

BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terdahulu

Adapun penelitian terkait yang akan digunakan sebagai sumber acuan yang relevan dan terkini yaitu:

Berdasarkan penelitian yang dilakukan Arfianto dan Nugrahanti (2018) dengan judul Rancang Bangun Aplikasi Penjualan Perumahan Berbasis Web Pada CV. Grand Permata Residence Magetan, Arfianto dan Nugrahanti menyimpulkan bahwa Aplikasi ini telah membantu untuk mengetahui cara mendapatkan informasi dan transaksi penjualan perumahan yang akan di pesan.

Berdasarkan penelitian yang dilakukan Henny Leidiyana, dkk (2020) dengan judul Aplikasi pengendalian persediaan barang berbais web dengan metode Economic Order Quantity (EOQ) pada Bengkel Dunia Motor. Dunia motor merupakan sebuah bengkel sepeda motor khusus Honda yang memiliki permasalahan yaitu persediaan suku cadang slow moving yang banyak sementara kekurangan persediaan fast moving. Karyawan di bagian suku cadang juga sering bingung dalam mengambil suku cadang dari gudang karena informasi persediaan yang kurang jelas. Sebagai solusi maka dibuat aplikasi yang menerapkan perhitungan persediaan barang yang optimal menggunakan Economic Order Quantity (EOQ), yaitu metode yang digunakan untuk menentukan jumlah pembelian yang paling ekonomis yang dilaksanakan setiap kali pembelian.

Berdasarkan penelitian yang dilakukan Ayu Pratiwi, dkk (2021) dengan judul Rancang bangun aplikasi berbasis web dengan menggunakan metode Economic Order Quantity (EOQ) Pada PT. Yakin Sejahtera Mandiri Muara Enim. Penelitian ini, dibangun sebuah aplikasi pada telepon seluler pintar berbasis Web dengan menggunakan metode yaitu Economic Order Quantity. Tujuannya adalah membantu. Memudahkan PT. Yakin Sejahtera Mandiri. dapat menghitung berapa jumlah pemesanan ekonomis, safety stock, reorder point, dan frekuensi pembelian. Pada penulisan ini juga akan diterangkan tahapan pengerjaan, mulai dari proses analisa, desain, pengodean, pengujian menggunakan Adobe Flash CS6.

Berdasarkan penelitian dari Riski Septian Anwar, dkk (2019) dengan judul Perancangan aplikasi berbasis web dengan metode Economic Order Quantity (EOQ) di PT. Samawa Tirta alam sumbawa. Penelitian ini bertujuan untuk membuat perancangan aplikasi menggunakan metode Economic Order Quantity (EOQ) sebagai pengendalian suplai bahan baku di perusahaan PT. Samawa Tirta Alam yang di sebut Monitoring inventory, Dengan menerapkan metode ini dapat di ketahui frekuensi pemesanan yang optimal, safety stock, reorder point dan total biaya persediaan dalam mengadakan bahan baku.

Berdasarkan penelitian yang dilakukan oleh Ogana, dkk (2020) mengenai penerapan Metode Economic Quantity (EOQ) berbasis *Web* tentang penjualan produk rumah kreatif baturaja mitra binaan PT Semen Baturaja (Persero) Tbk, disimpulkan bahwa dengan adanya *Web* penjualan produk maka PT Semen Baturaja dapat menginformasikan penjualan produk ke masyarakat dengan lebih mudah. Tujuan dari penelitian Ogana, dkk yaitu menghasilkan aplikasi *Web* penjualan produk rumah kreatif baturaja mitra binaan PT Semen Baturaja (Persero) Tbk,

sehingga mempermudah PT Semen Baturaja dan menginformasikan dan menjual produk rumah kreatif baturaja.

Adapun perbedaan penelitian yang dilakukan penulis dengan penelitian terkait sebelumnya yaitu pada penelitian ini penulis menggunakan *Web* untuk melakukan pemesanan rumah (Pre Order) pada PT. Ira Widya Utama. Sehingga dengan adanya aplikasi ini, dapat memudahkan customer dalam melakukan pemesanan rumah (Pre Order) dan mempercepat proses pemesanan rumah atau (Pre Order) pada PT. Ira Widya Utama.

II.2. Landasan Teori

Berikut ini adalah landasan teori yang berkaitan dengan penelitian yang diambil dari beberapa jurnal terkait penelitian :

II.2.1. Perancangan Aplikasi

Perancangan Aplikasi adalah proses untuk sesuatu yang dikerjakan dengan yang bervariasi untuk membuat tampilan antarmuka program. (Azis, 2018).

Perancangan Aplikasi adalah program siap pakai yang dapat digunakan untuk menjalankan perintah - perintah dari pengguna aplikasi tersebut dengan tujuan mendapatkan hasil yang lebih akurat sesuai dengan tujuan pembuatan aplikasi tersebut, aplikasi mempunyai arti yaitu pemecahan masalah yang menggunakan salah satu teknik pemrosesan data aplikasi yang biasanya berpacu pada sebuah komputansi yang diinginkan atau diharapkan maupun pemrosesan data yang diharapkan.

II.2.2. Aplikasi

Pengertian aplikasi adalah suatu bagian dari perangkat lunak yang dapat digunakan untuk menyelesaikan masalah-masalah yang khusus yang dihadapi *user* dengan menggunakan kemampuan komputer. Sedangkan pengertian pemesanan adalah suatu proses seseorang atau organisasi untuk meyakinkan *customer* untuk memesan (Pre Order) rumah yang akan ditawarkan.

Aplikasi adalah program siap pakai yang dapat digunakan untuk menjalankan perintah-perintah dari pengguna aplikasi tersebut dengan tujuan mendapatkan hasil yang lebih akurat sesuai dengan tujuan pembuatan aplikasi tersebut. Aplikasi mempunyai arti yaitu pemecahan masalah yang menggunakan salah satu teknik pemrosesan data aplikasi yang biasanya berpacu pada sebuah komputasi yang diinginkan atau diharapkan maupun pemrosesan data yang diharapkan. Pengertian aplikasi secara umum adalah alat terapan yang difungsikan secara khusus dan terpadu sesuai kemampuan yang dimilikinya. Aplikasi merupakan suatu perangkat komputer yang siap pakai bagi *user*. (Armanda dan Putra, 2020 : 18).

II.2.3. Pre Order

Pre Order adalah transaksi jual beli online atau yang disebut dengan e-commerce terdapat beberapa hal yang saling berkaitan antara yang satu dengan yang lainnya. Diantaranya yang pertama adalah adanya subyek hukum, yang menjadi subyek hukum disini adalah pelaku usaha dan konsumen. Lalu kedua subyek hukum tadi melakukan transaksi melalui media teknologi informasi dengan layanan internet yang kemudian lahirnya perjanjian jual beli yang didasarkan

dengan alat bukti elektronik dan menghasilkan tanggung jawab bagi para pihak tersebut. Adanya alat bukti elektronik yang berupa dokumen elektronik merupakan hal yang penting untuk menghindari adanya penyalahgunaan perdagangan elektronik atau kejahatan dalam perdagangan elektronik. (Iis Melinda Putri, 2020 :17101163125)

II.2.4. Perumahan

Menurut UU No. 4 Tahun 1992 tentang Perumahan dan Permukiman, rumah adalah bangunan yang berfungsi sebagai tempat tinggal atau hunian dan sarana pembinaan keluarga. Menurut John F.C Turner, 1972, dalam bukunya *Freedom To Build* mengatakan, “Rumah adalah bagian yang utuh dari permukiman, dan bukan hasil fisik sekali jadi semata, melainkan merupakan suatu proses yang terus berkembang dan terkait dengan mobilitas sosial ekonomi penghuninya dalam suatu kurun waktu. Menurut Siswono Yudohusodo (Rumah Untuk Seluruh Rakyat, 1991: 432), rumah adalah bangunan yang berfungsi sebagai tempat tinggal atau hunian dan sarana pembinaan keluarga.

Jadi, selain berfungsi sebagai tempat tinggal atau hunian yang digunakan untuk berlindung dari gangguan iklim dan makhluk hidup lainnya, rumah merupakan tempat awal pengembangan kehidupan.

II.2.5. Metode Economic Order Quantity (EOQ)

Metode economic order quantity (EOQ) merupakan metode yang digunakan untuk analisis perencanaan dan pengendalian persediaan. Dari kajian literatur, banyak ahli telah memaparkan definisi Economic Order Quantity (EOQ). Metode EOQ diartikan sebagai jumlah pesanan yang bisa meminimumkan biaya persediaan secara keseluruhan, pembelian optimal dan berusaha mencapai tingkat persediaan yang semimumum mungkin dan biaya yang rendah (Trihudiyatmanto, 2017). Umami et al.,(2018) mendefinisikan metode EOQ sebagai nilai jumlah bahan yang dibutuhkan selama setiap kali melakukan pemesanan atau pembelian dengan menggunakan biaya paling ekonomis dan semimumum mungkin.

Metode pengendalian persediaan ini menjawab dua pertanyaan penting yakni kapan harus memesan dan berapa banyak harus memesan.

1. Rumus untuk mengetahui EOQ dengan menggunakan : $EOQ = \sqrt{(2 \cdot D \cdot S) / C_2}$.
2. Frekuensi Pemesanan Tahun Frekuensi pemesanan adalah jumlah pembelian yang harus dilakukan dalam satu periode. Rumus frekuensi pemesanan dapat dihitung dengan : $F = D / EOQ$
3. ROP (Reorder Point) ROP adalah tindakan perusahaan dalam menentukan kapan sebaiknya pemesanan dilakukan. Rumus ROP dapat diketahui dengan menggunakan : $ROP = D / 365 \times \text{Lead Time}$
4. SS (Safety Stock) Safety Stock adalah suatu persediaan yang dicadangkan sebagai pengaman dari kelangsungan proses produksi perusahaan untuk menghindari terjadinya kekurangan barang. Rumus

Safety Stock dapat diketahui dengan menggunakan : $SS = (\text{Pemakaian maksimum} - \text{Rerata}) \times \text{Lead Time}$ Dimana:

EOQ: Jumlah pemesanan ekonomis

F: Frekuensi pemesanan D: Jumlah permintaan selama 1 tahun

C: Biaya penyimpanan S: Biaya setiap melakukan pemesanan/barang

II.2.6. Web

Web dapat diartikan sekumpulan halaman yang terdiri dari beberapa laman yang berisi informasi dalam bentuk data digital baik berupa *text*, gambar, video, audio, animasi dan lainnya yang disediakan melalui jalur koneksi internet. (Armanda dan Putra, 2020 : 18).

Web merupakan kumpulan halaman *web* yang saling berkaitan. *Web* adalah sebuah sistem yang berisi beragam informasi baik berupa teks, gambar, audio maupun video dan dapat diakses melalui perangkat yang biasa disebut *web browser*. *Web* merupakan kumpulan informasi baik yang bersifat statis maupun dinamis yang terdiri dari halaman yang dibuat. (Oktasari dan Kurniadi, 2019 : 151).

Web merupakan salah satu aplikasi untuk mengelola data *online* yaitu yang terhubung dalam jaringan internet. *Web* adalah halaman situs sistem informasi yang dapat diakses secara cepat. *Web* adalah pengelolaan data sesuai dengan layar pengguna dan dikembangkan lagi menjadi *Responsive Web Design* (RWD), yang dapat mengoptimalkan kegunaan dari *web* itu sendiri dengan tampilan yang dapat dikelola sesuai dengan layar pengguna dan dapat dilakukan dimana saja dan kapan saja.

Dengan menggunakan *web*, pengguna bisa leluasa mengakses *website* menggunakan beraneka macam *gadget* (PC, laptop, *smartphone*, *tablet*, dan lain lain). (Anita, dkk, 2020 : 75).

II.2.7. *Hypertext Preprocessor (PHP)*

PHP singkatan dari *Hypertext Preprocessor* yang merupakan bahasa *server-side scripting* yang berada pada halaman *Hypertext Markup Language (HTML)* yang ditujukan untuk membantu *programmer* menulis halaman *web* secara mudah dan cepat. PHP bersifat *server-side* berarti semua pengerjaan *script* dilakukan di *server* dan hasilnya dikirim ke *browser*. PHP merupakan bahasa pemrograman yang memprogram situs *web* dinamis, yaitu mampu mengoperasikan *web* secara terus-menerus. *Hypertext Preprocessor (PHP)* merupakan perangkat lunak *open source* yang berada dibawah aturan *general purpose lisences (GPL)*. (Oktasari dan Kurniadi, 2019 : 150).

Hypertext Preprocessor atau yang sering disebut dengan PHP merupakan bahasa pemrograman yang bersifat *Open Source*. Bahasa pemrograman ini bersifat *server side*, maksudnya adalah tanpa adanya server yang berjalan beriringan dengan program atau aplikasi *script* PHP tidak dapat berjalan. PHP merupakan *script* yang disisipkan dalam *code HTML* untuk membuat halaman pada *website* bekerja secara otomatis dan dapat berfungsi sebagai pengolah data pada *server* dimana *script* tersebut dijalankan. PHP memiliki kemampuan untuk memisahkan diri dari kode HTML saat sebuah *website* dilakukan “*View Page Source*” oleh *user*. (Septyasari, dkk, 2018 : 215).

PHP adalah bahasa pemrograman yang sering disisipkan ke dalam HTML. PHP sendiri berasal dari kata *Hypertext Preprocessor*. Sejarah PHP pada awalnya merupakan kependekan dari *Personal Home Page* (Situs personal). PHP pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. Pada waktu itu PHP masih bernama Form Interpreted (FI), yang wujudnya berupa sekumpulan skrip yang digunakan untuk mengolah data formulir dari *web*. Bahasa pemrograman ini menggunakan sistem *server-side*. *Server-side programming* adalah jenis bahasa pemrograman yang nantinya *script*/program tersebut akan dijalankan/diproses oleh *server*. Kelebihannya adalah mudah digunakan, sederhana, dan mudah untuk dimengerti dan dipelajari. (Sugijanto, dkk, 2020 : 2).

II.2.8. Hypertext Markup Language (HTML)

Hypertext Markup Language (HTML) adalah sebuah bahasa markah yang digunakan untuk membuat sebuah halaman *web*, menampilkan berbagai informasi di dalam sebuah penjelajah *web* Internet dan pemformatan *hipertext* sederhana yang ditulis dalam berkas format ASCII agar dapat menghasilkan tampilan wujud yang terintegrasi. Dengan kata lain, berkas yang dibuat dalam perangkat lunak pengolah kata dan disimpan dalam format ASCII normal sehingga menjadi halaman *web* dengan perintah-perintah HTML. Bermula dari sebuah bahasa yang sebelumnya banyak digunakan di dunia penerbitan dan percetakan yang disebut dengan SGML (*Standard Generalized Markup Language*), HTML adalah sebuah standar yang digunakan secara luas untuk menampilkan halaman *web*. HTML saat ini merupakan standar Internet yang didefinisikan dan dikendalikan penggunaannya oleh World Wide Web Consortium (W3C).

HTML dibuat oleh kolaborasi Caillau TIM dengan Berners-lee Robert ketika mereka bekerja di CERN pada tahun 1989 (CERN adalah lembaga penelitian fisika energi tinggi di Jenewa). (Arthalia dan Prasetyo, 2020 : 96).

HTML adalah singkatan dari *Hypertext Markup Language*. HTML memungkinkan seorang *user* untuk membuat dan menyusun bagian paragraf, *heading*, link atau tautan, dan *blockquote* untuk halaman *web* dan aplikasi. (Sugijanto, dkk, 2020 : 2).

HTML ialah kepanjangan dari *Hypertext Markup Language*. Definisi HTML adalah bahasa yang digunakan untuk menulis halaman *web*. fungsi utama HTML ialah memberi perintah pada *browser* untuk melakukan manipulasi tampilan melalui *tag-tag* yang ditulis dalam HTML. (Rahmasari, 2019 : 415).

Ada 4 komponen HTML Menurut Yang, Linyi; Ng, Tin L.; Smyth, Barry; and Dong, Riuhai, (2020). :

- (1) Encoder transformator tingkat token
- (2) Multimedia fusi informasi
- (3) Pembuat encode transformator tingkat kalimat
- (4) Prediksi multi-tugas. Secara singkat, untuk memulai, fitur teks dan audio diekstraksi dari teks mentah/konten panggilan audio token teks adalah diekstraksi dari data teks dan dikodekan ke dalam vektor menggunakan amodel bahasa praterlatih, sementara rentang 27 audio yang berbeda fitur diekstraksi dari data audio menggunakan Praat.

II.2.9. MySQL

MySQL merupakan turunan dari *Structured Query Language* (SQL), SQL merupakan operasional basisdata yang dapat dikerjakan dengan mudah dan otomatis. MySQL bertugas mengatur dan mengelola data-data pada *database*. MySQL merupakan perangkat lunak relasi yang dapat digunakan secara gratis. MySQL adalah perangkat lunak sistem manajemen basis data pada *Structured Query Language* (SQL) yang *multiuser*. MySQL merupakan manajemen basis data yang paling stabil digunakan sebagai media penyimpanan data. (Oktasari dan Kurniadi, 2019 : 150).

MySQL adalah *database open source* terpopuler di dunia. Dengan kinerja, kehandalan dan kemudahan penggunaan yang terbukti, MySQL telah menjadi pilihan *database* terdepan untuk aplikasi berbasis *web*, yang digunakan oleh properti *web* profil tinggi termasuk Facebook, Twitter, YouTube, Yahoo! dan banyak lagi. Kata "SQL" dari "MySQL" adalah singkatan dari "*Structured Query Language*". SQL adalah bahasa standar yang paling umum digunakan untuk mengakses *database*. Bergantung pada lingkungan pemrograman yang kita pakai, kita bisa memasukkan SQL secara langsung (misalnya, untuk *men-generate* laporan), memasukkan pernyataan SQL ke dalam kode yang ditulis dalam bahasa lain, atau menggunakan API khusus yang dapat menyembunyikan sintaks SQL. (Arfianto dan Nugrahanti, 2018 : 175).

MySQL merupakan *database engine* atau *server database* yang mendukung bahasa *database* pencarian SQL. MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL atau DBMS yang *multithread*, *multi-user*. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis dibawah lisensi GNU

General Public License (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL. (Sugijanto, dkk, 2020 : 2).

II.2.10. *Unified Modelling Language* (UML)

UML yaitu satu kumpulan konvensi permodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem perangkat lunak yang terkait dengan objek. *UML* merupakan suatu kumpulan teknik terbaik yang telah terbukti sukses dalam memodelkan system yang besar dan kompleks. *UML* tidak hanya digunakan dalam proses pemodelan perangkat lunak, namun hampir dalam semua bidang yang membutuhkan pemodelan. (Andikos, 2019 : 39).

Unified Modeling Language (UML) adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak. UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem . Perancangan desain sistem yang akan dibangun menggunakan pemodelan *Unified Modelling System* (UML). Diagram-diagram yang digunakan *use case diagram*, *activity diagram*, *class diagram* dan *sequence diagram*. (Dicky Juliawan, Ratih Puspasari, dan Charles Jhony Manto Sianturi, 2017).

Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML adalah sebagai berikut :

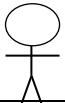
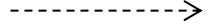
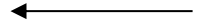
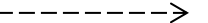
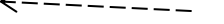
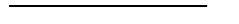
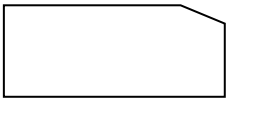
1. *Use Case Diagram*

Use case diagram menyajikan interaksi antara use case dan aktor. Dimana, aktor dapat berupa orang, peralatan atau sistem lain yang berinteraksi dengan sistem

yang sedang dibangun. Use case menggambarkan fungsional sistem atau persyaratan-persyaratan yang harus dipenuhi sistem dari pandangan pemakai. (2018): Jurnal Interkom.

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah use case merepresentasikan sebuah interaksi antara aktor dengan sistem. use case diagram dapat digambarkan dengan sumber-sumber pada Tabel II.1.

Tabel II.1. Simbol Use Case

Gambar	Nama	Keterangan
	<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
	<i>Depedency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (<i>independent</i>).
	<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
	<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara eksplisit.
	<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
	<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
	<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
	<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
	<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan prilaku yang lebih besar dari jumlah dan elemen-elemennya (<i>sinergi</i>).
	<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

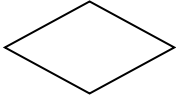
(Sumber : Andikos, 2019 : 39)

2. Diagram Aktivitas (*Activity Diagram*)

Activity diagram menggambarkan aliran fungsional sistem. Pada tahap pemodelan bisnis, diagram aktivitas dapat digunakan untuk menunjukkan aliran kerja bisnis. Dapat juga digunakan untuk menggambarkan aliran kejadian dalam use case. (2018): Jurnal Interkom.

Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. *Activity diagram* dapat digambarkan dengan simbol-simbol seperti pada tabel II.2.

Tabel II.2. Simbol *Activity Diagram*

Gambar	Nama	Keterangan
	<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain.
	<i>Action</i>	State dari sistem yang mencerminkan eksekusi dari suatu aksi.
	<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.
	<i>Activity Final</i>	Bagaimana objek dibentuk dan dihancurkan
	<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran
	<i>Decision Points</i>	Menggambarkan pilihan untuk pengambilan keputusan, <i>true</i> , <i>false</i> .

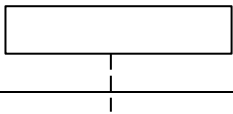
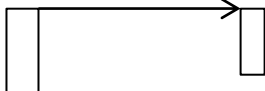
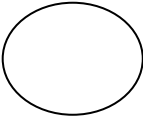
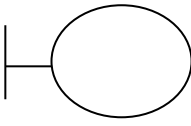
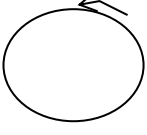
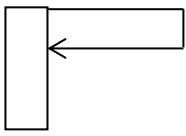
(Sumber : Andikos, 2019 : 39)

3. Diagram Urutan (*Sequence Diagram*)

Sequence diagram digunakan untuk menunjukkan aliran fungsionalitas dalam use case. Setiap objek yang terlibat dalam diagram *use case* digambarkan dengan garis putus-putus vertikal, kemudian message yang dikirim oleh objek digambarkan dengan garis horizontal secara kronologis dari atas ke bawah. (2018): Jurnal Interkom.

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, display, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence Diagram* dapat digambarkan dengan simbol-simbol seperti pada Tabel II.3.

Tabel II.3. Simbol *Sequence Diagram*

Gambar	Nama	Keterangan
	<i>Lifeline</i>	Objek entity, antarmuka yang saling berinteraksi.
	<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi.
	<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi.
	<i>Entity Class</i>	Merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.
	<i>Boundary Class</i>	Berisi kumpulan kelas yang menjadi <i>interface</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan formentry dan <i>form</i> cetak.
	<i>Control class</i>	Suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek.
	<i>Recursive</i>	Menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.

(Sumber : Andikos, 2019 : 39)

4. *Class Diagram* (Diagram Kelas)

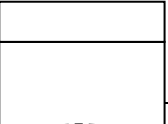
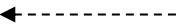
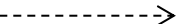
Class diagram menunjukkan interaksi antar kelas dalam sistem. Diagram kelas dibangun berdasarkan diagram *use case* dan diagram sekuensial yang telah dibuat sebelumnya. Kelas memiliki tiga area pokok yaitu : Nama (stereotype), Atribut dan Metode. Atribut dan Metode dapat memiliki salah satu sifat berikut:

- a) Private, tidak dapat dipanggil dari luar kelas yang bersangkutan.
- b) Protected, hanya dapat dipanggil oleh kelas yang bersangkutan dan anak-anak yang mewarisinya.
- c) Public, dapat dipanggil oleh siapa saja. (2018): Jurnal Interkom

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class diagram* menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain.

Class diagram secara khas meliputi : Kelas (*Class*), Relasi *Associations*, *Generalization* dan *Aggregation*, atribut (*Attributes*), operasi (*operation/method*) dan *visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan *Multiplicity* atau *Cardinality* *Class diagram* dapat digambarkan dengan simbol-simbol seperti pada Tabel II.4.

Tabel II.4. *Class Diagram*

Gambar	Nama	Keterangan
	<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
	<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
	<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
	<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor
	<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
	<i>Depedency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri akan mempegaruhi elemen yang bergantung padanya elemen yang tidak mandiri
	<i>Assocation</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya

Multiplicity	Penjelasan
1	Satu dan hanya satu
0..*	Boleh tidak ada atau 1 atau lebih
1..*	1 atau lebih
0..1	Boleh tidak ada, maksimal 1
n..n	Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimum 4

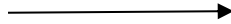
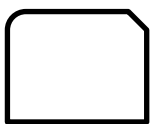
(Sumber : Andikos, 2019 : 39)

5. Statechart Diagram

Statechart diagram adalah jenis diagram yang digunakan dalam ilmu komputer dan bidang terkait untuk menggambarkan perilaku sistem. *Statechart diagram* mengharuskan sistem yang dijelaskan terdiri dari sejumlah negara terbatas; kadang-kadang, ini memang masalahnya, sementara di lain waktu ini adalah abstraksi yang masuk akal.

Statechart Diagram menggambarkan transisi dan perubahan keadaan (dari satu state ke state lainnya) suatu objek pada sistem sebagai akibat dari stimuli yang diterima. Pada umumnya *statechart diagram* menggambarkan class tertentu (satu class dapat memiliki lebih dari satu *statechart diagram*). StateChart Diagram dapat digambarkan dengan simbol-simbol seperti pada table II.5

Tabel II.5. Statechart Diagram

Gambar	Nama	Keterangan
	<i>State</i>	Nilai atribut dan nilai link pada suatu waktu tertentu, yang dimiliki oleh suatu objek.
	<i>Initial Pseudo State</i>	Bagaimana objek dibentuk atau diawali
	<i>Final State</i>	Bagaimana objek dibentuk atau dihancurkan.
	<i>Transition</i>	Bagaimana objek dibentuk dan dihancuSebuah kejadian yang memicu sebuah state objek dengan cara memperbaharui satu atau lebih nilai atributnya.
	<i>Node</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

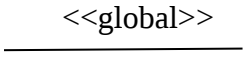
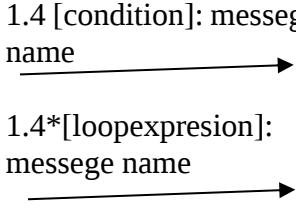
(Sumber : Andikos, 2019 : 39)

6. Collaboration Diagram

Collaboration diagram ialah sebuah diagram yang mengelompokkan pesan atau message dalam kumpulan sequence diagram menjadi sebuah diagram yang berbeda.

Collaboration diagram juga menggambarkan interaksi antar objek seperti *sequence diagram*, tetapi lebih menekankan pada peran masing-masing objek dan bukan pada waktu penyampaian message. Setiap message memiliki sequence number, di mana message dari level tertinggi memiliki nomor 1. Messages dari level yang sama memiliki prefiks yang sama. Collaboration Diagram dapat digambarkan dengan symbol-simbol seperti pada table II.6.

Tabel II.6. Collaboration Diagram

Gambar	Nama	Keterangan
	<i>Kelas peran</i>	peran Kelas menjelaskan bagaimana objek berperilaku. Gunakan simbol UML objek untuk mengilustrasikan peran kelas, tetapi tidak daftar atribut objek.
	<i>Asosiasi peran</i>	Asosiasi menggambarkan bagaimana peran asosiasi akan berperilaku diberikan situasi tertentu. Anda dapat menarik peran asosiasi menggunakan baris sederhana dilabeli dengan stereotip.
	<i>Pesan</i>	Tidak seperti urutan diagram, diagram kolaborasi tidak memiliki cara eksplisit untuk menunjukkan waktu dan bukannya jumlah pesan dalam rangka eksekusi. urutan penomoran dapat menjadi bersarang menggunakan sistem desimal Dewey.

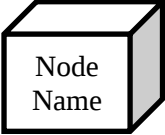
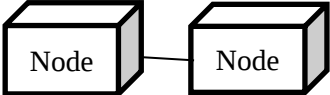
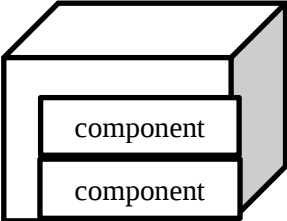
(Sumber : Andikos, 2019 : 39)

7. *Deployment Diagram*

Deployment diagram merupakan sekumpulan diagram dengan fungsi utamanya memetakan perangkat lunak ke dalam processing node.

Deployment/physical diagram menggambarkan detail bagaimana komponen dideploy dalam infrastruktur sistem, di mana komponen akan terletak (pada mesin, server atau piranti keras apa), bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi server, dan hal-hal lain yang bersifat fisik. Deployment Diagram dapat digambarkan dengan simbol-simbol seperti pada table II.7.

Tabel II.7. Deployment Diagram

Gambar	Nama	Keterangan
	<i>Komponen</i>	node adalah sumber daya fisik yang menjalankan kode komponen.
	<i>Asosiasi</i>	Asosiasi mengacu pada koneksi fisik antara node, seperti Ethernet.
	<i>Komponen dan Nodes</i>	komponen di dalam node yang menyebarkan mereka.

(Sumber : Andikos, 2019 : 39)

