

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pada Desember 2019, beberapa kasus penyakit pernapasan yang parah dilaporkan muncul di Wuhan, Provinsi Hubei, Cina. Hingga 25 Januari 2020, 1.975 kasus telah dilaporkan sejak pasien dirawat di rumah sakit pada 12 Desember 2019 (Wu et al., 2020). Penyakit pernapasan tersebut disebabkan oleh virus yang ditetapkan oleh *International Committee on Taxonomy of Viruses* (ICTV) sebagai *Severe Acute Respiratory Syndrome Coronavirus 2* (SARS-CoV-2) pada 11 Februari 2020 (Yang & Wang, 2020). Wabah SARS-CoV-2 awalnya diduga dimulai melalui penularan dari hewan ke manusia maupun sebaliknya yang berkaitan dengan pasar makanan laut (*seafood market*) di Wuhan, Cina (Wu et al., 2020).

Pada 30 Januari 2020, Komisi Darurat *World Health Organization* (WHO) mendeklarasikan darurat kesehatan masyarakat global berdasarkan peningkatan laporan China dan internasional (P. Velavan & Christian G., 2020). Penyakit *coronavirus* baru yang disebabkan oleh SARS-CoV-2 tersebut diberi nama *Coronavirus Disease 2019* (COVID19) (Yang & Wang, 2020).

Menghadapi jenis penyakit baru ini dan rekomendasi WHO, sebagian besar negara di dunia melakukan berbagai tindakan pencegahan penularan seperti *lockdown*, penerapan *social distancing* yaitu menjaga jarak terhadap orang yang terinfeksi maupun orang yang sehat, karantina paksa bagi mereka yang bepergian ke negara – negara tersebut, penutupan pendidikan formal dari taman kanak – kanak hingga universitas serta penutupan pendidikan non formal dari tempat kursus

hingga lembaga pelatihan khusus, pemaksaan penggunaan masker, larangan berkumpul hingga 4 orang di tempat yang sama, penutupan semua acara olahraga, sering mencuci tangan, pembatasan perjalanan domestik dan internasional telah menyebabkan penurunan industri penghasil bisnis negara yang mencakup transportasi udara, transportasi laut, penanganan makanan, sektor akomodasi, hiburan dan rekreasi (Berihe et al., 2020).

Di Indonesia, penyebaran virus ini dimulai sejak tanggal 02 Maret 2020, diduga berawal dari salah satu warga negara Indonesia yang melakukan kontak langsung dengan warga negara asing (Nalini, 2021). Dimana, salah satu tindakan pemerintah Indonesia dalam mencegah penularan COVID-19 adalah dengan menerapkan Pemberlakuan Pembatasan Kegiatan Masyarakat (PPKM).

Terjadinya peningkatan kasus COVID-19 telah memengaruhi perekonomian dunia termasuk Indonesia. Pandemi COVID-19 yang telah ditetapkan PBB, berdampak pada sektor transportasi, pariwisata, perdagangan, kesehatan dan lainnya. Kebijakan *lock down* diambil oleh berbagai negara untuk mencegah penyebaran COVID-19 dampak dari kebijakan tersebut menyebabkan perekonomian terhambat dan memberi tekanan pada pertumbuhan ekonomi dunia ke depan termasuk pertumbuhan ekonomi Indonesia (Kementerian Keuangan RI, 2020).

Oleh karena itu, untuk meminimalisir kerugian material maupun immaterial, membantu kesiapan pemerintah Indonesia dalam melakukan tindakan pencegahan dengan cepat dan tepat serta agar dapat membantu masyarakat umum untuk mendapatkan informasi COVID-19 dan prakiraan perkembangannya maka pemerintah perlu mendasarkan kebijakan dengan pertimbangan perkembangan

kasus untuk jangka waktu kedepan yaitu dengan cara memprakirakan perkembangan kasus aktif COVID-19 di Indonesia. Dengan menggunakan layanan REST API untuk mengumpulkan data kasus aktif COVID-19 di Indonesia, model prakiraan dapat dikembangkan dengan menggunakan metode *single exponential smoothing*.

REST API biasanya mengumpulkan data dengan mengakses beberapa *endpoint*. Misalnya, membiarkan klien ingin mengambil data peneliti yang mencakup id (*identifier*) yang bersifat unik, judul penelitian, dan anggota proyek penelitian. Server mengirimkan representasi dari keadaan sumber daya yang diminta ke klien yang sebagian besar dalam format JSON (*JavaScript Object Notation*) atau XML (*eXtensible Markup Language*) (Lawi et al., 2021).

Metode *single exponential smoothing* merupakan teknik peramalan rata-rata bergerak dengan pembobotan dimana data diberi bobot oleh sebuah fungsi *exponential* (L. W. S. Rahayu & B. G. Anandita, 2019).

Berdasarkan permasalahan yang telah disebutkan di atas, maka penulis melakukan penelitian dengan judul “IMPLEMENTASI LAYANAN REST API SEBAGAI APLIKASI PRAKIRAAN PERKEMBANGAN COVID-19 DI INDONESIA MENGGUNAKAN METODE SINGLE EXPONENTIAL SMOOTHING BERBASIS WEB”.

1.2 Rumusan Masalah

Berdasarkan uraian latar belakang di atas, penulis merumuskan masalah:

1. Bagaimana mengimplementasikan REST API sebagai media layanan penyedia data COVID-19 di Indonesia?

2. Bagaimana memprakirakan perkembangan COVID-19 di Indonesia dengan menggunakan metode *single exponential smoothing*?

1.3 Batasan Masalah

Agar pembahasan lebih terarah dan sesuai dengan judul tugas skripsi yang telah ditentukan, penulis hanya membahas pokok-pokok bahasan sebagai berikut :

1. Aplikasi yang dibangun merupakan aplikasi berbasis *web*.
2. Pembuatan aplikasi menggunakan bahasa pemrograman JavaScript.
3. Data kasus COVID-19 yang digunakan merupakan data kasus COVID-19 di Indonesia yang terkonfirmasi dengan tanggal dari tahun 2020 sampai dengan tahun 2022.
4. Data yang dihitung merupakan kumpulan dari data provinsi yang didapatkan dari *public API* yang kemudian diolah menjadi himpunan data COVID-19 seluruh provinsi di Indonesia (nasional).
5. Data COVID-19 merupakan data yang dinamis bukan data statis yang di-*update* secara *real-time* sehingga dapat berubah sewaktu-waktu dan dapat mengalami perbedaan antara aplikasi dengan perhitungan yang telah dilakukan pada penelitian ini.
6. Nilai α (alpha) yang digunakan pada perhitungan dengan menggunakan metode *single exponential smoothing* adalah 0,9.

1.4 Tujuan dan Manfaat Penelitian

1.4.1 Tujuan Penelitian

Tujuan yang ingin dicapai dalam penelitian tugas skripsi ini adalah:

1. Untuk mengimplementasikan REST API sebagai media layanan penyedia data COVID-19.
2. Untuk memprakirakan perkembangan COVID-19 di Indonesia dengan menggunakan metode *single exponential smoothing*.

1.4.2 Manfaat Penelitian

Adapun manfaat penelitian ini adalah sebagai berikut:

1. Penelitian ini dapat memperluas pilihan metode-metode yang dapat digunakan untuk memprakirakan perkembangan kasus aktif COVID-19.
2. Tugas skripsi ini dapat menambah referensi dalam bidang aplikasi prakiraan, khususnya menggunakan REST API sebagai media layanan data COVID-19 serta metode *single exponential smoothing* sebagai teknik perhitungan prakiraan.

1.5 Metodologi Penelitian

Metodologi penelitian yang digunakan pada penelitian ini adalah:

1. Studi Kepustakaan

Pada tahap ini dilakukan studi kepustakaan yaitu proses mengumpulkan informasi dengan melakukan pengumpulan, mempelajari, dan membaca berbagai bahan referensi yang berkaitan dengan aplikasi, prakiraan, REST

API, dan *single exponential smoothing* dari buku, artikel, paper, jurnal, makalah, internet dan sumber lainnya.

2. Analisis dan Perancangan

Pada tahap ini dilakukan analisis spesifikasi aplikasi prakiraan perkembangan COVID-19 dan melakukan perancangan aplikasi prakiraan perkembangan COVID-19.

3. Implementasi

Pada tahap ini dilakukan pengkodean aplikasi prakiraan perkembangan COVID-19 sesuai dengan analisis spesifikasi dan perancangan yang telah ditentukan.

4. Pengujian Aplikasi

Pada tahap ini dilakukan pengujian terhadap aplikasi prakiraan perkembangan COVID-19 yang sudah dibuat, dan bagaimana keakuratan dari sistem aplikasi yang dibuat.

5. Penyusunan Laporan

Pada tahap ini dilakukan penulisan dokumentasi dan laporan dari aplikasi prakiraan perkembangan COVID-19 yang dikembangkan.

1.6 Sistematika Penulisan

Sistematika penulisan tugas skripsi ini dibagi atas beberapa bab, di mana masing-masing bab dibagi atas beberapa sub agar mempermudah penjelasan mengenai penelitian yang dilakukan dan mempermudah pembaca dalam memahami isi penelitian. Adapun sistematika penulisan tugas skripsi ini adalah sebagai berikut:

BAB 1 PENDAHULUAN

Pendahuluan berisi tentang latar belakang masalah, rumusan masalah, tujuan, manfaat, batasan masalah, metodologi penelitian dan sistematika penulisan dalam pembuatan tugas skripsi.

BAB 2 LANDASAN TEORI

Bab ini berisi teori-teori pengetahuan dasar yang di peroleh dari studi kepustakaan atau literatur dan penelitian terdahulu yang digunakan untuk memahami permasalahan yang dibahas pada penelitian ini. Teori-teori pengetahuan dasar yang disajikan antara lain tentang aplikasi, prakiraan, REST API, dan *single exponential smoothing*.

BAB 3 METODE PENELITIAN

Dalam bab ini penulis akan menguraikan jenis dan lokasi penelitian, instrumen penelitian yang digunakan dalam membuat aplikasi prakiraan perkembangan COVID-19 menggunakan REST API dan metode *single exponential smoothing*.

BAB 4 HASIL DAN PEMBAHASAN

Bab ini berisi tentang hasil dan pembahasan dari aplikasi prakiraan perkembangan COVID-19 menggunakan REST API dan metode *single exponential smoothing* yang telah dibuat.

BAB 5 KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan dari keseluruhan uraian bab-bab penulisan skripsi dan saran yang diajukan untuk pengembangan lebih lanjut.

BAB II

LANDASAN TEORI

2.1 Perancangan Sistem

2.1.1 Pengertian Perancangan

Jogiyanto dalam (Hadi, 2017) menjelaskan bahwa, “perancangan adalah penggambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah dari suatu kesatuan yang utuh dan berfungsi”. Sedangkan berdasarkan kesimpulan yang di yang sampaikan oleh Murti (2016) bahwa, “Perancangan sistem merupakan spesifikasi baru yang menggambarkan secara rinci untuk tahap lanjutan dari proses menganalisa sistem”.

Berdasarkan definisi – definisi di atas dapat disimpulkan bahwa, perancangan sistem merupakan suatu kegiatan yang bertujuan untuk menyelesaikan masalah dengan menjadikan beberapa elemen menjadi satu.

2.1.2 Definisi Sistem

“Sistem adalah sekelompok unsur atau elemen yang saling terhubung secara bersama sama dalam mencapai suatu tujuan dalam pengelolaan sistem” (Murti, 2016). Supriati dalam (Hadi, 2017) menjelaskan definisi sistem bahwa, “sistem merupakan kumpulan atau unsur dari sub – sub sistem/komponen - komponen/prosedur – prosedur baik fisik maupun fisik yang mempunyai fungsi dan prosedur tertentu, saling bekerja sama secara harmonis untuk mencapai suatu tujuan tertentu”.

Menurut Afni (2017) menyatakan bahwa, “Sistem berasal dari bahasa Latin (*systema*) dan bahasa Yunani (*systema*) adalah sekumpulan unsur / elemen yang saling berkaitan dan saling mempengaruhi dalam melakukan kegiatan bersama untuk mencapai suatu tujuan”. Dengan demikian berdasarkan definisi diatas dapat disimpulkan bahwa sistem adalah sekumpulan elemen atau variabel yang saling berkaitan, berinteraksi, dan bergantung satu sama lain untuk mencapai suatu tujuan.

2.1.3 Karakteristik Sistem

Berikut merupakan karakteristik system menurut (Aida, 2018):

a. Komponen Sistem (*Components*)

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang artinya saling bekerja sama membentuk satu kesatuan.

b. Batas Sistem (*Boundary*)

Batas sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya.

c. Lingkungan Luar Sistem (*Environments*)

Lingkungan luar dari suatu sistem adalah apapun diluar batas dari sistem yang mempengaruhi operasi sistem.

d. Penghubung (*Interface*) Sistem

Penghubung sistem merupakan media penghubung antara satu subsistem dengan subsistem lainnya.

e. Masukan (*Input*) Sistem

Masukan sistem adalah energi yang dimasukkan ke dalam sistem. masukan dapat berupa masukan perawatan (*maintenance input*) dan masukan sinyal (*signal input*).

f. Keluaran (*Output*) Sistem

Keluaran sistem adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan. Keluaran dapat merupakan masukan untuk subsistem yang lain atau kepada supersistem.

g. Sasaran (*Objectives*) atau Tujuan (*Goal*)








Suatu sistem pasti mempunyai tujuan atau sasaran. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuannya. Perbedaan suatu sasaran (*objectives*) dan suatu tujuan (*goal*) adalah, tujuan biasanya dihubungkan dengan ruang lingkup yang lebih luas dan sasaran dalam ruang lingkup yang lebih sempit.

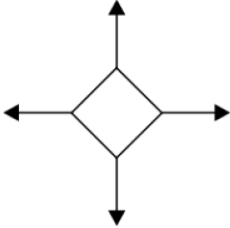


2.2 *Flowchart*

Flowchart atau diagram alir adalah representasi grafis dari langkah – langkah suatu algoritma. *Flowchart* menggunakan kotak dengan bentuk yang berbeda untuk menunjukkan jenis operasi yang berbeda. Kotak ini kemudian dihubungkan oleh garis dengan panah yang menunjukkan aliran atau arah untuk menemukan langkah selanjutnya. Garis penghubung ini dikenal sebagai aliran garis atau *flow lines* (Chaudhuri, 2020).

Berikut ini merupakan simbol yang digunakan dalam program *flowchart*:

Tabel 2.1 Tabel Daftar Simbol *Flowchart*

Simbol	Nama	Keterangan
	Terminal	Digunakan untuk menunjukkan awal dan akhir dari serangkaian proses
	Input/Output	Digunakan untuk menunjukkan operasi <i>input/output</i>
	<i>Process</i>	Digunakan untuk menunjukkan pemrosesan apa pun dilakukan oleh sistem komputer
	<i>Predefined processing</i>	Digunakan untuk menunjukkan proses apa pun yang tidak didefinisikan secara khusus dalam <i>flowchart</i>
	<i>Comment</i>	Digunakan untuk menulis pernyataan penjelasan apa pun diperlukan untuk mengklarifikasi sesuatu
	<i>Flow line</i>	Digunakan untuk menghubungkan simbol
	<i>Document Input/Output</i>	Digunakan untuk menyatakan input berasal dari dokumen dalam bentuk kertas atau output dicetak ke kertas

	<p><i>Decision</i></p>	<p>Digunakan untuk menunjukkan titik mana pun dalam proses dimana keputusan harus dibuat untuk menentukan tindakan lebih lanjut</p>
	<p><i>On-page connector</i></p>	<p>Digunakan untuk menghubungkan bagian – bagian dari <i>flowchart</i> berlanjut di halaman yang sama</p>
	<p><i>Off-page connector</i></p>	<p>Digunakan untuk menghubungkan bagian – bagian dari <i>flowchart</i> berlanjut di halaman yang terpisah</p>

Sumber: (Chaudhuri, 2020)

2.3 Teknik Pemodelan Sistem

Tujuan dari adanya teknik pemodelan sistem adalah agar memudahkan dalam memahami sistem yang dibutuhkan *user* dapat dijadikan sebagai perantara antara gambaran sistem dan model *design* (Maharani, 2018). Menurut Maharani (2018) terdapat 4 (empat) jenis pemodelan sistem, salah satu jenis pemodelan sistem adalah dengan berdasarkan pemodelan skenario (*Scenario Based Modelling*). Pemodelan ini dilakukan berdasarkan dari sudut pandang *user* serta pemodelan ini menggunakan UML.

2.3.1 UML (*Unified Modeling Language*)

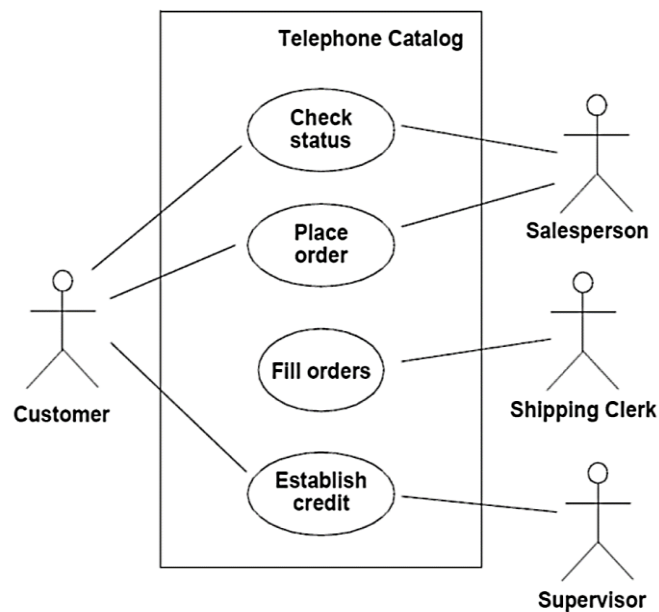
UML (*Unified Modeling Language*) merupakan bahasa pemodelan perangkat lunak yang telah distandarisasi sebagai media penulisan untuk cetak biru (*blueprints*) perangkat lunak. UML dapat digunakan untuk visualisasi, spesifikasi,

konstruksi dan beberapa dokumentasi sistem yang ada dalam perangkat lunak. UML digunakan untuk membantu *programmer* atau *developer* dalam membuat dan membangun *software* atau perangkat lunak (Sumiati et al., 2021).

Berdasarkan penjelasan (Mulyani, 2016) dalam bukunya yang berjudul *Metode Analisis dan Perancangan Sistem*, diagram yang didefinisikan oleh UML (*Unified Modeling Language*) diantaranya adalah sebagai berikut:

a. *Use Case Diagram*

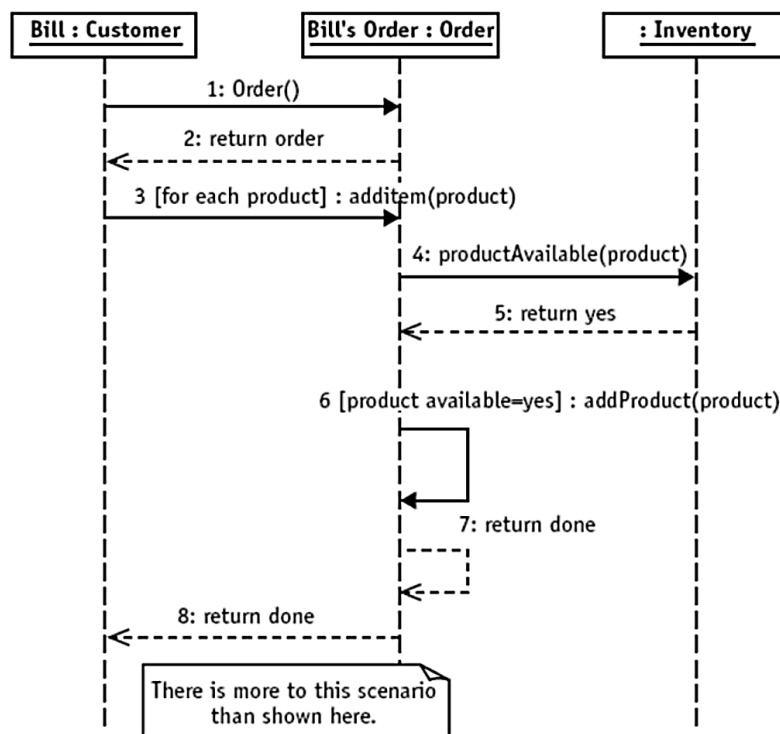
Use case diagram merupakan diagram yang menggambarkan dan mewakili aktor, *use cases*, dan *dependencies* dari sebuah proyek. Tujuan dari diagram ini adalah untuk menggambarkan konsep hubungan antara sistem dan dunia luar. Berikut merupakan contoh dari *use case diagram*:



Gambar 2.1 Contoh *Use Case Diagram*

b. *Sequence Diagram*

Sequence diagram adalah diagram yang menggambarkan interaksi antar objek. *Sequence diagram* biasanya menggambarkan perilaku skenario tunggal. Diagram ini menunjukkan sekumpulan objek sampel dan pesan yang melewatinya dalam sebuah *use case*. Contoh *sequence diagram* dapat dilihat pada gambar di bawah ini.



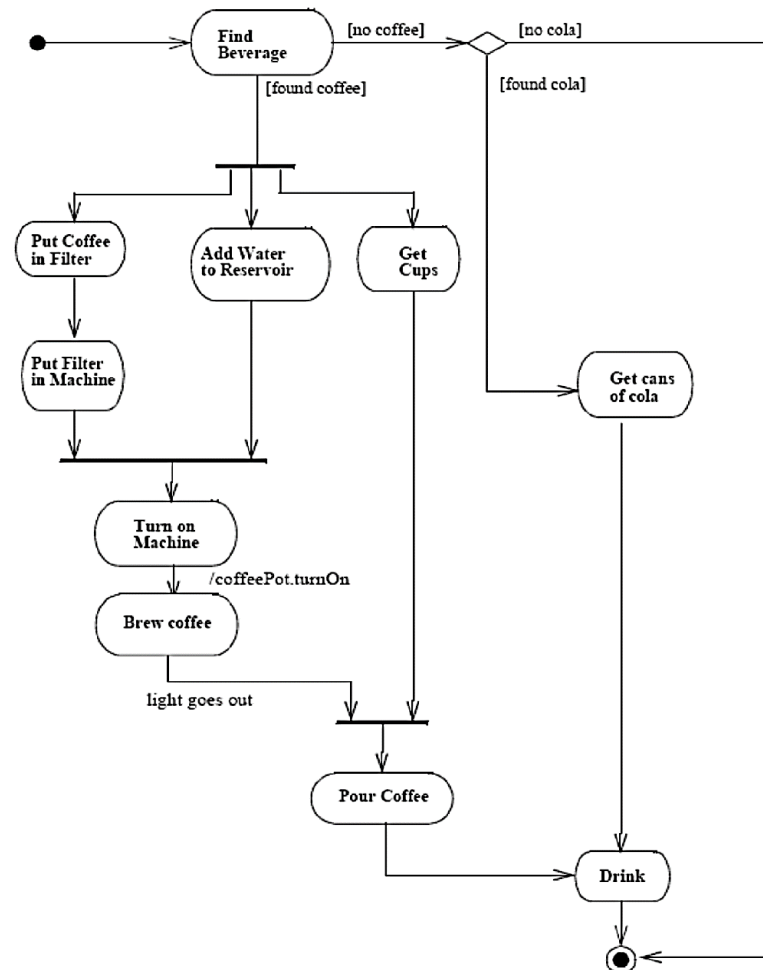
Gambar 2.2 Contoh *Sequence Diagram*

c. *Activity Diagram*

Activity diagram sangat mirip dengan *flowchart*. Perbedaannya adalah *activity diagram* dapat mencabangkan aktivitas. *Activity diagram* juga memungkinkan untuk mempartisi aktivitas antar aktor. *Activity diagram* adalah diagram UML yang digunakan untuk menggambarkan aliran aktivitas dalam suatu proses. *Activity diagram* memungkinkan orang yang menjalankan proses untuk

memilih urutan proses yang akan dilakukan. Dengan kata lain, diagram hanya menyebutkan seperangkat aturan dasar yang harus diikuti. Hal ini penting untuk pemodelan bisnis karena proses sering berjalan secara paralel. Dan juga berguna dalam algoritma paralel di mana urutan independen dapat dieksekusi secara paralel.

Berikut ini merupakan contoh dari *activity diagram*:

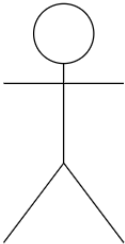
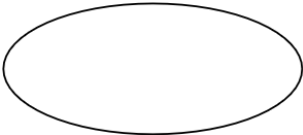
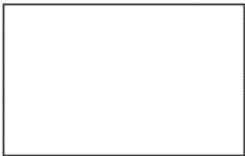

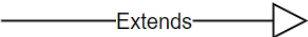


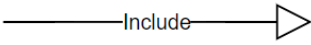
Gambar 2.3 Contoh *Activity Diagram*

2.4 Daftar Simbol Diagram

a. Daftar Simbol *Use Case Diagram*

Tabel 2.2 Tabel Daftar Simbol *Use Case Diagram*

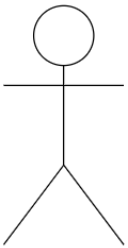
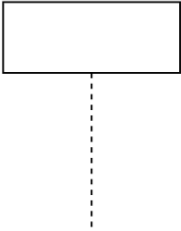

Simbol	Nama	Keterangan
	<i>Actor</i>	Menspesifikasikan himpunan peran ketika berinteraksi dengan sistem usulan.
	<i>Use Case</i>	Deskripsi dari urutan aksi – aksi yang ditampilkan sistem, dan mewakili sebagian besar sistem secara fungsional.
	Sistem	Menggambarkan ruang lingkup sistem.
	Asosiasi	Menghubungkan aktor dengan <i>use case</i> yang berinteraksi.
	<i>Extend</i>	Relasi yang menggambarkan bahwa sebuah <i>use case</i> (<i>sub use case</i>) bisa berdiri sendiri atau bisa berjalan tanpa menjalankan <i>main use case</i> terlebih dahulu.



	<i>Include</i>	Relasi yang menggambarkan bahwa sebuah <i>use case</i> (<i>sub use case</i>) harus menjalankan <i>use case</i> lain terlebih dahulu sebelum menjalankan fungsinya.
-----------------------------------------------------------------------------------	----------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------

Sumber: (Maharani, 2018)

b. Daftar Simbol *Sequence Diagram*

Tabel 2.3 Tabel Daftar Simbol *Sequence Diagram*



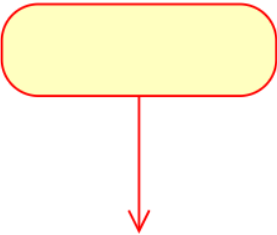
Simbol	Nama	Keterangan
	<i>Actor</i>	Menspesifikasikan himpunan peran ketika berinteraksi dengan sistem usulan.
	<i>Object lifeline</i>	Menyatakan hidup suatu <i>object</i> dalam basis waktu.
	<i>Activation</i>	Menyatakan <i>object</i> dalam keadaan aktif dan berinteraksi.

	<i>Message</i>	Pesan antar <i>object</i> , dan menggambarkan urutan kejadian.
	<i>Message return</i>	Menyatakan arah kembali antara urutan kejadian.

Sumber: (Maharani, 2018)

c. Daftar Simbol *Activity Diagram*

Tabel 2.4 Tabel Daftar Simbol *Activity Diagram*

Simbol	Nama	Keterangan
	<i>Start Point</i>	Merupakan awal penelusuran. Sebuah <i>activity diagram</i> selalu dimulai dengan <i>start point</i> .
	<i>End Point</i>	Merupakan akhir dari penelusuran. Sebuah <i>activity diagram</i> selalu diakhiri dengan <i>End Point</i> .
	<i>Activities</i>	<i>Activity</i> menggambarkan proses, diisi dengan kata kerja atau merupakan <i>state</i> dari sistem yang mencerminkan eksekusi dari suatu aksi.

<table border="1"> <tr> <td data-bbox="331 266 475 311"></td> <td data-bbox="475 266 627 311"></td> </tr> <tr> <td data-bbox="331 311 475 432"></td> <td data-bbox="475 311 627 432"></td> </tr> </table>					<i>Swimline style</i>	Sebuah cara untuk mengelompokkan <i>activity</i> berdasarkan <i>actor</i> . <i>Actor</i> bisa ditulis dengan nama <i>actor</i> .

Sumber: (Maharani, 2018)

2.5 Pengertian Aplikasi

Menurut (Maimunah et al., 2017) Aplikasi merupakan program yang dikembangkan untuk memenuhi kebutuhan pengguna dalam menjalankan pekerjaan tertentu. Sedangkan menurut (Fauzi Siregar et al., 2018) “aplikasi adalah alat terapan yang difungsikan secara khusus dan terpadu sesuai kemampuan yang dimilikinya dan merupakan suatu perangkat komputer yang siap pakai bagi *user*”.

2.6 Pengertian Web

Menurut Rohi Abdulloh dalam (Ibnu Sa’ad, 2020), *website* atau *web* adalah sekumpulan halaman yang terdiri dari beberapa halaman yang berisi informasi berupa teks, gambar, video, audio, dan data digital animasi lainnya yang disediakan melalui koneksi internet. Pada dasarnya *website* adalah halaman yang dapat diakses oleh browser dan berisi informasi yang dapat memberikan informasi yang berguna bagi pengguna.

2.7 Aplikasi Web

Pada awalnya aplikasi *web* dibangun dengan hanya menggunakan bahasa HTML (*Hypertext Markup Language*). Pada perkembangan berikutnya, sejumlah skrip dan objek dikembangkan untuk memperluas kemampuan HTML seperti PHP

dan ASP pada skrip. Aplikasi *web* dapat dibagi menjadi dua jenis yaitu aplikasi *web* statis dan dinamis. *Web* statis dibentuk dengan menggunakan HTML. Kekurangan aplikasi seperti ini terletak pada keharusan untuk memelihara program secara terus menerus untuk mengikuti setiap perkembangan yang terjadi. Kelemahan ini diatasi oleh model aplikasi *web* dinamis. Pada aplikasi *web* dinamis, perubahan informasi dalam halaman *web* dilakukan tanpa perubahan program tetapi melalui perubahan data. Sebagai implementasi, aplikasi *web* dapat dikoneksikan ke basis data sehingga perubahan informasi dapat dilakukan oleh operator dan tidak menjadi tanggung jawab dari pembangun *web* (Fachreza, 2013).

2.8 HTML

HTML adalah singkatan dari *Hyper Text Markup Language*, bahasa pemrograman utama. Saat membuat *website*, HTML terdiri dari *header* dan *body*, memiliki *tag* dan atribut, tetapi belum dapat disebut sebagai bahasa pemrograman. Hal tersebut dikarenakan logika yang diperlukan untuk suatu bahasa pemrograman yaitu tidak dimiliki oleh HTML. HTML hanya memberikan *output* saja, sehingga HTML diibaratkan sebagai pondasi atau struktur *web* dan yang menjadi bahasa pemrogramannya yaitu PHP dan Javascript (Marlina et al., 2021).

Dengan menuliskan kode – kode HTML, pada baris kode HTML tersebut mengandung perintah – perintah teks yang biasa disebut *tag* untuk menampilkan tulisan, warna, media (suara, gambar, video, serta animasi) serta *link* yang dapat menghubungkan ke situs lain ataupun *link* yang kita buat sendiri untuk topik – topik lainnya.

2.9 CSS

CSS atau singkatan *Cascading Style Sheet* adalah aturan untuk menyesuaikan tampilan sebuah website agar terlihat lebih terstruktur di *web*. CSS bukanlah bahasa pemrograman, CSS mirip dengan konfigurasi tampilan *tag* pada sebuah *website*. CSS dapat mengubah teks, warna, latar belakang dan posisi *tag* (Marlina et al., 2021).

2.10 JavaScript

Di dalam buku yang berjudul “Javascript dari A sampai Z “ karangan (Siahaan & Sianipar, 2018), bahwa JavaScript adalah “bahasa *script* yang dipakai untuk menciptakan halaman *web* yang dapat berinteraksi dengan pengguna dan dapat merespon *event* yang terjadi pada halaman”. Oleh (Siahaan & Sianipar, 2018) lebih lanjut menjelaskan JavaScript sangat erat kaitannya dengan *browser*, dan JavaScript sangat terintegrasi dengan HTML. Ketika suatu *website* sudah selesai memuat halaman, maka konten *website* tersebut akan dibaca dan diinterpretasi baris demi baris sampai baris yang terdapat *script* JavaScript oleh *browser*. Dan pada saat itu maka *script* JavaScript tersebut akan dieksekusi sampai *script* tersebut diakhiri dengan tak penutup dari JavaScript tersebut. Sedangkan menurut (Yudhanto & Prasetyo, 2018), JavaScript merupakan *script* yang digunakan untuk membuat *website* menjadi lebih interaktif kepada *user* yang menggunakan *website* tersebut”.

2.11 COVID-19

Pada Desember 2019, beberapa kasus penyakit pernapasan yang parah dilaporkan muncul di Wuhan, Provinsi Hubei, Cina. Hingga 25 Januari 2020, 1.975 kasus telah dilaporkan sejak pasien dirawat di rumah sakit pada 12 Desember 2019 (Wu et al., 2020). Penyakit pernapasan tersebut disebabkan oleh virus yang ditetapkan oleh *International Committee on Taxonomy of Viruses* (ICTV) sebagai *Severe Acute Respiratory Syndrome Coronavirus 2* (SARS-CoV-2) pada 11 Februari 2020 (Yang & Wang, 2020).

Wabah SARS-CoV-2 awalnya diduga dimulai melalui penularan dari hewan ke manusia maupun sebaliknya yang berkaitan dengan pasar makanan laut (*seafood market*) di Wuhan, China (Wu et al., 2020). Pada 30 Januari 2020, Komisi Darurat WHO mendeklarasikan darurat kesehatan masyarakat global berdasarkan peningkatan laporan China dan internasional (P. Velavan & Christian G., 2020). Penyakit coronavirus baru yang disebabkan oleh SARS-CoV-2 tersebut diberi nama *Coronavirus Disease 2019* (COVID19) (Yang & Wang, 2020).

Di Indonesia, penyebaran virus ini dimulai sejak tanggal 02 Maret 2020, diduga berawal dari salah satu warga negara Indonesia yang melakukan kontak langsung dengan warga negara asing (Nalini, 2021). Salah satu tindakan pemerintah Indonesia dalam mencegah penularan COVID-19 adalah Pemberlakuan Pembatasan Kegiatan Masyarakat (PPKM).

2.12 REST API

Sederhananya, REST (*REpresentational State Transfer*) adalah gaya arsitektur yang didefinisikan untuk membantu menciptakan dan mengatur sistem terdistribusi. Kata kunci dari definisi tersebut adalah *style*, karena aspek penting dari REST adalah bahwa itu merupakan gaya arsitektur bukan pedoman, bukan standar, atau apa pun yang akan menyiratkan bahwa ada seperangkat aturan keras untuk diikuti untuk akhirnya memiliki arsitektur *RESTful* (Doglio, 2018).

Arsitektur internal dasar REST API berisi item berikut (Doglio, 2018):

1. *A request handler*

Merupakan titik fokus yang menerima setiap permintaan dan memrosesnya sebelum melakukan hal lain.

2. *A middleware/pre-process chain*

Item – item ini membantu membentuk permintaan dan memberikan bantuan untuk kontrol otentikasi.

3. *A routes handler*

Setelah penanganan permintaan selesai, dan permintaan sendiri telah diperiksa dengan semua yang dibutuhkan, *routes handler* merupakan komponen yang mencari tahu siapa yang perlu mengurus permintaan atau *request* tersebut.

4. *The controller*

Ini bertanggung jawab atas semua permintaan yang dilakukan terkait dengan satu sumber daya tertentu.

5. *The Model*

Juga dikenal sebagai *resources*. Pada item ini kebanyakan akan fokus dengan logika yang terkait dengan *resources* di sini.

6. *The representation layer*

Lapisan ini mengurus pembuatan representasi yang terlihat oleh aplikasi klien.

7. *The response handler*

Response handler mengurus pengiriman representasi respon kembali kepada klien.

Menurut (Lawi et al., 2021) selama dua dekade terakhir, *Representational State Transfer* (REST) telah dianggap sebagai arsitektur standar untuk merancang dan menerapkan *Application Programming Interfaces* (API) sebagai aplikasi *back-end* atau layanan *web* sisi server. REST API biasanya mengumpulkan data dengan mengakses beberapa *endpoint*. Misalnya, membiarkan klien ingin mengambil data peneliti yang mencakup id (*identifier*) yang bersifat unik, judul penelitian, dan anggota proyek penelitian. Server mengirimkan representasi dari keadaan sumber daya yang diminta ke klien yang sebagian besar dalam format JSON (*JavaScript Object Notation*) atau XML (*eXtensible Markup Language*) (Lawi et al., 2021).

2.13 Prakiraan

Kamus Besar Bahasa Indonesia (KBBI) menyebutkan bahwa prakiraan merupakan melakukan peramalan suatu kejadian berdasarkan hasil perhitungan rasional.

Sedangkan menurut (Shodiq, 2019), prakiraan merupakan suatu alat bantu membuat keputusan tentang kejadian masa depan dengan mengambil data historis dan memprediksi masa depan menggunakan model matematika sehingga hasil yang diperoleh dapat membantu dalam berbagai bidang dalam menentukan kebijakan.

(Shodiq, 2019) juga menyebutkan bahwa, prakiraan membutuhkan data masa lalu serta data yang diperoleh juga memiliki tingkat fluktuasi yang tidak konsisten sehingga diperlukan pendekatan matematis yang dapat memberikan tingkat kesalahan yang terkecil agar memperoleh prakiraan yang akurat.

2.14 *Single Exponential Smoothing*

Metode *single exponential smoothing* pengembangan dari metode peramalan rata – rata bergerak yang menambahkan bobot eksponensial atau *multi-level* pada data terbaru, dimana data yang lebih baru memiliki bobot lebih yang besar (Gibran et al., 2021). Sedangkan menurut (L. W. S. Rahayu & B. G. Anandita, 2019) metode *single exponential smoothing* adalah teknik peramalan rata – rata bergerak dengan pembobotan dimana data diberi bobot oleh sebuah fungsi *exponential*.

Menurut (Shahid & Rahaman, 2020), metode *single exponential smoothing* (SES) dapat digunakan untuk memprediksi nilai masa depan dari data *time series* yang tidak *trending* dan bersifat musiman. Berikut merupakan rumus menghitung *single exponential smoothing* (Baharaeen & S. Masud, 1986):

$$F_{t+1} = F_t + \alpha(X_t - F_t) \dots\dots\dots (2.1)$$

Dimana:

F_t = Nilai peramalan untuk periode waktu tertentu

X_t = Nilai aktual untuk 1 periode waktu t

α = Konstanta *smoothing* alpha yang bervariasi antara 0 dan 1

2.15 Ukuran Ketepatan Metode Prakiraan

Nilai ukuran ketepatan metode prakiraan dilakukan untuk mengetahui ketepatan hasil prakiraan yang dilakukan. Ketepatan suatu metode prakiraan merupakan kesesuaian dari suatu metode yang pada akhirnya menunjukkan seberapa jauh model prakiraan tersebut mampu memprediksi data yang telah diketahui. Nilai *Mean Absolute Deviation* (MAD) dan *Mean Absolute Percentage Error* (MAPE) dari hasil perhitungan prakiraan menunjukkan perbedaan hasil prediksi dengan data aktual. MAD merupakan rata-rata kesalahan mutlak selama periode tertentu. Nilai absolut berguna untuk menghindari nilai penyimpangan positif dan penyimpangan negatif saling meniadakan (Ramadania, 2018). MAD dapat dihitung dengan rumus berikut (Ramadania, 2018):

$$MAD = \frac{1}{n} \sum_{t=1}^n |e_t| \dots\dots\dots (2.2)$$

Keterangan:

$|e_t|$ = Nilai *absolute error* (nilai aktual – nilai prakiraan) untuk 1 periode waktu t

n = Jumlah data

Sedangkan MAPE merupakan ukuran ketepatan relatif yang digunakan untuk mengetahui persentase penyimpangan hasil prakiraan. Semakin kecil nilai MAPE maka semakin kecil kesalahan hasil prediksi, sebaliknya semakin besar nilai MAPE maka semakin besar kesalahan hasil prediksi (Ramadania, 2018).

Berikut merupakan nilai MAPE yang digunakan untuk menganalisis kinerja dari proses memprakirakan:

Tabel 2.5 Tabel Tingkat Akurasi Nilai MAPE

Nilai MAPE	Tingkat Akurasi
$MAPE \leq 10\%$	Tinggi
$10\% < MAPE \leq 20\%$	Baik
$20\% < MAPE \leq 50\%$	Wajar
$MAPE > 50\%$	Rendah

Sumber: (Gustriansyah et al., 2017)

MAPE dapat dihitung dengan rumus berikut (Ramadania, 2018):

$$MAPE = \frac{1}{n} \sum_{t=1}^n \frac{|e_t|}{|X_t|} \times 100\% \dots\dots\dots (2.3)$$

Keterangan:

$|e_t|$ = Nilai *absolute error* (nilai aktual – nilai prakiraan) untuk 1 periode waktu t

$|X_t|$ = Nilai *absolute* dari nilai aktual untuk 1 periode waktu t

n = Jumlah data

2.16 Contoh Kasus dengan Menggunakan Metode *Single Exponential*

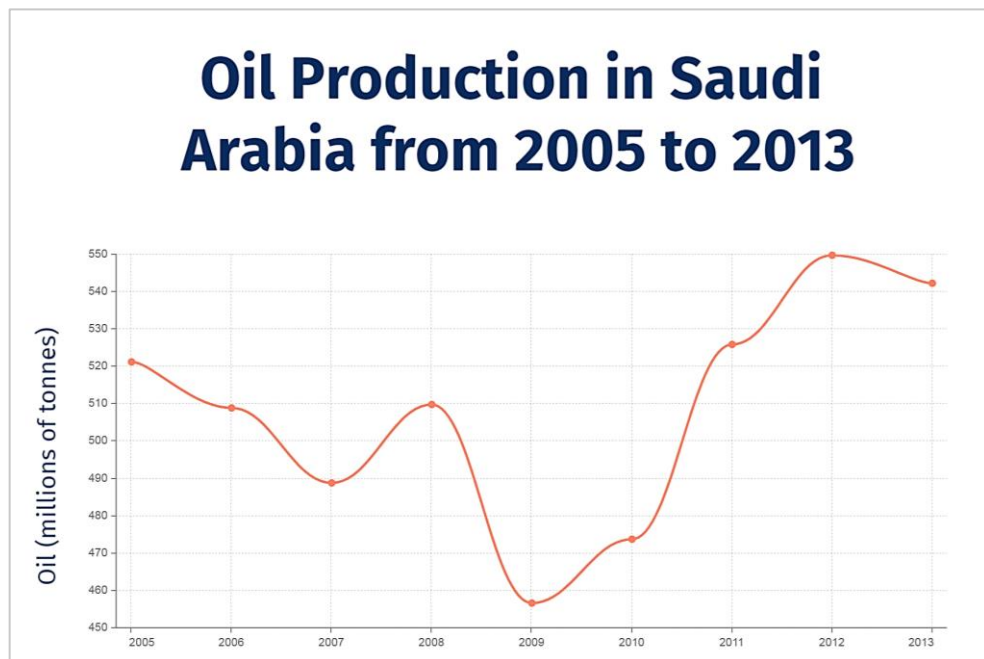
Smoothing

Sampel data total produksi minyak Arab Saudi dalam jutaan ton, berdasarkan buku yang berjudul “*Forecasting: Principles and Practice*” karya (Hyndman & Athanasopoulos, 2018) dapat dilihat pada tabel 2.6.

Tabel 2.6 Data Total Produksi Minyak Arab Saudi Tahun 2005 – 2013

Periode	Tahun	Jumlah
1	2005	521.28
2	2006	508.95
3	2007	488.89
4	2008	509.87
5	2009	456.72
6	2010	473.82
7	2011	525.95
8	2012	549.83
9	2013	542.34

Sumber: (Hyndman & Athanasopoulos, 2018)



Gambar 2.4 Grafik Data Produksi Minyak di Arab Saudi Tahun 2005 – 2013

Untuk mengetahui jumlah total produksi minyak Arab Saudi untuk tahun berikutnya yaitu tahun 2014, maka dilakukan perhitungan dengan menggunakan nilai parameter ($\alpha = 0.9$).

1. Perhitungan SES dengan $\alpha = 0.9$

a. Prakiraan Tahun 2005

Pada tahap Prakiraan tahun 2005, tidak ada data yang diketahui, oleh sebab itu hasil pada Prakiraan tahun 2005 dianggap sama dengan nilai jumlah aktual pada tahun 2005 ($F_1 = X_1$).

b. Prakiraan Tahun 2006

$$F_{1+1} = F_1 + \alpha(X_1 - F_1)$$

$$F_2 = 521.28 + 0.9(521.28 - 521.28)$$

$$F_2 = 521.28 + 0$$

$$F_2 = 521.28$$

c. Prakiraan Tahun 2007

$$F_{2+1} = F_2 + \alpha(X_2 - F_2)$$

$$F_3 = 521.28 + 0.9(508.95 - 521.28)$$

$$F_3 = 521.28 + (-11.097)$$

$$F_3 = 510.18$$

d. Prakiraan Tahun 2008

$$F_{3+1} = F_3 + \alpha(X_3 - F_3)$$

$$F_4 = 510.18 + 0.9(488.89 - 510.18)$$

$$F_4 = 510.18 + (-19.161)$$

$$F_4 = 491.01$$

e. Prakiraan Tahun 2009

$$F_{4+1} = F_4 + \alpha(X_4 - F_4)$$

$$F_5 = 491.01 + 0.9(509.87 - 491.01)$$

$$F_5 = 491.01 + 19.76$$

$$F_5 = 510.77$$

f. Prakiraan Tahun 2010

$$F_{5+1} = F_5 + \alpha(X_5 - F_5)$$

$$F_6 = 510.77 + 0.9(456.72 - 510.77)$$

$$F_6 = 510.77 + (-48.645)$$

$$F_6 = 462.12$$

g. Prakiraan Tahun 2011

$$F_{6+1} = F_6 + \alpha(X_6 - F_6)$$

$$F_7 = 462.12 + 0.9(473.82 - 462.12)$$

$$F_7 = 462.12 + 10.53$$

$$F_7 = 472.65$$

h. Prakiraan Tahun 2012

$$F_{7+1} = F_7 + \alpha(X_7 - F_7)$$

$$F_8 = 472.65 + 0.9(525.95 - 472.65)$$

$$F_8 = 472.65 + 47.97$$

$$F_8 = 520.62$$

i. Prakiraan Tahun 2013

$$F_{8+1} = F_8 + \alpha(X_8 - F_8)$$

$$F_9 = 520.62 + 0.9(549.83 - 520.62)$$

$$F_9 = 520.62 + 26.289$$

$$F_9 = 546.90$$

j. Prakiraan Tahun 2014

$$F_{9+1} = F_9 + \alpha(X_9 - F_9)$$

$$F_{10} = 546.90 + 0.9(542.34 - 546.90)$$

$$F_{10} = 546.90 + (-4.104)$$

$$F_{10} = 542.79$$

Tabel 2.7 Hasil Prakiraan SES dengan $\alpha = 0.9$

Periode	Tahun	Jumlah	Prakiraan ($\alpha = 0.9$)
1	2005	521.28	521.28
2	2006	508.95	521.28
3	2007	488.89	510.18
4	2008	509.87	491.01
5	2009	456.72	510.77
6	2010	473.82	462.12
7	2011	525.95	472.65
8	2012	549.83	520.62
9	2013	542.34	546.90
10	2014	?	542.79

Tabel 2.7 diatas merupakan tabel hasil prakiraan metode *single exponential smoothing* menggunakan nilai parameter ($\alpha = 0.9$).

2. Model nilai kesalahan

Dari persamaan (2.2) diperoleh nilai MAD yang ditunjukkan pada tabel 2.8 di bawah ini:

Tabel 2.8 Tabel Nilai MAD

Tahun	Jumlah (X_t)	Prakiraan ($\alpha = 0.9$)	e (Aktual - Prakiraan)	$ e_t $
2005	521.28	521.28	0	0
2006	508.95	521.28	-12.33	12.33
2007	488.89	510.18	-21.29	21.29
2008	509.87	491.01	18.86	18.86
2009	456.72	510.77	-54.05	54.05
2010	473.82	462.12	11.7	11.7
2011	525.95	472.65	53.30	53.30
2012	549.83	520.62	29.21	29.21
2013	542.34	546.90	-4.56	4.56
Total				205.3
MAD				22.81

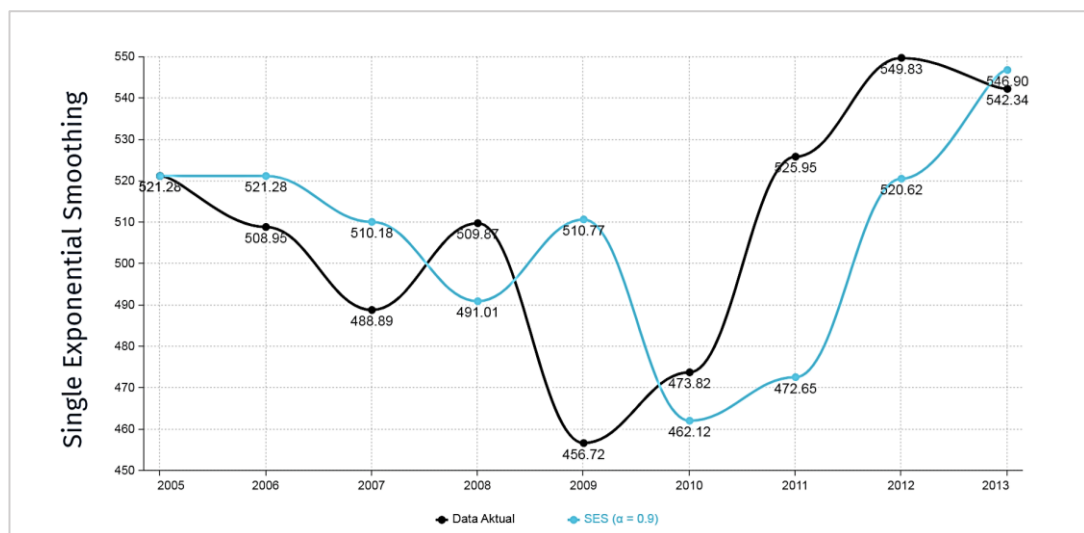
Tabel 2.8 Tabel Nilai MAD, merupakan tabel hasil perhitungan MAD yang dilakukan dengan mencari nilai *error* yaitu dengan cara menghitung selisih dari data aktual dengan data prakiraan lalu mengubah nilai *error* tersebut menjadi nilai *absolute*. Selanjutnya, menjumlahkan keseluruhan nilai *absolute error* dan dibagi dengan jumlah data yang ada.

Tabel 2.9 Tabel Nilai MAPE

Tahun	Jumlah (X_t)	Prakiraan ($\alpha = 0.9$)	e (Aktual - Prakiraan)	$\frac{ e_t }{ X_t } \times 100\%$
2005	521.28	521.28	0	0
2006	508.95	521.28	-12.33	2.42%
2007	488.89	510.18	-21.29	4.35%

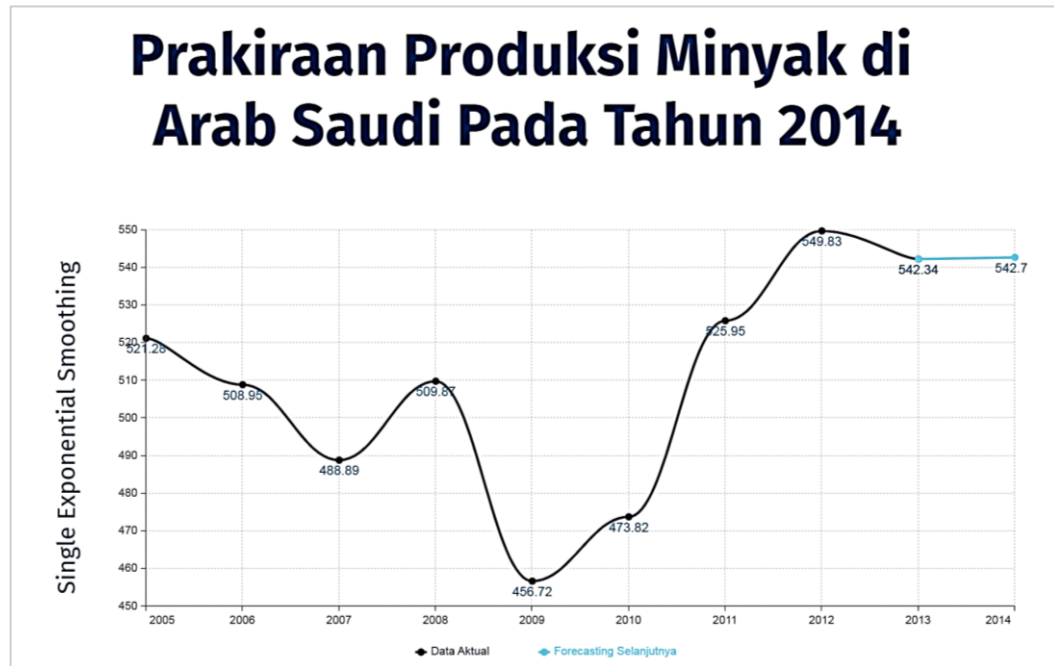
2008	509.87	491.01	18.86	3.69%
2009	456.72	510.77	-54.05	11.83%
2010	473.82	462.12	11.7	2.46%
2011	525.95	472.65	53.30	10.13%
2012	549.83	520.62	29.21	5.31%
2013	542.34	546.90	-4.56	0.84%
Total				41.03%
MAPE				4.55%

Tabel 2.9 Tabel Nilai MAPE, merupakan tabel hasil perhitungan MAPE yang dilakukan dengan mencari nilai *error* yaitu dengan cara menghitung selisih dari data aktual dengan data prakiraan lalu mengubah nilai *error* tersebut menjadi nilai *absolute*. Selanjutnya, nilai *absolute error* dibagi dengan nilai *absolute* dari data aktual, dikali dengan 100% dan dijumlahkan keseluruhan dari data yang ada. total penjumlahan yang telah dilakukan dibagi dengan jumlah data. Berdasarkan tabel 2.9, dapat dilihat bahwa nilai MAPE jika dinilai tingkat akurasi berdasarkan tabel 2.5 memiliki tingkat akurasi yang tinggi karena $4.55\% \leq 10\%$.



Gambar 2.5 Perbandingan Data Aktual dengan Data Prakiraan SES ($\alpha = 0.9$)

Gambar 2.5 merupakan grafik perbandingan data aktual yang ditandai dengan garis berwarna hitam dengan data prakiraan yang ditandai dengan garis berwarna biru menggunakan nilai $\alpha = 0.9$ pada metode *single exponential smoothing*.



Gambar 2.6 Prakiraan Tahun Berikutnya dengan SES ($\alpha = 0.9$)

Gambar 2.6 Prakiraan Tahun Berikutnya dengan SES ($\alpha = 0.9$), merupakan grafik data aktual dan data prakiraan pada tahun berikutnya yaitu tahun 2014. Data aktual ditandai dengan garis berwarna hitam sedangkan data prakiraan ditandai dengan garis berwarna biru menggunakan metode *single exponential smoothing* dengan nilai $\alpha = 0.9$, didapatkan nilai prakiraan pada tahun selanjutnya yaitu 542.7 juta ton.

Dari perhitungan yang telah dilakukan, dapat dilihat bahwa metode SES merupakan metode yang mudah dilakukan dan diterapkan. Selain itu, penulis menggunakan metode SES dengan melihat pola data COVID-19 yang tidak bersifat musiman dan tidak memiliki pola *trend*. Alasan ini didukung dengan pernyataan

oleh (Shahid & Rahaman, 2020) yang menyatakan bahwa, metode *single exponential smoothing* (SES) dapat digunakan untuk memprediksi nilai masa depan dari data *time series* yang tidak *trending* dan bersifat musiman.

Render dan Heizer (2001) dalam (Hudaningsih et al., 2020) menyatakan bahwa permasalahan yang sering dihadapi dalam metode ini adalah bagaimana menentukan nilai α yang tepat untuk meminimkan kesalahan prakiraan. Karena berlaku $0 < \alpha < 1$, maka dapat dilakukan panduan sebagai berikut:

- a. Apabila pola historis data sangat bergejolak atau tidak stabil maka pilih nilai α mendekati satu.
- b. Apabila pola historis data tidak bergejolak dan mendekati stabil maka pilih nilai α mendekati nilai nol.

Dari data COVID-19 di Indonesia dapat dilihat bahwa selain tidak bersifat musiman dan tidak memiliki pola *trend*, pola data historisnya sangat bergejolak atau tidak stabil. Oleh karena itu, penulis menggunakan nilai $\alpha = 0.9$ dalam melakukan penelitian ini.