

BAB II

TINJAUAN PUSTAKA

2.1 *Dasar Teori*

2.1.1 *Nibans Cake*

Nibans Cake merupakan toko yang menjual berbagai macam makanan, bertempat di komplek mandosi permai, jati asih, Bekasi selatan 17420. Didirikan oleh Ibu Yenni Elfiza Abbas sejak tahun 2019, Toko ini menjual berbagai macam makanan, mulai dari Roti, kue bolu, dan *pizza*, Toko Nibans Cake juga menerima pesanan dalam jumlah yang besar, seperti dihari lebaran, pernikahan, event acara dll. Ide untuk usaha dibidang produksi kue ditemukan Ibu Yenni Elfiza Abbas karena adanya pandemi di tahun 2019, dengan berjalannya usaha tersebut, semakin banyak pesanan yang didapati.

2.1.2 *Forecasting (Prediksi)*

Forecasting adalah sebuah perkiraan tentang suatu hal yang terjadi pada masa mendatang. Prediksi merupakan suatu alat bantu yang efektif dan efisien, digunakan sebagai dasar perencanaan ataupun pengambilan keputusan baik untuk jangka panjang maupun jangka pendek. Dalam hal keuangan, prediksi dapat digunakan sebagai dasar atau landasan dalam menentukan pengeluaran dan pengendalian biaya. Dari segi pemasaran, prediksi dapat membantu dalam perencanaan penjualan produk, kompensasi tenaga maupun keputusan penting lainnya. Pada aktifitas produksi dan operasi, hasil prediksi dapat digunakan untuk perencanaan kapasitas, fasilitas, produksi, penjadwalan, dan pengendalian persediaan. Pada setiap aktifitasnya, departemen atau organisasi selalu menentukan

sasaran maupun tujuan dan berusaha menduga faktor-faktor lingkungan, kemudian memilih tindakan yang diharapkan dapat menghasilkan pencapaian sasaran tujuan tersebut (Hilmy, M & Munawar, H. 2021). Prediksi dapat juga digunakan sebagai alat bantu untuk memperkirakan hal-hal lain seperti tingkat pertumbuhan ekonomi, tingkat pengangguran dan tingkat inflasi (Hilmy, M & Munawar, H. 2021).

2.1.3 Metode Forecasting

Pada dasarnya dalam horizon waktu prediksi, kondisi prediksi sangat berbeda. Dikarenakan bentuk pola dan berbagai hal lainnya sangat mempengaruhi hasil prediksi tersebut. Dalam menyelesaikan berbagai macam kasusnya, teknik prediksi telah mengalami perkembangan. Pada umumnya prediksi bisa dibedakan dari berbagai sudut pandang (Hilmy, M & Munawar, H. 2021).

Berdasarkan jangka waktu prediksi yang dibuat, prediksi dapat dibedakan menjadi dua, yaitu (Hilmy, M & Munawar, H. 2021) :

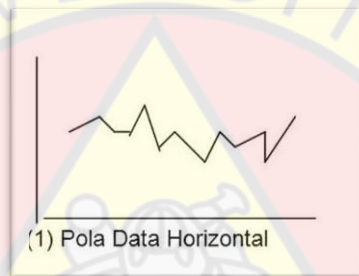
1. Prediksi jangka panjang, yaitu prediksi yang dilakukan yang penyusunan hasil prediksinya bertujuan untuk jangka waktu lebih dari satu setengah tahun.
2. Prediksi jangka pendek, yaitu prediksi yang dilakukan yang penyusunan hasil prediksinya bertujuan untuk jangka waktu kurang dari satu setengah tahun.

2.1.4 Data Time series

Analisis deret waktu diperkenalkan pada tahun 1970 oleh George E.P. Box dan Gwilym M. Deret waktu (*time series*) merupakan serangkaian data pengamatan yang terjadi berdasarkan indeks waktu secara berurutan dengan interval waktu tetap. Analisis deret waktu adalah salah satu tahap awal mengetahui kestasioneran

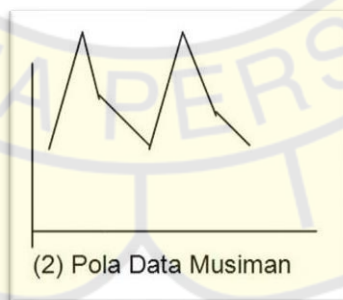
data dalam prosedur statistika yang diterapkan dengan menetapkan model deret waktu untuk meramalkan struktur probabilistik keadaan yang akan terjadi di masa yang akan datang dalam rangka pengambilan keputusan. Terdapat langkah penting dalam memilih suatu model deret waktu menurut Makridakis dkk (2003) model deret waktu mempertimbangkan jenis pola data, sehingga metode yang paling tepat dengan pola tersebut dapat diuji. Jenis pola data tersebut yaitu:

1. Pola Horizontal (H) terjadi bilamana nilai data berfluktuasi di sekitar nilai rata-rata yang konstan. Seperti yang ditunjukkan pada gambar 2.1



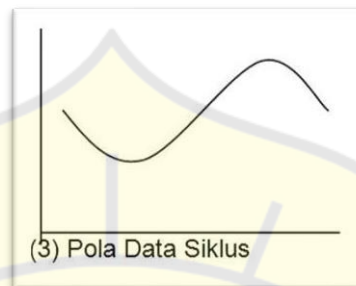
Gambar 2.1 Pola data horizontal (Muhammad Hilmy, 2021 : 12)

2. Pola Musiman (S) terjadi bilamana suatu deret dipengaruhi oleh faktor musiman (misalnya kuartal tahun tertentu, bulanan, atau hari-hari pada minggu tertentu).



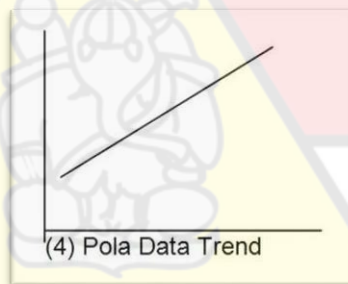
Gambar 2.2 Pola data musiman (Muhammad Hilmy, 2021 : 13)

3. Pola Siklis (C) terjadi bilamana suatu data dipengaruhi oleh fluktuasi ekonomi jangka panjang seperti yang berhubungan dengan siklus bisnis. Seperti yang ditunjukkan pada gambar 2.3



Gambar 2.3 Pola data siklus (Muhammad Hilmy, 2021 : 14)

4. Pola *Trend* (T) terjadi bilamana terdapat kenaikan atau penurunan sekuler jangka panjang dalam data. Seperti yang ditunjukkan pada gambar 2.4

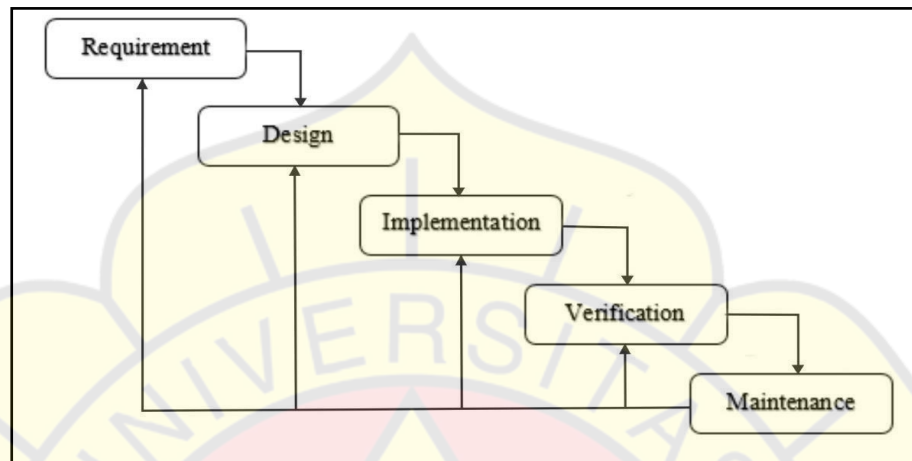


Gambar 2.4 Pola data *trend* (Muhammad Hilmy, 2021 : 13)

2.2 Metodologi Penelitian

Model *waterfall* adalah model klasik yang bersifat sistematis, berurutan dalam membangun software. Nama model ini sebenarnya adalah “*Linear Sequential Model*”. Model ini sering disebut juga dengan “*classic life cycle*” atau metode *waterfall*. Model ini termasuk ke dalam model *generic* pada rekayasa perangkat lunak dan pertama kali diperkenalkan oleh Winston Royce sekitar tahun 1970 sehingga sering dianggap kuno, tetapi merupakan model yang paling banyak dipakai dalam *Software Engineering* (SE). Model ini

melakukan pendekatan secara sistematis dan berurutan. Disebut dengan *waterfall* karena tahap demi tahap yang dilalui harus menunggu selesainya tahap sebelumnya dan berjalan berurutan.



Gambar 2.5 Metodologi *waterfall* (Tristanto, C. 2018)

Adapun penjelasan urutan dari tahapan-tahapan yang dimiliki metodologi *waterfall* adalah sebagai berikut:

2.2.1 Requirement (analisis kebutuhan)

Dalam langkah ini merupakan analisa terhadap kebutuhan sistem. Pengumpulan data dalam tahap ini bisa melakukan sebuah penelitian, wawancara atau *study literatur*. Seorang *system analisis* akan menggali informasi sebanyak mungkin dari user sehingga akan menjadi acuan dalam pembuatan sebuah sistem komputer yang diinginkan oleh *user* tersebut. Dari tahapan menghasilkan dokumen *user requirement* atau data yang berhubungan dengan keinginan *user* dalam pembuatan sistem. Dokumen inilah yang akan menjadi acuan *system analisis* untuk menterjemahkan kedalam bahasa pemrograman.

2.2.2 Design System

Proses *design* akan menterjemahkan syarat kebutuhan sebuah perancangan perangkat lunak yang dapat diperkirakan sebelum dibuat koding. Proses ini berfokus pada struktur data, arsitektur perangkat lunak, representasi *interface*, dan *detail* (algoritma) prosedural. Tahapan ini akan menghasilkan dokumen yang disebut *software requirement*. Dokumen inilah yang akan digunakan programmer untuk melakukan aktivitas pembuatan sistemnya.

2.2.3 Implementation & Testing (implementasi dan uji coba program)

Tahapan inilah yang merupakan tahapan secara nyata dalam mengerjakan suatu sistem. Dalam artian penggunaan computer akan dimaksimalkan dalam tahapan ini. Tujuan testing adalah menemukan kesalahan-kesalahan terhadap sistem tersebut dan kemudian bisa diperbaiki.

2.2.4 Integration & Testing (Penerapan / Pengujian Program)

Tahapan ini bisa dikatakan *final* dalam pembuatan sebuah sistem. Setelah melakukan analisa, design dan pengkodean maka sistem yang sudah jadi akan digunakan oleh *user*.

2.2.5 Operation & Maintenance (Pemeliharaan)

Perangkat lunak yang susah disampaikan kepada pelanggan pasti akan mengalami perubahan. Perubahan tersebut bisa karena mengalami kesalahan karena perangkat lunak harus menyesuaikan dengan lingkungan (peripheral atau sistem operasi baru) baru, atau karena pelanggan membutuhkan perkembangan fungsional.

2.3 Metode ARIMA

Model-model *Autoregressive Integrated Moving Average* (ARIMA) telah dipelajari secara mendalam oleh George Box dan Gwilym Jenkins (1976) dan nama mereka berdua yang disinonimkan dengan proses ARIMA yang diterapkan untuk proses deret berkala, peramalan dan kontrol atau pengendalian. Model Autoregresif (AR) pertama kali dikenalkan oleh Yule (1926) dan kemudian dikembangkan oleh Walker (1931), sedangkan model *Moving Average* (MA) pertama kali digunakan oleh Slutsky (1937). Akan tetapi Wold-lah (1938) yang menghasilkan dasar-dasar teoritis dari proses kombinasi ARMA. Wold membentuk model ARMA yang berkembang dalam tiga arah (untuk proses AR, MA, dan ARMA campuran). Perluasan dari hasil tersebut mencakup deret berkala musiman (seasonal time series) dan pengembangan sederhana yang mencakup proses-proses nonstasioner.

Box dan Jenkins berhasil dalam menentukan informasi relevan yang diperlukan untuk memahami dan memakai model-model ARIMA untuk deret berkala. Menurut Putri dan Anggraeni (2018) penggunaan metode ARIMA karena metode ini memiliki sifat yang fleksibel (mengikuti pola data), dan memiliki tingkat akurasi peramalan yang cukup tinggi.

2.3.1 Model Box-Jenkins (ARIMA) dibagi kedalam tiga kelompok, yaitu:

Model *Autoregressive* (AR), *Moving Average* (MA), dan model campuran ARIMA (*autoregressive moving average*) yang mempunyai karakteristik dari dua model pertama.

1. *Proses Autoregressive Model* (AR).

$$AR(p) (1 - \phi_1 B - \phi_2 B^2 - \dots - \phi_p B^p) Y_t = \delta + \epsilon_t$$

dimana:

δ = nilai konstan ϕ

p = parameter auto regressive

ε = nilai error pada saat t

2. Proses Moving Average (MA).

$$Y_t = \mu + (1 - \theta_1 B - \theta_2 B^2 - \dots - \theta_q B^q) \varepsilon_t$$

dimana:

μ = nilai konstan θ

q = parameter moving average

ε = nilai error pada saat t

3. Model campuran.

Berdasarkan AR (1) dan MA (1) akan diperoleh bentuk umum sebagai berikut:

ARMA (1,1)

$$Y_t = \mu + \phi_1 Y_{t-1} + \varepsilon_t - \theta_1 \varepsilon_{t-1}$$

$$(1 - \phi_1 B) Y_t = \mu + (1 - \theta_1 B) \varepsilon_t$$

Apabila non-stasioneritas ditambahkan pada campuran proses ARMA, maka model umum ARIMA (p,d,q) terpenuhi. Persamaan untuk kasus sederhana ARIMA (1,1,1) adalah berikut:

$$(1 - B) (1 - \phi_1 B) Y_t = \mu + (1 - \theta_1 B) \varepsilon_t.$$

Langkah-langkah dalam penerapan ARIMA sebagai berikut:

1. Identifikasi Model Model ARIMA hanya dapat digunakan untuk deret waktu yang stasioner. Oleh karenanya hal pertama yang dilakukan adalah menyelidiki apakah data deret waktu sudah stasioner atau belum. Jika data

deret waktu belum stasioner, yang harus dilakukan adalah memeriksa pada pembedaan (*differencing*) berapa data akan stasioner.

2. Identifikasi ACF dan PACF

Di samping menentukan nilai d , pada tahap ini juga ditentukan berapa jumlah nilai lag residual (q) dan nilai lag dependen (p) yang digunakan dalam model.

Alat utama yang digunakan untuk mengidentifikasi q dan p adalah ACF dan PACF (*Partial Auto Correlation Function*/Koefisien Autokorelasi Parsial), dan correlogram yang menunjukkan plot nilai ACF dan PACF terhadap lag.

Koefisien auto korelasi parsial mengukur tingkat keeratan hubungan antara X_t dan X_{t-k} , sedangkan pengaruh dari time lag 1, 2, 3, ..., $k-1$ dianggap konstan.

3. Pemilihan Model ARIMA terbaik.

Dari hasil identifikasi stasioneritas dan identifikasi ACF dan PACF maka akan diperoleh beberapa alternatif model ARIMA.

4. *Diagnostic Checking*

Setelah melakukan estimasi dan mendapatkan penduga parameter, agar model sementara dapat digunakan untuk peramalan, perlu dilakukan uji kelayakan terhadap *model* tersebut. Tahap ini disebut *diagnostic checking*, di mana pada tahap ini diuji apakah spesifikasi model sudah benar atau belum.

5. Peramalan/*Forecasting*

Setelah model terbaik diperoleh, selanjutnya peramalan dapat dilakukan.

Dalam berbagai kasus, peramalan dengan metode ini lebih dipercaya daripada peramalan yang dilakukan dengan model ekonometri tradisional.

2.4 Double Exponential Smoothing

Metode *Double Exponential Smoothing* merupakan suatu metode prediksi dengan memberi nilai pembobot pada beberapa periode atau pengamatan sebelumnya untuk memprediksi nilai pada periode yang akan datang (Makridakis *dkk*, 1999). Metode *Double Exponential Smoothing* ada dua macam yaitu dari *Holt* dan *Brown*. *Double Exponential Smoothing Brown* yaitu pemulusan eksponensial yang hanya menggunakan satu parameter, sedangkan *Double Exponential Smoothing Holt* menggunakan 2 parameter (α dan β atau γ) yang memuluskan nilai *trend* (kecenderungan yang konsisten) dengan parameter yang berbeda dari parameter yang digunakan dari deret asli (Makridakis *dkk*, 1999).

Berikut adalah penjelasan dari ketiga parameter, antara lain:

1. Alpha (α) merupakan parameter yang mengontrol pembobotan relatif pada pengamatan yang baru dilakukan. Nilai alpha berkisar dari 0 sampai 1. Parameter alpha digunakan semua model.
2. Beta (β) merupakan parameter yang mengontrol pembobotan relatif pada pengamatan yang baru dilakukan untuk mengestimasi kemunculan trend seri. Nilai beta berkisar dari 0 sampai 1. Nilai semakin besar menunjukkan pemberian bobot yang semakin besar pada pengamatan terbaru. Parameter beta digunakan pada model yang memiliki komponen trend linier atau *eksponensial* dengan tidak memiliki variasi musiman.
3. Gamma (γ) merupakan parameter yang mengontrol pembobotan relatif pada pengamatan yang baru dilakukan untuk mengestimasi kemunculan variasi musiman. Parameter gamma digunakan pada model yang memiliki variasi musiman.

Persamaan dari rumus *double exponential smoothing Holt* didapatkan dengan menggunakan tiga persamaan, meliputi (Makridakis *dkk*, 1999) :

$$S_t = \alpha Y_t + (1 - \alpha)(S_{t-1} + T_{t-1}) \quad (1)$$

Dimana :

S_t = nilai pemulusan eksponensial bulan yang dihitung

S_{t-1} = nilai pemulusan eksponensial bulan sebelumnya

α = konstanta pemulusan untuk data ($0 \leq \alpha \leq 1$)

Y_t = nilai aktual pada periode t

t = periode yang dihitung

Langkah selanjutnya yaitu persamaan untuk memuluskan *trend*, meliputi:

$$T_t = \beta(S_t - S_{t-1}) + (1 - \beta)T_{t-1} \quad (2)$$

Dimana :

S_t = nilai pemulusan eksponensial bulan yang dihitung

S_{t-1} = nilai pemulusan eksponensial bulan sebelumnya

β = konstanta pemulusan untuk estimasi *trend* ($0 \leq \beta \leq 1$)

T_t = estimasi *trend* bulan yang dihitung

T_{t-1} = estimasi *trend* bulan sebelumnya

t = periode yang dihitung

Persamaan yang digunakan untuk membuat prediksi pada periode yang akan datang adalah:

$$F_{t+p} = S_t + T_t p \quad (3)$$

Dimana :

S_t = nilai pemulusan eksponensial pada periode ke t

α = parameter pemulusan untuk data ($0 \leq \alpha \leq 1$)

β = parameter pemulusan untuk estimasi *trend* ($0 \leq \beta \leq 1$)
 Y_t = nilai aktual pada periode t

T_t = estimasi *trend* pada periode ke t

p = jumlah periode ke depan yang akan diprediksi

F = hasil prediksi

t = periode yang dihitung

2.5 *Mean Absolute Percentage Error (MAPE)*

Mean Absolute Percent Error atau nilai kesalahan absolut rata-rata yaitu melakukan perhitungan perbedaan antara data asli dan data hasil prediksi, kemudian hasil perbedaan tersebut diabsolutkan lalu menghitungnya ke dalam bentuk presentase terhadap data asli. Hasil persentase tersebut kemudian didapatkan nilai mean-nya (Aritonang, 2002). Adapun untuk menghitung MAPE yaitu :

$$MAPE = \sum_{t=1}^n \left(\frac{y_t - \hat{y}_t}{\hat{y}_t} \right) \times 100\%$$

Keterangan:

y_t = data nilai asli pada periode ke- t

\hat{y}_t = data nilai ramalan pada periode ke- t

\hat{y}_t = banyaknya periode waktu


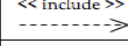
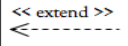
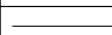


2.6 UML (*Unified Modelling Language*)

Pemodelan perangkat lunak dalam pembangunan perangkat lunak sangatlah penting, karena pemodelan berfungsi untuk mempermudah langkah berikutnya dari pengembangan sebuah sistem informasi sehingga lebih terencana. Perangkat pemodelan adalah suatu model yang digunakan untuk menguraikan sistem menjadi bagian-bagian yang dapat diatur dan menkomunikasikan ciri konseptual dan fungsional kepada pengamat. Salah satu perangkat pemodelan adalah *Unified Modelling Language* (UML). UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasi, menggambarkan, membangun dan dokumentasi dari sistem perangkat lunak. (Salahuddin & S, 2013)

2.6.1 *Use Case Diagram*

Use Case adalah pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara garis besar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu. Ada dua hal utama pada *use case* yaitu pendefinisian *actor* dan *use case*. Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri. Berikut ini adalah simbol-simbol yang ada pada diagram *use case*. (Salahuddin & S, 2013)

Tabel 2.1 Penjelasan Use Case Diagram


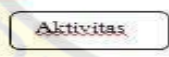



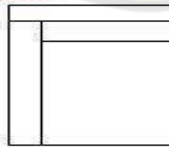
NO	GAMBAR	NAMA	KETERANGAN
1		Actor	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan use case.
2		Include	Menspesifikasikan bahwa use case sumber secara eksplisit.
3		Extend	Menspesifikasikan bahwa use case target memperluas perilaku dari use case sumber pada suatu titik yang diberikan.
4		Association	Apa yang menghubungkan antara objek satu dengan objek lainnya.
5		System	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
6		Use Case	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor

(Sumber : Salahuddin & S, 2013)

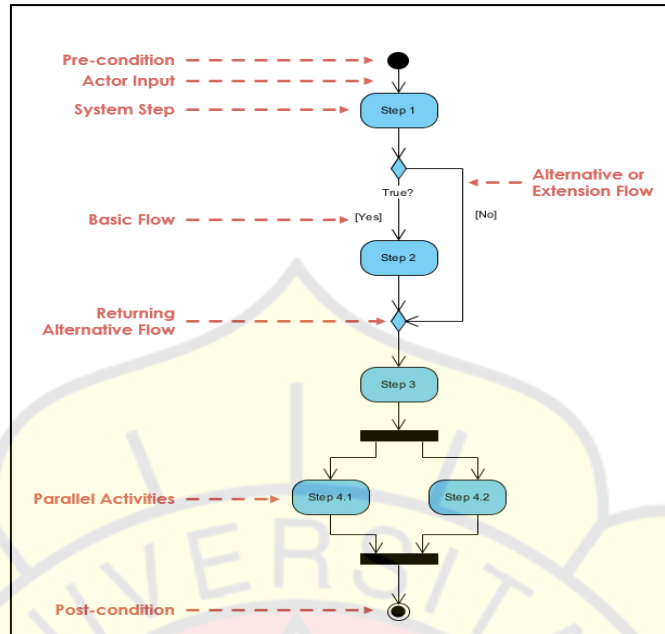
2.6.2 Activity Diagram

“Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak.” (Shalahuddin, 2013)

Tabel 2.2 Penjelasan Activity Diagram

No.	Simbol	Deskripsi
1.		Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
2.		Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3.		Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
4.		Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
5.		Status akhir yang dilakukan oleh sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
6.		Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

(Sumber : Salahuddin & S, 2013)



Gambar 2.6 Contoh Activity Diagram (Salahuddin & S, 2013)

2.6.3 Sequence Diagram

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek. Berikut adalah simbol-simbol yang ada pada diagram sekuen. (Salahuddin & S, 2013)

2.7 User Interface

Menurut I.Rochmawati (2019) “User Interface merupakan mekanisme komunikasi antara *user* dengan system. User Interface dapat menerima informasi dari *user* dan menerima informasi dari *user* untuk membantu mengarahkan alur penelusuran masalah sampai ditemukan pemecahannya”.

2.8 Implementation

Pada tahap ini, sistem pertama kali dikembangkan di program kecil yang disebut *unit*, yang terintegrasi dalam tahap selanjutnya. Setiap *unit* dikembangkan dan diuji untuk fungsionalitas yang disebut sebagai *unit testing*.

2.8.1 HTML

HTML adalah kependekan dari *Hyper Text Markup Language*, yaitu sebuah bahasa markup yang digunakan untuk membuat sebuah halaman *web*. HTML inilah yang menyusun sebuah halaman web menjadi sebagaimana yang kitalihat melalui *browser* (penjelajah internet). (Anhar, 2014)

2.8.2 CSS

CSS adalah bahasa *Cascading Style Sheet* dan biasanya digunakan untuk mengatur tampilan elemen yang tertulis dalam bahasa markup, seperti HTML. CSS berfungsi untuk memisahkan konten dari tampilan visualnya di situs. (Anhar,2014)

2.8.3 PHP

PHP adalah *Hypertext Preprocessor* adalah bahasa skrip yang dapat ditanamkan atau disisipkan ke dalam HTML. PHP banyak dipakai untuk memprogram situs web dinamis. PHP dapat digunakan untuk membangun sebuah CMS. (Supono & Putratama, 2018)

2.8.4 XAMPP

XAMPP yang merupakan singkatan dari *Apache, MySQL, PHP* dan *Perl* sedangkan huruf "X" dimaksudkan sebagai suatu software yang dapat dijalankan di empat OS utama seperti Windows, Mac OS, Linux dan Solaris. Istilah ini seringkali disebut dengan *cross platform (software multi OS)*.(Sutarman , 2010)

2.8.5 Apache

Server HTTP Apache atau Server Web/WWW Apache adalah server *web* yang dapat dijalankan di banyak sistem operasi (*Unix, BSD, Linux, Microsoft Windows* dan *Novell Netware* serta *platform* lainnya) yang berguna untuk melayani dan memfungsikan situs *web*.(Sutarman , 2010)

2.8.6 *MySQL*

Menurut Enterprise (2018:3), “*MySQL* adalah sebuah *software database*. *MySQL* merupakan tipe data relasional yang artinya *MySQL* menyimpan datanya dalam bentuk table-tabel yang saling berhubungan. Keuntungan menyimpan data di *database* adalah kemudahannya dalam penyimpanan dan menampilkan data karena dalam bentuk tabel.”

Menurut Enterprise (2018), “*MySQL (My Structured Query Language)* atau yang biasa dibaca, ai-se-kuel adalah sebuah program pembuatan dan pengelola database atau yang sering disebut dengan DBMS (*Database Management System*). Kelebihan lain dari *MySQL* adalah menggunakan bahasa query (permintaan) standar SQL (*Structured Query Language*). SQL adalah suatu bahasa permintaan yang terstruktur”.

2.8.7 *PHPMysqlAdmin*

PhpMyAdmin adalah perangkat lunak bebas yang ditulis dalam bahasa pemrograman administrasi *MySQL* melalui website Jejaring Jagat Jembar (*World Wide Web*) . *phpMyAdmin* mendukung berbagai operasi *MySQL*, diantaranya (mengelola basis data, tabel-tabel, bidang (*fields*), relasi (*relations*), indeks, pengguna (*users*), perizinan (*permissions*), dan lain-lain.(Sutarman , 2010)

2.8.8 *Sublime Text*

Sublime Text adalah editor kode sumber lintas-platform milik dengan antarmuka pemrograman aplikasi *Python* (API). Ini secara asli mendukung banyak bahasa pemrograman dan bahasa markup, dan fungsi dapat ditambahkan oleh pengguna dengan plugin, biasanya dibangun komunitas dan dikelola di bawah lisensi perangkat lunak bebas.(Sutarman , 2010)