaptasi pada kebutuhan sistem yang baru atau berubah nantinya.

2.2 Landasan Teori

2.2.1 Rumah Susun Sederhana Sewa(RUSUNAWA)

Rusunawa adalah singkatan dari Rumah Susun Sederhana Sewa. Rusunawa merupakan bangunan Gedung bertingkat yang dibangun dalam suatu lingkungan yang terbagi dalam bagian-bagian terstruktur secara fungsional dalam arah horizontal maupun vertical dan merupakan satuan-satuan yang masing-masing digunakan secara terpisah, status penguasaannya sewa serta dibangun dengan menggunakan dana Anggaran Pendapatan dan Belanja Negara dan/atau Anggaran Pendapatan Belanja Daerah dengan fungsi utamanya sebagai hunian.(Kementerian Negara Perumahan Rakyat Republik Indonesia,2008). Menurut UU No 20 Tahun 2011 pada Pasal 1 menuturkan,“ Rumah Susun merupakan bangunan gedung bersusun yang dibentuk dalam sesuatu kawasan, yang dibagi dalam bagianbagian yang distrukturkan secara fungsional dalam arah horizontal ataupun vertikal serta merupakan satuan- satuan yang masing- masing bisa dimiliki serta dipakai secara

terpisah, paling utama untuk tempat hunian, yang dilengkapi dengan bagian-bersama, benda- bersama serta tanah- bersama”.

2.2.2 Sistem Informasi

Sistem informasi merupakan gabungan dari empat bagian utama. Keempat bagian utama tersebut mencakup perangkat lunak (software), perangkat keras (hardware), infrastruktur, dan Sumber Daya Manusia (SDM) yang terlatih. Keempat bagian utama ini saling berkaitan untuk menciptakan sebuah sistem yang dapat mengelola data menjadi informasi yang bermanfaat (Risdiyansyah, 2017).

Sistem informasi juga merupakan metode yang memungkinkan pengguna dapat melihat informasi yang disajikan dengan benar. Bukan hanya sebuah perusahaan saja yang menyediakan sebuah sistem informasi. Untuk perguruan tinggi negeri sangat membutuhkan yang namanya sistem informasi guna untuk memudahkan mahasiswa, dosen atau bahkan karyawan untuk mencari tau ataupun mengakses sesuatu hal yang berhubungan dengan perguruan tinggi negeri tersebut. Dengan adanya sebuah sistem informasi dapat memajukan perguruan tinggi negeri tersebut dari segi sistemnya (Khusniatul Fahriya, 2018).

2.2.3 Fleksibilitas Sistem Informasi

Fleksibilitas dapat diterjemahkan sebagai kemampuan dari perangkat lunak (software), yang bisa diimplementasikan ke dalam berbagai jenis dan spesifikasi sistem komputer. Dalam Kamus Besar Bahasa Indonesia Fleksibilitas adalah kelenturan atau mudah diatur, pengertian fleksibel dalam kewirausahaan adalah mampu untuk menyesuaikan bisnis dalam perusahaan atau organisasi dengan situasi dan kondisi tertentu, dengan kata lain arti dari fleksibilitas ini mengungkapkan bahwa perusahaan atau organisasi dituntut bisa mengikuti perkembangan jaman dalam hal ini terutama perkembangan teknologi informasi. Fleksibilitas juga merupakan kemampuan organisasi dalam menyikapi atau merespon terhadap perubahan lingkungan, baik internal maupun eksternal. Sedangkan fleksibilitas sistem informasi dapat diartikan sebagai kemampuan dari sebuah sistem informasi dalam menyikapi atau merespon kebutuhan pengguna akan requirement yang baru, berbeda, atau berubah. Salah satu

faktor penting untuk meningkatkan kinerja sistem informasi adalah dengan menggunakan asas fleksibilitas pada sistem informasi tersebut.

(Arafat, 2016).

Ada beberapa karakteristik proses bisnis yang menjadi bagian dari analisis fleksibilitas sistem informasi yaitu ketidakpastian (*uncertainty*), keanekaragaman (*variability*), dan keterbatasan waktu (*time-criticality*) (Judith Gebauer, 2006).Kolerasi antara karakteristik proses bisnis dengan starategi fleksibilitas sistem informasi meliputi :

1. Karakteristik ketidakpastian (*uncertainty*) berkolerasi dengan strategi *flexibility-to-change*, unsur ketidakpastian dari proses bisnis menyebabkan adanya kemungkinan perubahan proses bisnis, sehingga sistem informasi haruslah relatif mudah untuk berubah (bersifat adaptif).
2. Karakteristik keanekaragaman (*variability*) berkorelasi dengan strategi *flexibility-to-use*. Unsur keberagaman dari proses bisnis menyebabkan sistem informasi yang ada haruslah menyediakan berbagai macam “alur” penggunaan (bersifat dinamis), misalnya pengguna dapat mengubah layout laporan.

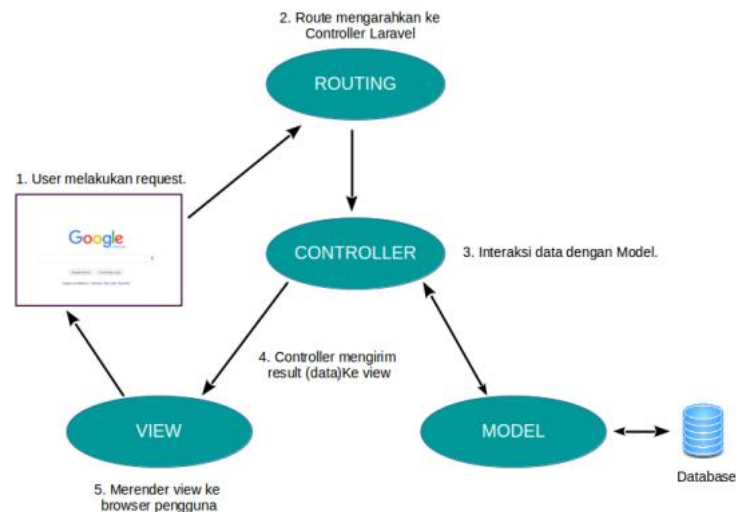
Kedua strategi tersebut ditujukan untuk mendukung pelaksanaan proses bisnis yang bersifat *time-criticality*.

2.2.4 Website

Website adalah kumpulan dari halaman-halaman situs yang terdapat dalam sebuah domain atau subdomain yang berada didalam *World Wide Web*(WWW) di internet. Alasan seseorang mengunjungi website adalah konten yang tersedia di *website* tersebut. Fungsi utama dari sebuah website adalah menyampaikan informasi. Dengan tersedianya informasi, website dapat digunakan untuk mengubah pengunjung menjadi prospek. Untuk mengubah pengunjung situs web menjadi prospek, pengelola website dapat menyediakan formulir agar pengunjung dapat menyampaikan alamat email dan informasi lainnya sehingga menjadi prospek yang teridentifikasi. (Laily, 2022).

2.2.5 Laravel

Menurut (Tamus Bin Tahir, 2019) Laravel adalah sebuah framework web berbasis PHP yang open-source dan tidak berbayar, diciptakan oleh Taylor Otwell dan diperuntukkan untuk pengembangan aplikasi web yang menggunakan pola MVC. Struktur pola MVC pada laravel sedikit berbeda pada struktur pola MVC pada umumnya. Di Laravel terdapat routing yang menjembatani antara request dari user dan controller. Jadi controller tidak langsung menerima request tersebut



Gambar 2. 1 Konsep MVC Pada Laravel

Ada 5 konsep arsitektur pada framework laravel yang mempunyai masing-masing fungsi diantaranya:

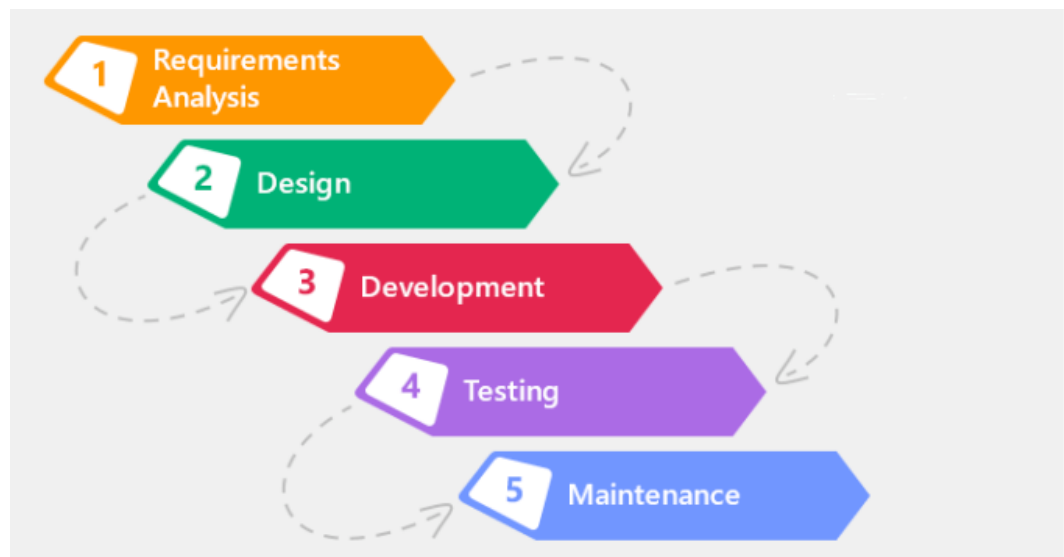
- Routes*, berfungsi sebagai pemberi akses pada setiap *request* sesuai alur yang telah di tentukan.
- Controller*, adalah bagian yang menjadi penghubung antara model dan *view*. *Controller* memiliki perintah-perintah yang berfungsi untuk memproses bagaimana data ditampilkan dari Model ke *View* atau sebaliknya.
- Model, merupakan sekumpulan data yang memiliki fungsi-fungsi untuk mengelola suatu *table* pada sebuah *database*. Struktur pemodelan data pada laravel yakni memiliki fungsi yang terdiri dari *table*, *primaryKey* dan *fillable*. Dimana ketiga fungsi tersebut harus di *protected*. Pada bagian *table* harus diisi dengan nama *table* yang sesuai pada *database*, di bagian *primaryKey* harus diisi sesuai *primary key* pada *table* tersebut dan pada

bagian *fillable* diisi dengan bagian-bagian yang mencakup dalam *table* tersebut

- d. *View*, merupakan file yang berisi kode html (*HyperText Markup Language*) yang berfungsi untuk menampilkan suatu data ke dalam browser. Format *view* pada laravel harus menggunakan istilah *blade*, contohnya seperti: `view.blade.php`.
- e. *Migrations*, merupakan proses perancangan suatu table, dalam hal ini *migrations* berfungsi sebagai *blueprint database* atau dapat diistilahkan sebagai penyedia sistem kontrol untuk skema *database*.

Framework laravel memiliki keunggulan tersendiri yang menjadikannya lebih baik dari pada framework lainnya, berikut ini merupakan kelebihan dari laravel yaitu, performance lebih cepat, reload data lebih stabil, memiliki keamanan data, menggunakan fitur canggih seperti blade menggunakan konsep HMVC (*Hierarchical Model View Controller*), tersedianya *library-library* yang sudah siap untuk digunakan dan adanya fitur pengelolaan migrations untuk pembuatan skema table pada *database*.

2.2.6 SDLC(*Software Development Lyfe Cycle*)



Gambar 2. 2 SDLC

System Development Life Cycle (SDLC) atau siklus hidup pengembangan sistem dalam rekayasa sistem dan rekayasa perangkat lunak adalah proses pembuatan dan perubahan sistem serta model dan metodologi yang digunakan untuk mengembangkan sistem-sistem tersebut. SDLC juga merupakan pola untuk mengembangkan sistem perangkat lunak yang terdiri dari tahapan perencanaan (*planning*), analisis (*analyst*), desain (*design*), implementasi (*implementation*), uji coba (*testing*) dan pengelolaan (*maintenance*) (Wahid, 2020). Dalam rekayasa sistem atau rekayasa perangkat lunak adalah proses pembuatan dan perubahan sistem serta model dan metodologi yang digunakan untuk mengembangkan sistem-sistem (Rani Susanto, 2016). SDLC mempunyai beberapa fase, yaitu:

1. Perencanaan sistem (System Planing)
2. Analisis Sistem (System Analysis)
3. Perancangan Sistem (System Design)
4. Implementasi Sistem (System Implementation)
5. Pemeliharaan Sistem (System Maintenance)

2.2.6.1 Object Oriented Analysis Design(OOAD)

Object Oriented Analysis and Design (OOAD) merupakan metode untuk menganalisa dan merancang sistem dengan pendekatan berorientasi *object*. Object diartikan sebagai entitas yang memiliki identitas, state dan behavior. Pada analisa identitas *object* menjelaskan bagaimana user membedakannya dari *object* lain dan *behavior object* digambarkan melalui event yang dilakukan. Pada perancangan, identitas *object* dengan bagaimana *object* lain mengenalinya sehingga bisa diakses dan *behavior* dengan operation yang dilakukan. Sehingga *object* satu bisa mempengaruhi *object* lain dalam sistem (Purwaningtias, 2018).

Adapun tahapan- tahapan yang ada pada metode OOAD meliputi sebagai berikut :

1. Analisis Kebutuhan

Tahapan ini merupakan langkah awal dalam pembangunan sistem dimana dalam tahapan ini dilakukan beberapa kegiatan perencanaan yaitu, identifikasi permasalahan, menganalisa kebutuhan sampai dengan penetapan jadwal pelaksanaan pembangunan sistem. Pada

tahapan ini dapat dimulai dengan mendengarkan kumpulan kebutuhan aktifitas dari suatu sistem yang memungkinkan pengguna dapat memahami proses bisnis untuk sistem dan mendapatkan gambaran yang jelas dalam mengenai fitur utama, fungsionalitas dan keluaran yang diinginkan.

2. Perancangan

Tahapan berikutnya adalah perancangan dimana pada tahapan ini dilakukan kegiatan pemodelan yang dimulai dari pemodelan sistem, pemodelan arsitektur sampai dengan pemodelan basis data. Pemodelan sistem dan arsitektur menggunakan diagram *Unified Modelling Language* (UML) sedangkan pemodelan basis data menggunakan *Entity Relationship Diagram* (ERD).

3. Implementasi

Pembuatan perangkat lunak merupakan tahap membuat sistem yang dilengkapi dengan perintah-perintah algoritma yang telah dirancang sebelumnya yang bertujuan untuk menerapkan sistem yang telah dirancang untuk dilakukan kodefikasi dengan pemrograman yang sesuai dengan kebutuhan dan perancangan sistem. Adapun bahasa pemrograman yang digunakan adalah PHP dengan metode terstruktur. Untuk sistem manajemen basis data menggunakan piranti lunak MySQL.

4. Pengujian

Setelah tahapan pengkodean selesai, kemudian dilakukan tahapan pengujian sistem untuk mengetahui kesalahan apa saja yang timbul saat aplikasi sedang berjalan serta mengetahui apakah sistem yang dibangun sudah sesuai dengan kebutuhan pengguna. Metode pengujian yang digunakan pada tahapan ini adalah metode *blackbox* testing dan tes UAT, dimana pengujian yang dilakukan terhadap form beberapa masukkan apakah sudah berjalan sesuai dengan fungsinya masing-masing.

5. Pemeliharaan

Tahapan ini merupakan tahap pemeliharaan sistem yang telah diimplementasikan. Tahap ini meliputi beberapa aktivitas yang bertujuan untuk memastikan bahwa sistem dapat berjalan dengan baik dan dapat memenuhi kebutuhan pengguna secara efektif dan efisien. Tahap maintenance ini sangat penting dilakukan agar sistem dapat terus digunakan dengan baik dan dapat memberikan manfaat yang diharapkan oleh pengguna dan *stakeholder*.

2.2.7 UML(*Unified Modelling Language*)

Teknik untuk perancangan OOAD (*Object Oriented Analysis Design*) menggunakan *Unified Modelling Language* (UML). Dikutip dari Wikipedia *Unified Modeling Language* (UML) adalah himpunan struktur dan teknik untuk pemodelan desain program berorientasi objek (OOP) serta aplikasinya. UML adalah metodologi untuk mengembangkan sistem OOP dan sekelompok perangkat tool untuk mendukung pengembangan sistem tersebut. Ada juga menurut (Dede Wira Trise Putra, 2019) UML adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan requirement, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek.

UML menyediakan 10 macam diagram untuk memodelkan aplikasi berorientasi objek, yaitu :

1. *Use Case Diagram* untuk memodelkan proses bisnis.
2. *Conceptual Diagram* untuk memodelkan konsep-konsep yang ada di dalam aplikasi.
3. *Sequence Diagram* untuk memodelkan pengiriman pesan (message) antar objects.
4. *Collaboration Diagram* untuk memodelkan interaksi antar *objects*.
5. *State Diagram* untuk memodelkan perilaku objects di dalam sistem.
6. *Activity Diagram* untuk memodelkan perilaku *Use Cases* dan objects di dalam system.
7. *Class Diagram* untuk memodelkan struktur kelas.
8. *Object Diagram* untuk memodelkan struktur object.

9. *Component Diagram* untuk memodelkan komponen object.
10. *Deployment Diagram* untuk memodelkan distribusi aplikasi.

2.2.7.1 Use Case Diagram

Use case diagram merupakan pemodelan untuk kelakuan sistem informasi yang akan dibuat. Use case bekerja dengan mendeskripsikan tipikal interaksi antara user sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sistem itu dipakai. (Dede Wira Trise Putra, 2019).

2.2.7.2 Activity Diagram

Activity diagram, dalam bahasa Indonesia diagram aktivitas, yaitu diagram yang dapat memodelkan proses-proses yang terjadi pada sebuah sistem. Runtutan proses dari suatu sistem digambarkan secara vertikal. *Activity diagram* merupakan pengembangan dari *Use Case* yang memiliki alur aktivitas. *Activity Diagram* memiliki beberapa tujuan diantaranya :

- a. Menjelaskan urutan aktivitas dalam suatu proses.
- b. Di dalam dunia bisnis biasanya digunakan untuk modeling (memperlihatkan urutan proses bisnis).
- c. Mudah dalam memahami proses yang ada dalam sistem secara keseluruhan.
- d. Merupakan metode perancangan yang terstruktur, mirip dengan Flowchart maupun *Data Flow Diagram* (DFD).
- e. Mengetahui aktivitas aktor/pengguna berdasarkan *use case/diagram* yang dibuat sebelumnya.

2.2.7.3 Class Diagram

Menurut (Rinaldi, 2019) *Class diagram* adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class diagram* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi). *Class diagram* menggambarkan struktur dan deskripsi class, package dan objek beserta hubungan satu sama lain seperti containment, pewarisan, asosiasi, dan lain-lain. Sedangkan menurut (Rosa Ariani Sukanto, 2016) Diagram kelas atau class diagram menggambarkan struktur

sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem.

2.2.8 Pengujian Perangkat Lunak

Pengujian adalah suatu proses pengeksekusian program yang bertujuan untuk menemukan kesalahan. Pengujian sebaiknya menemukan kesalahan yang tidak disengaja dan pengujian dinyatakan sukses jika berhasil memperbaiki kesalahan tersebut. Selain itu, pengujian juga bertujuan untuk menunjukkan kesesuaian fungsi-fungsi perangkat lunak dengan spesifikasinya. Sebuah perangkat lunak dinyatakan gagal, jika perangkat lunak tersebut tidak memenuhi spesifikasi. (MZ, 2016).

2.2.8.1 Pengujian *Black Box Testing*

Pendekatan pengujian *Black-Box* adalah metode pengujian di mana data tes berasal dari persyaratan fungsional yang ditentukan tanpa memperhatikan struktur program akhir (Komarudin, 2016). *Blackbox* testing berusaha untuk menemukan kesalahan dalam kategori sebagai berikut:

1. Fungsi-fungsi yang tidak benar atau fungsi yang hilang
2. Kesalahan antarmuka (*interface*)
3. Kesalahan dalam struktur data atau akses database eksternal
4. Kesalahan perilaku (*behavior*) atau kesalahan kinerja
5. Inisialisasi dan pemutusan kesalahan.

2.2.8.2 Pengujian *User Acceptance Test(UAT)*

User Acceptance Test (UAT) atau Uji Penerimaan Pengguna adalah suatu proses pengujian oleh pengguna yang dimaksudkan untuk menghasilkan dokumen yang dijadikan bukti bahwa perangkat lunak yang telah dikembangkan telah dapat diterima oleh pengguna, apabila hasil pengujian sudah bisa dianggap memenuhi kebutuhan dari pengguna. Pengujian UAT dilakukan oleh pengguna target dari aplikasi yang dibangun untuk mengetahui aplikasi berjalan sesuai dengan proses yang telah ditentukan bertujuan untuk mencatat dan mengoreksi bug yang

ditemukan sebelum aplikasi di rilis. UAT biasanya dinilai oleh pengguna aplikasi tersebut, dengan menggunakan kusioner sesuai test case yang diinginkan.

Pengujian UAT didasarkan pada dokumen requirement yang disepakati bersama. Dokumen requirement adalah dokumen yang berisi lingkup pekerjaan perangkat lunak yang harus dikembangkan, dengan demikian maka dokumen ini semestinya menjadi acuan untuk pengujian.

Proses dalam UAT meliputi pemeriksaan dan pengujian terhadap hasil pekerjaan. Setelah diperiksa selanjutnya memastikan item-item yang ada dalam dokumen requirement sudah ada dalam perangkat lunak yang diuji atau tidak. Selanjutnya menguji semua item yang telah ada telah dapat memenuhi kebutuhan penggunanya. Hasil dari UAT adalah dokumen yang menunjukkan bukti pengujian, berdasarkan bukti pengujian inilah dapat diambil kesimpulan, apakah software yang diuji telah dapat diterima atau tidak. Dokumen dapat berupa kusioner uji test case.

Tabel 2. 1 Pilihan Jawaban UAT

1	Sangat : Mudah/Bagus/Sesuai/Jelas
2	Mudah/Bagus/Sesuai/Jelas
3	Netral
4	Cukup : Sulit/ Bagus/Sesuai/Jelas
5	Sangat : Sulit/ Bagus/Sesuai/Jelas

Tabel 2. 2 Bobot Nilai Jawaban

Jawaban	Bobot
A. Sangat : Mudah/Bagus/Sesuai/Jelas	5
B. Mudah/Bagus/Sesuai/Jelas	4
C. Netral	3
D. Cukup : Sulit/ Bagus/Sesuai/Jelas	2
E. Sangat : Sulit/ Bagus/Sesuai/Jelas	1

2.2.8.3 Pengujian *Usability*

Usability adalah sebuah analisa kualitatif yang menentukan seberapa mudah user menggunakan antarmuka suatu aplikasi, aplikasi dapat dikatakan usable apabila fungsinya dapat dijalankan secara efektif, efisien, dan memuaskan (Riyadi, 2019). Pengujian dalam usability dapat dilakukan dengan melibatkan pengguna atau tidak sama sekali. Pengujian dengan melibatkan pengguna menggunakan sistem serta permasalahan yang dihadapi. Pengujian dilakukan dengan cara menggunakan kuisioner yang dapat mengolah data yang berhubungan dengan *learnability*, *flexibility*, *effectiveness*, dan *attitude* dalam penggunaan aplikasi tersebut. Untuk penelitian ini melakukan pengujian usability pada aplikasi Sistem Informasi Rusunawa Untan untuk menentukan apakah pengujian tersebut dapat meningkatkan kinerja dan kelayakan dari aplikasi tersebut. Metode pengujian *usability* yang dipilih adalah metode kuisioner, pengujian ini meliputi admin Rusunawa Untan 1 orang dan mahasiswa aktif Untan sebanyak 9 orang.

Rumus perhitungan nilai presentasi usability yang akan digunakan untuk melakukan pengukuran terhadap aplikasi Sistem Informasi Rusunawa Untan disajikan pada persamaan (1). Nilai presentasi usability adalah rata-rata dari aspek *learnability*, *flexibility*, *effectiveness* dan *attitude*.

$$Usability (\%) = \frac{A+B+C+D}{4} \times 100\%$$

Keterangan :

A= presentasi nilai *learnability*

B= presentasi nilai *flexibility*

C= presentasi nilai *effectiveness*

D= presentasi nilai *attitude*