



BAB II

TINJAUAN PUSTAKA

2.1 Teori Umum

2.1.1 Perangkat Lunak

2.1.1.1 Pengertian Perangkat Lunak

Sastrahidayat (2019:2), menyatakan bahwa "Istilah *software* diambil dari program komputer yang mempunyai makna khusus; yakni untuk data yang diformat, dan disimpan secara digital, termasuk program komputer, dokumentasinya, dan berbagai informasi yang bisa dibaca, dan ditulis oleh komputer". Sedangkan menurut Rianto (2021:5), mendefinisikan bahwa "*Software* (Perangkat Lunak) merupakan suatu data yang diprogram sedemikian rupa dan disimpan dalam bentuk digital yang tidak terlihat secara fisik tetapi tersimpan dalam media penyimpanan komputer". Pendapat lain dikemukakan oleh Sukamto dan Shalahuddin (2016:2), bahwa "Perangkat lunak (*software*) adalah program komputer yang terasosiasi dengan dokumentasi perangkat lunak seperti dokumentasi kebutuhan, model desain, dan cara penggunaan (*user manual*)."

Berdasarkan beberapa definisi perangkat lunak di atas, dapat disimpulkan bahwa perangkat lunak atau *software* adalah instruksi-instruksi atau data yang diformat secara digital, yang bisa dibaca dan ditulis oleh komputer sesuai kehendak pemakai.

2.1.1.2 Klasifikasi Perangkat Lunak

Klasifikasi perangkat lunak menurut Alda (2021:4-5), adalah sebagai berikut :

1. Sistem Operasi, merupakan perangkat lunak yang mengoperasikan komputer serta menyediakan antarmuka dengan perangkat lunak lain atau dengan pengguna. Contoh sistem operasi: *MS DOS*, *MS Windows* (dengan berbagai



- generasi), *Macintosh*, *OS/2*, *UNIX* (dengan berbagai versi), *LINUX* (dengan berbagai distribusi), *NetWare*, dan lain-lain.
2. Program Utilitas, merupakan program khusus yang berfungsi sebagai perangkat pemeliharaan komputer, seperti anti virus, partisi hardisk, manajemen hardisk, dan lain-lain. Contoh produk program utilitas: *Norton Utilities*, *PartitionMagic*, *McAfee*, dan lain-lain.
 3. Program Aplikasi, merupakan program yang dikembangkan untuk memenuhi kebutuhan yang spesifik. Contoh: Aplikasi Akuntansi, Aplikasi Perbankan, Aplikasi Manufaktur, dan lain-lain.
 4. Program Paket, merupakan program yang dikembangkan untuk kebutuhan umum, seperti:
 - a. Pengolah kata/*editor* naskah: *Wordstar*, *MS Word*, *Word Perfect*, *AmiPro*.
 - b. Pengolah angka/lembar kerja: *Lotus123*, *MS Excell*, *QuattroPro*.
 - c. Presentasi: *MS PowerPoint*.
 - d. Desain Grafis: *CorelDraw*, *PhotoShop*.
 5. Bahasa Pemrograman, merupakan perangkat lunak untuk pembuatan atau pengembangan perangkat lunak lain.

2.1.2 Komputer

2.1.2.1 Pengertian Komputer

Pengertian komputer menurut Rianto (2021:3), "Komputer merupakan suatu perangkat elektronika yang memiliki kemampuan untuk menerima dan mengolah data menjadi informasi, menjalankan program yang tersimpan dalam memori, serta dapat bekerja secara otomatis berdasarkan perangkat aturan tertentu". Sedangkan menurut Prawiro (dalam Harmayani, dkk, 2021:1), "Komputer adalah perangkat elektronik yang dapat digunakan untuk mengolah data sesuai dengan prosedur yang telah dirumuskan sebelumnya sehingga menghasilkan informasi bermanfaat bagi penggunanya."



Berdasarkan beberapa definisi di atas, dapat disimpulkan bahwa komputer adalah alat elektronik yang dapat mengolah data digital, dengan mengikuti serangkaian program yang digunakan untuk membantu pelaksanaan pekerjaan.

2.1.2.2 Elemen-Elemen dan Sistem Komputer

Menurut Sinaga (2020:9-10), Elemen-elemen dan sistem komputer adalah *software*, *hardware*, dan *brain ware* :

1. *Hardware* (Perangkat Keras) adalah peralatan di sistem komputer yang secara fisik terlihat dan dapat dijamah, seperti *monitor*, *keyboard*, *mouse*, dan lain-lain.
2. *Software* (Perangkat Lunak) adalah program yang berisi perintah-perintah untuk melakukan pengolahan data, *Software* terdiri dari :
 - a. Sistem operasi, *DOS*, *IjnuX*, *Windows*, *Mac*, dan lain-lain.
 - b. Bahasa pemrograman, *Visual Basic*, *C++*, *Pascal*, *Java*, *Visual C*, dan lain-lain.
 - c. Aplikasi, *MS Office*, *Antivirus*, *Winamp*, *Mozilla*, dan lain-lain.
3. *Brain ware* adalah manusia yang terlihat dalam mengoperasikan serta

2.1.3 Internet

2.1.3.1 Pengertian Internet

Pengertian internet menurut Ladiqi dan Wekke (2018:16-17), "Internet merupakan singkatan dari *International Networking* (jaringan internasional), secara teknis merupakan dua komputer atau lebih saling berhubungan membentuk jaringan komputer hingga meliputi jutaan komputer di dunia yang saling berinteraksi dan bertukar informasi." Pendapat lain dikemukakan oleh Sobri, dkk (2017:4), "Internet merupakan sebuah dunia maya jaringan komputer yang berbentuk dari miliaran komputer di dunia."

Berdasarkan beberapa definisi di atas, maka dapat disimpulkan bahwa internet adalah jaringan global yang menghubungkan seluruh jaringan di dunia agar saling berinteraksi dan melakukan pertukaran informasi.



2.1.3.2 Fungsi dan Tujuan Internet

Menurut Kumaran (2021), Internet memiliki beberapa fungsi dan tujuan sebagai berikut:

1. Memudahkan arus pertukaran informasi.
2. Menghubungkan manusia yang terpisah oleh jarak.

2.1.4 Pengertian Data

Pengertian Data menurut Pamungkas (2017:1), "Data merupakan nilai yang merepresentasikan deskripsi dari suatu objek atau kejadian". Adapun pengertian lainnya dikemukakan oleh Pane, dkk, (2020:7), "Data adalah kumpulan keterangan-keterangan atau deskripsi dasar dari suatu hal (objek atau kejadian) yang diperoleh dari hasil pengamatan (observasi), dan dapat diolah menjadi bentuk yang lebih kompleks, seperti: informasi, *database*, atau solusi untuk masalah tertentu."

Maka dapat disimpulkan bahwa data adalah kumpulan atas angka, karakter, dan fakta yang akan diklasifikasi, dimodifikasi, dan diolah oleh program agar dapat menjadi informasi yang dapat berguna bagi penggunanya.

2.1.5 Pengertian Basis Data (*Database*)

Rachmadi (2020:1), menyatakan jika "Basis Data adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan".

Menurut Pamungkas (2017:2), mendefinisikan "Basis data merupakan kumpulan data terhubung yang disimpan secara bersama-sama pada suatu media, yang diorganisasikan berdasarkan sebuah skema atau struktur tertentu, dan dengan *software* untuk melakukan manipulasi untuk kegunaan tertentu."

Berdasarkan beberapa definisi di atas, maka dapat disimpulkan bahwa basis data (*database*) adalah kumpulan data yang saling berhubungan, diorganisasi secara sistematis, menggunakan program komputer agar dapat dimanipulasi dan diperoleh dengan cepat dan mudah.



2.1.6 Metode Pengembangan Sistem

Penelitian Laporan Akhir ini menggunakan metode pengembangan sistem *Waterfall*. Menurut Pressman (dikutip Sanubari, dkk, 2020:41), menyatakan bahwa "Metode air terjun atau yang sering disebut metode *waterfall* sering dinamakan siklus hidup klasik (*classic life cycle*), dimana hal ini menggambarkan pendekatan yang sistematis dan juga berurutan pada pengembangan perangkat lunak, dimulai dengan spesifikasi kebutuhan pengguna lalu berlanjut melalui tahapan-tahapan perencanaan (*planning*), permodelan (*modeling*), konstruksi (*construction*), serta penyerahan sistem ke pengguna (*deployment*), yang diakhiri dengan dukungan pada perangkat lunak lengkap yang dihasilkan."

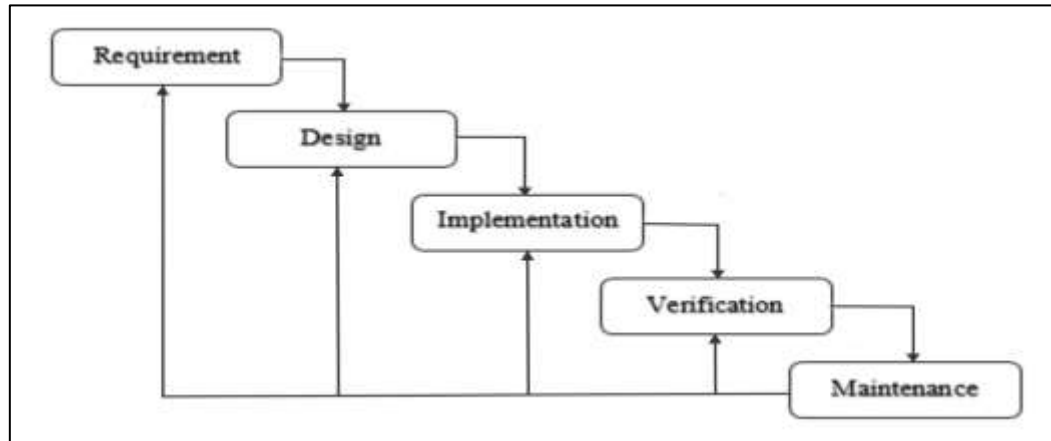
Menurut Sukamto dan Shalahuddin (2016:28), menyatakan bahwa "Model SDLC air terjun (*waterfallII*) sering juga disebut model sekuensial linier (*sequential linear*) atau alur hidup klasik (*classic life cycle*). Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut mulai dari analisis, desain, pengkodean, pengujian, dan tahap pendukung (*support*)."

Berdasarkan beberapa definisi di atas, maka dapat disimpulkan bahwa metode *waterfall* adalah suatu proses atau metode pengembangan sistem secara berurutan yang mengalir ke bawah (seperti air terjun) melalui tahapan-tahapan *Requirement Analysis, Design System, Implementation, Verification & Testing*, dan *Maintenance*.

Pengembangan sistem dengan menggunakan metode *waterfall* memiliki beberapa tahapan yang berurut yaitu: *Requirement* (analisis kebutuhan), *Design System* (desain sistem), *Coding* (pengkodean) & *Testing* (pengujian), Penerapan Program, Pemeliharaan.



Tahapan metode *waterfall* dapat dilihat pada gambar di bawah ini:



Sumber : Sanubari, dkk (2020:41).

Gambar 2.1 Tahapan-Tahapan Metode *Waterfall*

Tahapan-tahapan dari metode *waterfall* menurut Sanubari, dkk (2020:41-42), sebagai berikut:

a. *Requirement Analisis*

Tahap ini pengembang sistem diperlukan komunikasi yang bertujuan untuk memahami perangkat lunak yang diharapkan oleh pengguna dan batasan perangkat lunak tersebut. Informasi ini biasanya dapat diperoleh melalui wawancara, diskusi atau survei langsung. Informasi dianalisis untuk mendapatkan data yang dibutuhkan oleh pengguna.

b. *System Design*

Spesifikasi kebutuhan tahap sebelumnya akan dipelajari dalam fase ini dan desain sistem dipersiapkan. Desain sistem membantu dalam menentukan perangkat keras (*hardware*) dan sistem persyaratan dan juga membantu dalam mendefinisikan arsitektur sistem secara keseluruhan.

c. *Implementation*

Tahap ini dimana 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*.



d. *Integration & Testing*

Seluruh unit yang dikembangkan dalam tahap implementasi diintegrasikan ke dalam sistem setelah pengujian yang dilakukan masing-masing unit. Setelah integrasi seluruh sistem diuji untuk mengecek setiap kegagalan maupun kesalahan.

e. *Operation & Maintenance*

Operation & Maintenance merupakan tahap akhir dalam model waterfall. Perangkat lunak yang sudah jadi dijalankan serta dilakukan pemeliharaan. Pemeliharaan termasuk dalam memperbaiki kesalahan yang tidak ditemukan pada langkah sebelumnya. Perbaiki implementasi unit sistem dan peningkatan jasa sistem sebagai kebutuhan baru.

2.2 Teori Khusus

2.2.1 *Unified Modeling Language* (UML)

2.2.1.1 Pengertian *Unified Modeling Language* (UML)



Gambar 2.2 Logo UML

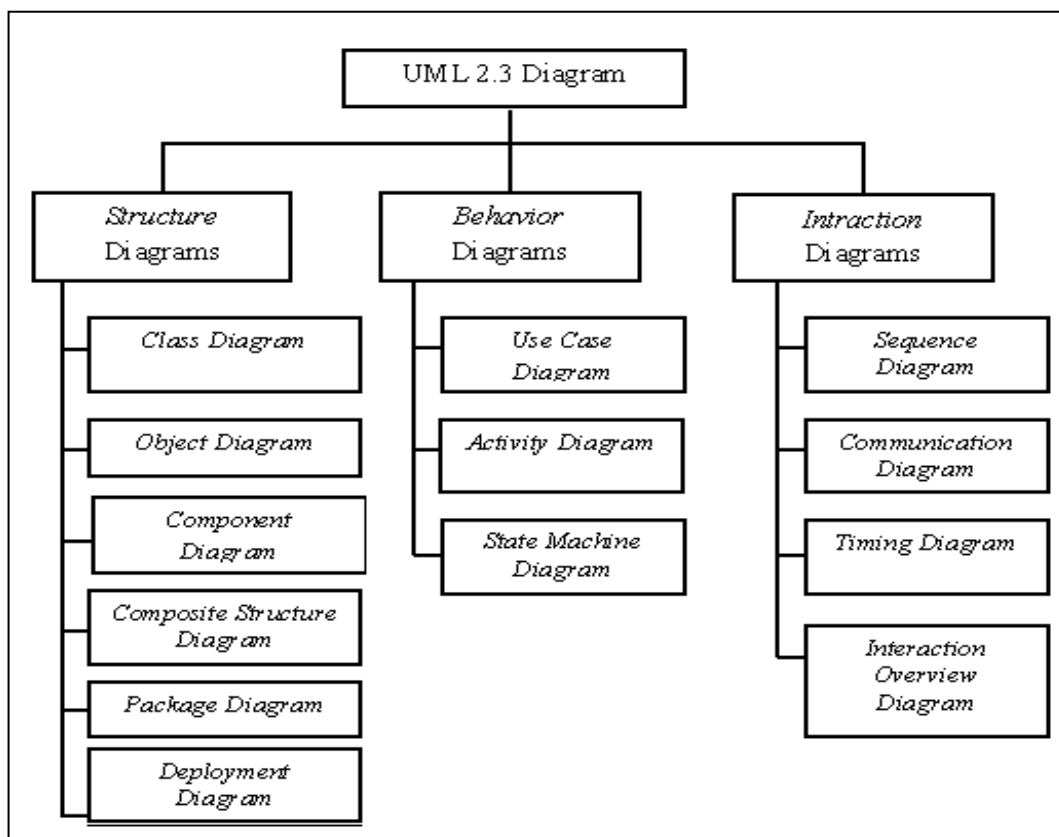
Menurut Haqi (2019:29), "UML merupakan singkatan dari *“Unified Modelling Language”* yaitu suatu metode permodelan secara visual untuk sarana perancangan sistem berorientasi objek, atau definisi UML yaitu sebagai suatu bahasa yang sudah menjadi standar pada visualisasi, perancangan dan juga pendokumentasian sistem *software*." Sedangkan menurut Herlinah dan Musliadi (2019:50), "UML merupakan salah satu alat bantu pengembangan sistem berorientasi objek."



Berdasarkan beberapa definisi di atas, dapat disimpulkan bahwa UML (*Unified Modelling Language*) adalah metode atau alat bantu dalam melakukan permodelan dan pengembangan sistem yang berorientasi objek.

2.2.1.2 Kategori *Unified Modeling Language* (UML)

Unified Modeling Language (UML) terdiri dari 13 macam diagram dan dikelompokkan menjadi 3 kategori sebagai berikut:



Sumber : Sukamto dan Shalahuddin (2016:140).

Gambar 2.3 Kategori dan Macam-Macam Diagram UML

Berikut penjelasan singkat dari pembagian kategori pada diagram UML menurut Sukamto dan Shalahuddin (2016:141):

- 1) *Structure diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan.



-
- 2) *Behavior diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sistem.
 - 3) *Interaction diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem maupun interaksi antar subsistem pada suatu sistem.

2.2.2 Jenis-Jenis Diagram *Unified Modeling Language* (UML)

2.2.2.1 Pengertian *Use Case Diagram*

Menurut Sukamto dan Shalahuddin (2016:155), "*Use case* atau diagram *use case* merupakan 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". Sedangkan menurut Setyawati, dkk (2021:13), "*Use case diagram* merupakan konsep informasi dengan menggunakan fungsi apa saja yang ada di dalam sistem informasi & siapa saja *actor* yang terlibat."

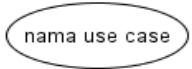
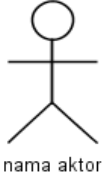

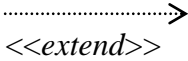
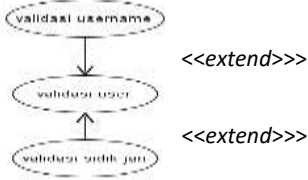
Berdasarkan beberapa definisi di atas, dapat disimpulkan bahwa *Use Case Diagram* adalah suatu permodelan atau konsep informasi untuk mengetahui kelakuan sistem informasi dengan mendeskripsikan interaksi antara aktor dan sistem informasi.

Menurut Sukamto dan Shalahuddin (2016:155), ada dua hal utama pada *use case* yaitu pendefinisian apa yang disebut aktor dan *use case*.

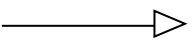
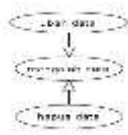
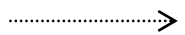
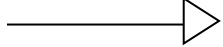
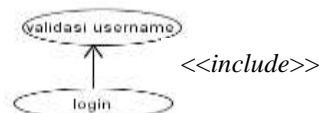
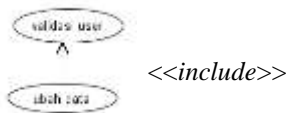
- a. **Aktor** merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.
- b. *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.

Simbol-simbol pada *Use Case Diagram* adalah sebagai berikut:

Tabel 2.1 Simbol-Simbol *Use Case Diagram*

No	Simbol	Deskripsi
1.	<p><i>Use case</i></p> 	<p>fungsi yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal-awal frase nama <i>use case</i>.</p>
2.	<p>aktor / <i>actor</i></p> 	<p>orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang dibuat di luar sistem informasi yang dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama <i>actor</i>.</p>
3.	<p>asosiasi / <i>association</i></p> 	<p>komunikasi antar aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan <i>actor</i>.</p>
4.	<p>ekstensi / <i>extend</i></p> 	<p>relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang di tambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu; mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek; biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan, misalnya</p>  <p>arah panah mengarah pada <i>use case</i> yang ditambahkan; biasanya <i>use case</i> yang menjadi <i>extend</i>-nya merupakan jenis yang sama dengan <i>use case</i> yang menjadi induknya</p>

Lanjutan Tabel 2.1 Simbol-Simbol *Use Case Diagram*

No	Simbol	Deskripsi
5.	Generalisasi / <i>generalization</i> 	hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi lebih umum dari lainnya,  misalnya: arah panah mengarah pada <i>use case</i> yang menjadi generalisasinya (umum).
6.	menggunakan / <i>include / uses</i>  <<include>>  <<uses>>	relasi tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya. ada dua sudut pandang yang cukup besar mengenai include di <i>use case</i> : a. <i>Include</i> berarti <i>use case</i> yang ditambahkan akan selalu di panggil saat <i>use case</i> tambahan dijalankan, misalnya pada kasus berikut:  b. <i>Include</i> berarti <i>use case</i> yang tambahan akan selalu melakukan pengecekan apakah <i>use case</i> yang di tambahkan telah dijalankan sebelum <i>use case</i> tambahan dijalankan, misal pada kasus berikut:  kedua interpretasi di atas dapat dianut salah satu atau keduanya tergantung pada pertimbangan dan interpretasi yang dibutuhkan.

Sumber : Sukamto dan Shalahuddin (2016:156-158).



2.2.2.2 Pengertian *Class Diagram*

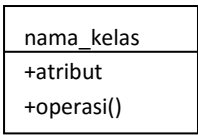



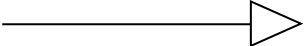
Menurut Sukamto dan Shalahuddin (2018:141-142), "*Class Diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem."

Menurut Haqi (2019:32), "*Class diagram* yaitu salah satu jenis diagram pada UML yang digunakan untuk menampilkan kelas-kelas maupun paket-paket yang ada pada suatu sistem yang nantinya akan digunakan. Jadi diagram ini dapat memberikan sebuah gambaran mengenai sistem maupun relasi yang terdapat pada sistem tersebut."


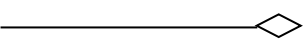
Berdasarkan beberapa definisi di atas, dapat disimpulkan bahwa *Class Diagram* adalah permodelan yang menggambarkan struktur sistem yang digunakan untuk menampilkan kelas atau paket mengenai sistem dan juga relasi yang ada pada sistem.

Simbol-simbol pada *Class Diagram* adalah sebagai berikut:

Tabel 2.2 Simbol-Simbol *Class Diagram*

No	Simbol	Deskripsi
1.	kelas 	Kelas pada struktur sistem.
2.	antarmuka / <i>interface</i> 	Sama dengan konsep interface dalam pemrograman berorientasi objek.
3.	asosiasi / <i>association</i> 	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai <i>multiplicity</i> .
4.	asosiasi berarah / <i>directed association</i> 	Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
5.	generalisasi 	Relasi antarkelas dengan makna generalisasi – spesialisasi (umum khusus).

Lanjutan Tabel 2.2 Simbol-Simbol *Class Diagram*

No	Simbol	Deskripsi
6.	kebergantungan / <i>dependency</i> 	Relasi antarkelas dengan makna kebergantungan antar kelas.
7.	agregasi / <i>aggregation</i> 	Relasi antarkelas dengan makna semua-bagian (<i>whole-part</i>).

Sumber : Sukamto dan Shalahuddin (2016:146-147).

2.2.2.3 Pengertian *Activity Diagram*


Menurut Sukamto dan Shalahuddin (2016:161), "Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem."

Menurut Haqi (2019:31), "*Activity diagram* atau diagram aktivitas yaitu salah satu jenis diagram pada UML yang dapat memodelkan proses-proses apa saja yang terjadi pada sistem."

Berdasarkan beberapa definisi di atas, dapat disimpulkan bahwa *Activity Diagram* merupakan permodelan yang menggambarkan aliran kerja mengenai proses-proses apa saja yang terjadi di dalam sebuah sistem yang ada pada perangkat lunak.

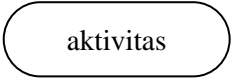




Simbol-simbol pada *Activity Diagram* adalah sebagai berikut:

Tabel 2.3 Simbol-simbol *Activity Diagram*

No	Simbol	Deskripsi
1.	Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.



Lanjutan Tabel 2.3 Simbol-simbol *Activity Diagram*

No	Simbol	Deskripsi
2.	Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3.	Percabangan / <i>decision</i> 	Asosiasi percabangan di mana jika ada pilihan aktivitas lebih dari satu.
4.	Penggabungan / <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
5.	Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
6.	Swimlane  atau	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

Sumber : Sukamto dan Shalahuddin (2016:162-163).

2.2.2.4 Pengertian *Sequence Diagram*

Menurut Sukamto dan Shalahuddin (2016:165), "Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansikan menjadi objek itu. Membuat diagram sekuen juga dibutuhkan untuk melihat *scenario* yang ada pada *use case*."


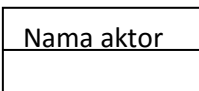
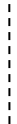
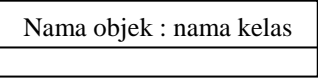


Menurut Haqi (2019:31), "*Sequence* diagram yaitu salah satu jenis diagram pada UML yang menjelaskan interaksi objek yang berdasarkan urutan waktu, *sequence* diagram juga dapat menggambarkan urutan atau tahapan yang harus dilakukan untuk dapat menghasilkan sesuatu seperti pada *use case* diagram."

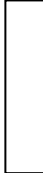
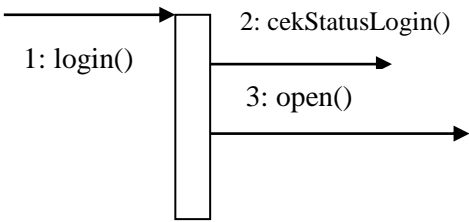


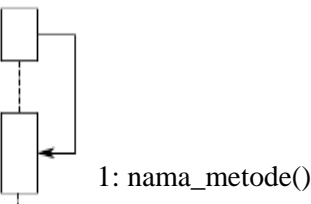
Berdasarkan beberapa definisi di atas, dapat disimpulkan bahwa *Sequence Diagram* merupakan permodelan yang mendeskripsikan interaksi objek dengan waktu hidup yang dikirimkan dan diterima objek, dan dapat menggambarkan tahapan yang harus dilakukan oleh sistem untuk memenuhi kebutuhan yang ada pada *use case*.

Simbol-simbol pada *Sequence Diagram* adalah sebagai berikut:

Tabel 2.4 Simbol-Simbol *Sequence Diagram*

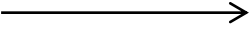
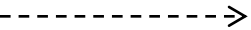
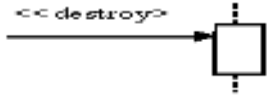
No	Simbol	Deskripsi
1.	<p><i>Actor</i></p>  <p>nama aktor</p> <p>Atau</p>  <p>Nama aktor</p> <p>tanpa waktu aktif</p>	orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama <i>actor</i> .
2.	<p>Garis hidup / <i>lifeline</i></p> 	menyatakan kehidupan suatu objek.
3.	<p>Objek</p>  <p>Nama objek : nama kelas</p>	menyatakan objek yang berinteraksi pesan.

Lanjutan Tabel 2.4 Simbol-Simbol *Sequence Diagram*

No	Simbol	Deskripsi
4.	<p>Waktu aktif</p> 	<p>menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya, misalnya</p>  <p>maka cek Status Login () dan open() dilakukan di dalam metode login() aktor tidak memiliki waktu aktif.</p>
5.	<p>Pesan tipe <i>create</i></p> <p><<create>></p> 	<p>menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.</p>
6.	<p>Pesan tipe <i>call</i></p> <p>1: nama_metode()</p> 	<p>menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri,</p>  <p>arah panah mengarah pada objek yang memiliki operasi/metode, karena ini memanggil operasi/metode maka operasi/metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi.</p>



Lanjutan Tabel 2.4 Simbol-Simbol *Sequence Diagram*

No	Simbol	Deskripsi
7.	Pesan tipe <i>send</i> 1: masukan 	menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.
8.	Pesan tipe <i>return</i> 1: keluaran 	menyatakan suatu objek telah menjalankan suatu operasi atau metode menghasilkan kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.
9.	Pesan tipe <i>destroy</i> 	menyatakan suatu objek mengakhiri hidup objek lain, arah panah mengarah pada objek yang diakhiri, jika ada <i>create</i> maka ada <i>destroy</i> .

Sumber : Sukamto dan Shalahuddin (2016:165-167).

2.3 Teori Judul

2.3.1 Pengertian Perancangan

Menurut Nur dan Suyuti (2018:5), mendefinisikan bahwa "Perancangan adalah suatu proses yang bertujuan untuk menganalisis, menilai, memperbaiki, dan menyusun suatu sistem, baik sistem fisik maupun non fisik yang optimum untuk waktu yang akan datang dengan memanfaatkan informasi yang ada."

Menurut Prasetyowibowo (dikutip Prasnowo, dkk, 2020:22), mendefinisikan "Perancangan (*design*) merupakan suatu kegiatan atau rekayasa rancang bangun yang dimulai dari ide-ide inovasi desain, atau kemampuan untuk menghasilkan karya dan cipta yang benar-benar dapat menjabarkan permintaan pasar karena adanya penelitian dan pengembangan teknologi."

Berdasarkan beberapa definisi di atas, maka dapat disimpulkan bahwa perancangan adalah rekayasa rancang bangun untuk memberikan gambaran secara umum dan fungsionalitas dari aplikasi yang akan dibuat kepada pemakai.



2.3.2 Pengertian Aplikasi

Menurut Pane, dkk (2020:53), istilah aplikasi berasal dari bahasa Inggris yaitu "*Application*" yang diartikan sebagai penerapan atau penggunaan. Sedangkan pengertian Aplikasi menurut Fathoroni, dkk (2020:1), mendefinisikan bahwa "Aplikasi adalah sebuah program siap pakai yang bisa digunakan untuk menjalankan sejumlah perintah dari pengguna aplikasi itu sendiri."

Berdasarkan beberapa definisi di atas, maka dapat disimpulkan bahwa aplikasi merupakan perangkat lunak (*software*) yang dibuat untuk menolong beberapa aktivitas manusia dalam melaksanakan tugas tertentu.

2.3.3 Pengertian *E-Booking*

Menurut Putri (2018:6), menyatakan *E-Booking* (*Electronic Booking*) yaitu pemesanan secara *online* menggunakan internet dengan mengakses halaman *website* yang tersedia. Namun, menurut Aprilia (2020:5), mendefinisikan bahwa "Pemesanan adalah aktivitas yang dilakukan konsumen sebelum proses membeli."

Pendapat lain dikemukakan oleh Susanti (2017:2) "Pemesanan adalah suatu perjanjian yang dilakukan oleh 2 (dua) pihak atau lebih yaitu pemberi dan pemakai jasa atau barang untuk memenuhi kebutuhannya dalam mengusahakan barang atau jasa tersebut sehingga dapat digunakan."

Berdasarkan beberapa definisi di atas, maka dapat disimpulkan bahwa *e-booking* (pemesanan secara elektronik) adalah proses melakukan pemesanan oleh konsumen atau pelanggan yang dilakukan secara *online* menggunakan jaringan internet dengan melakukan pengaksesan terhadap halaman *website*.

2.3.4 Pengertian Pengelolaan

Pengertian Pengelolaan menurut Indrawan (2020:123), "Pengelolaan adalah seni dalam menyelesaikan pekerjaan melalui orang lain." Sedangkan menurut Terry (dikutip Chotimah dan Mariyani, 2021:129), "Pengelolaan adalah suatu proses atau kerangka kerja yang melibatkan bimbingan atau pengarahan suatu kelompok orang-orang ke arah tujuan organisasi atau maksud yang nyata."



Berdasarkan beberapa definisi di atas, dapat disimpulkan pengelolaan adalah suatu proses yang dilakukan oleh manusia untuk melakukan pengawasan, pengorganisasian, dan pelaksanaan kebijakan untuk mencapai tujuan.

2.3.5 Pengertian Data

Menurut Yusuf dan Daris (2018:3), mendefinisikan bahwa "Data didefinisikan sebagai deskripsi atau keterangan sebuah objek yang belum memiliki makna secara utuh, dapat berupa angka (*numeric*), karakter (*text*), gambar (*image*), suara (*sound*) ataupun lambang (*symbol*)."

Menurut Pane, dkk, (2020:17), "Data adalah kumpulan keterangan-keterangan atau deskripsi dasar dari suatu hal (objek atau kejadian) yang diperoleh dari hasil pengamatan (*observasi*), dan dapat diolah menjadi bentuk yang lebih kompleks, seperti: informasi, *database*, atau solusi untuk masalah tertentu."

Berdasarkan beberapa definisi di atas, maka dapat disimpulkan bahwa data adalah sekumpulan keterangan atau deskripsi dasar sebuah objek atau kejadian yang belum memiliki makna secara utuh, dapat berupa angka (*numeric*), karakter (*text*), gambar (*image*), suara (*sound*) ataupun lambang (*symbol*) yang dapat diolah menjadi bentuk yang lebih kompleks agar menghasilkan suatu informasi.

2.3.6 Pengertian Aset (*Asset*)

Menurut Wahyuni dan Khoirudin (2020:132), "Aset adalah sesuatu atau barang yang dimiliki oleh seseorang, badan usaha, atau instansi, yang memiliki nilai ekonomi, nilai komersial, dan nilai tukar, di mana aset tersebut dapat berupa barang tidak bergerak (tanah dan bangunan) atau barang bergerak (modal, simpanan, hutang) yang dapat dianggap sebagai harta kekayaan."

Menurut Zamzami dan Nusa (2016:16), menyatakan bahwa "Aset merupakan sesuatu yang memiliki nilai ekonomi yang dapat membantu perusahaan menyediakann barang dan jasa untuk pelanggan, baik secara langsung maupun tidak langsung."



Pendapat lain dikemukakan oleh Gie (2021:39), "Aset adalah sumber daya yang dimiliki atau dikendalikan oleh individu, perusahaan, atau pemerintah dengan harapan akan menghasilkan arus kas masa depan."

Berdasarkan beberapa definisi di atas, maka dapat disimpulkan bahwa aset adalah sesuatu atau barang harta kekayaan yang memiliki nilai jual / ekonomi yang dimiliki oleh seseorang, badan usaha atau instansi.

2.3.7 Pengertian Rumah

Pengertian Rumah dalam situs *website* Wikipedia (2021), "Rumah adalah salah satu bangunan yang dijadikan tempat tinggal selama jangka waktu tertentu". Sedangkan menurut Abdullah (2018:1), menyatakan bahwa "Rumah merupakan kebutuhan dasar manusia yang selain berfungsi sebagai tempat berteduh dan melakukan kegiatan sehari-hari dalam keluarga, juga berperan besar dalam pembentukan karakter keluarga."

Berdasarkan beberapa definisi di atas, maka dapat disimpulkan bahwa pengertian rumah adalah bangunan yang digunakan manusia sebagai tempat tinggal / tempat berteduh dan sebagai tempat bagi penghuninya untuk melakukan kegiatan sehari-hari.

2.3.8 Pengertian Kavling

Menurut Zayyadi dan Halim (2020:104), mendefinisikan bahwa "Kavling merupakan bagian tanah yang sudah dipetak-petak dengan ukuran tertentu yang akan dijadikan bangunan atau tempat tinggal". Pengertian lain dari Kavling dalam situs *website* Paramount Land (2021), "Kavling adalah tanah yang memiliki ukuran pasti, keberadaannya bisa diluar ataupun di dalam sebuah perumahan".

Berdasarkan beberapa definisi di atas, maka dapat disimpulkan bahwa kavling adalah tanah yang sudah dipetak-petak dengan ukuran yang pasti digunakan untuk mendirikan bangunan ataupun tempat tinggal.



2.3.9 Pengertian Website

Pengertian *Website* menurut Sa'ad (2020:3), "*Website* merupakan halaman-halaman yang berisi informasi yang dapat diakses oleh *browser* dan mampu memberikan informasi yang berguna bagi para pemkainya."

Pengertian *Website* lainnya dikemukakan oleh Marisa (2017:1), "*Website* dapat diartikan sebagai kumpulan halaman-halaman yang digunakan untuk mempublikasikan informasi berupa teks, gambar, dan program multimedia lainnya berupa animasi (gambar gerak, tulisan gerak), suara, dan atau gabungan dari semuanya itu baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait antara satu halaman dengan halaman yang lain yang sering disebut sebagai *hyperlink*."

Berdasarkan beberapa definisi di atas, maka dapat disimpulkan bahwa *website* adalah halaman yang berisi informasi berupa teks, gambar, suara, dan video yang dapat disimpan pada komputer lokal maupun *server web*, yang terhubung melalui link *hypertext* dan dapat dijadikan atau ditampilkan pada browser.

2.3.10 Pengertian Aplikasi *E-Booking* dan Pengelolaan Data Aset Rumah dan Kavling pada PT Mahaputra Karya Konstruksindo Palembang Berbasis Website

Aplikasi *E-Booking* dan Pengelolaan Data *Asset* Rumah dan Kavling pada PT Mahaputra Karya Konstruksindo Palembang Berbasis *Website* adalah aplikasi untuk memudahkan konsumen dalam melakukan proses *booking online* pembelian rumah dan kavling dan memudahkan perusahaan dalam melakukan pengelolaan terhadap data aset yang berupa rumah dan kavling.

Manfaat yang akan diterima oleh konsumen dan pihak perusahaan dengan adanya aplikasi ini adalah sebagai berikut:

- a. Dapat memudahkan konsumen dalam melakukan proses *booking* secara *online* dan dapat mengetahui mengenai data dari *asset* rumah dan kavling yang masih tersedia atau dijual oleh pihak perusahaan.



- b. Dapat memudahkan pihak perusahaan dalam melakukan pengelolaan data *asset*, data penjualan, dan data laporan secara lebih efektif dan efisien. Selain itu, data-data tersebut telah tersimpan di dalam *database* yang tentunya akan mempermudah pihak perusahaan dalam melakukan pencarian data.

2.4 Teori Program

2.4.1 Pengertian HTML



Gambar 2.4 Logo HTML

Pengertian HTML menurut Surya dan Jannah (2020:1), "HTML adalah bahasa markah yang digunakan peramban untuk menafsirkan dan menulis teks, gambar, dan bahan lainnya ke dalam halaman web secara virtual maupun suara". Sedangkan menurut Devi (2020:1), "*Hypertext Markup Language* (HTML) adalah sebuah bahasa markah yang digunakan untuk membuat sebuah halaman web, menampilkan berbagai informasi dalam sebuah penjelajah web internet dan memformat hiperteks sederhana yang ditulis dalam berkas format ASCII agar dapat menghasilkan tampilan wujud yang terintegrasi."

Berdasarkan beberapa definisi di atas, maka dapat disimpulkan bahwa HTML (*Hypertext Markup Language*) adalah skrip yang berupa tag-tag untuk membuat dan mengatur struktur serta menandai bagian-bagian dari sebuah halaman website untuk menampilkan konten di web.



2.4.2 Pengertian JavaScript



Gambar 2.5 Logo JavaScript

Menurut Siahaan dan Sianipar, (2020:1), "*Javascript* adalah sebuah skript dinamis yang dapat dipakai untuk membangun interaktifitas pada halaman-halaman HTML statis." Sedangkan menurut Sulistiono (2018:3), "*Javascript* adalah bahasa pemrograman tingkat tinggi untuk membuat website dinamis."

Berdasarkan beberapa definisi di atas, maka dapat disimpulkan bahwa *JavaScript* adalah bahasa pemrograman web yang pemrosesanya dilakukan di sisi client untuk memberikan efek dinamis, interaktif dan bersifat independen, yang dieksekusi di browser.

2.4.3 Pengertian CSS



Gambar 2.6 Logo CSS

Menurut Abdulloh (2016:2), "CSS singkatan dari *Cascading Style Sheets*, yaitu skrip yang digunakan untuk mengatur desain website." Sedangkan menurut Enterprise (2016:94), "CSS merupakan kumpulan kode untuk mendesain atau mempercantik tampilan halaman *website*."



Berdasarkan beberapa definisi di atas, dapat disimpulkan bahwa CSS adalah bahasa pengkodean atau dokumen yang berfungsi mengatur elemen HTML dengan berbagai *property* untuk menata gaya tampilan halaman web agar lebih cantik dan indah saat di tampilkan di web browser.

2.4.4 Pengertian PHP



Gambar 2.7 Logo PHP

Menurut Supono dan Putratama (2018:3), "PHP (PHP: *Hypertext Preprocessor*) adalah suatu bahasa pemrograman yang digunakan untuk menerjemahkan baris kode program menjadi kode mesin yang dapat dimengerti oleh komputer yang bersifat *server-side* yang dapat ditambahkan ke dalam HTML." Pengertian PHP menurut Abdulloh (2016:3), "PHP singkatan dari *Hypertext Preprocessor*, yaitu bahasa pemrograman yang diproses di sisi server."

Berdasarkan beberapa definisi di atas, maka dapat disimpulkan bahwa PHP adalah bahasa pemrograman web berbasis *server-side programming* (bahasa pemrograman yang diproses disisi server) yang memarsing kode PHP dari kode web dengan ekstensi (.php), yang kemudian akan dikirim ke browser web.

2.4.5 Pengertian CodeIgniter



Gambar 2.8 Logo CodeIgniter

Menurut Supono dan Putratama (2018:109), mendefinisikan bahwa "*CodeIgniter* adalah aplikasi *open source* berupa *framework* dengan model MVC



(*Model, View, Controller*) untuk membangun *website* dinamis dengan menggunakan PHP". Sedangkan menurut Habibi dan Aprilian (2019:9), mendefinisikan bahwa "*CodeIgniter* adalah sebuah *web application network* yang bersifat *open source* yang digunakan untuk membangun aplikasi php dinamis."

Berdasarkan beberapa definisi di atas, maka dapat disimpulkan bahwa *CodeIgniter* adalah sebuah *framework* yang bersifat *open source* untuk membangun *website* / aplikasi dengan menggunakan bahasa pemrograman PHP.

2.4.6 Pengertian XAMPP



Gambar 2.9 Logo XAMPP

Menurut Habibi, dkk (2020:5), menyatakan bahwa "XAMPP merupakan perangkat lunak bebas, yang mendukung banyak *system* operasi merupakan kompilasi dari beberapa program, *xampp* adalah perangkat yang menggabungkan tiga aplikasi ke dalam satu paket yaitu *Apache*, *MySQL*, 6 dan *PHP my admin* dengan *xampp* pekerjaan anda sangat dimudahkan karena dapat menginstalasi dan mengkonfigurasi ketiga aplikasi tersebut dengan sekaligus dan otomatis". Pengertian lainnya menurut Nurhidayah, dkk (2020:40), menyatakan bahwa "XAMPP adalah aplikasi yang bersifat *open source* yang bisa digunakan di banyak sistem operasi seperti Windows, Linux, dan Mac OS."

Berdasarkan beberapa definisi di atas, maka dapat disimpulkan bahwa XAMPP merupakan aplikasi pendukung bersifat *open source* yang menggabungkan tiga aplikasi ke dalam satu paket yaitu *Apache*, *MySQL*, 6 dan *PHP my admin* untuk membaca file yang memiliki format PHP.



2.4.7 Pengertian MySQL



Gambar 2.10 Logo MySQL

Menurut Fathoroni, dkk (2020:55), "MySQL adalah *database management system* yang menggunakan bahasa *SQL* sebagai bahasa penghubung antara perangkat lunak aplikasi dengan *database server*". Menurut Fitri (2020:7), "MySQL merupakan *database engine* atau *server database* yang mendukung bahasa *database SQL* sebagai bahasa interaktif dalam mengelola data."

Berdasarkan beberapa definisi di atas, dapat disimpulkan bahwa MySQL adalah *database management system* sebagai bahasa interaktif penghubung perangkat aplikasi dengan *database server* yang digunakan dalam melakukan pengelolaan data.

2.4.8 Pengertian Sublime Text



Gambar 2.11 Logo Sublime Text

Menurut Ardhana (dikutip Sa'ad, 2020:39), "*Sublime Text* adalah salah satu program yang digunakan untuk melakukan *editor*, seperti *HTML*, *PHP*, *CSS*". Sedangkan menurut Harani dan Hasanah (2020:84), menyatakan bahwa "*Aplikasi Sublime Text* merupakan sebuah aplikasi *editor* digunakan pada kode dan teks yang dapat berfungsi di berbagai platform *operating system* (OS) dengan menggunakan teknologi *Phyton API*."



Sublime Text menjunjung berbagai jenis bahasa pemrograman dan dapat menyuguhkan fitur *syntax highlight* hampir pada semua bahasa pemrograman yang didukung maupun dikembangkan oleh komunitas seperti; C, C++, SQL, TCL, Textile, CSS, D, LaTeX, Lisp, Lua, PHP, Python, XML, Dylan, Erlang, Java, JavaScript, OCaml, Perl, C#, HTML, Markdown, MATLAB, Groovy, Ruby, R and Haskell."

Berdasarkan beberapa definisi di atas, dapat disimpulkan bahwa *Sublime Text* merupakan program aplikasi pada komputer yang digunakan sebagai *editor* kode dan teks yang banyak menjunjung tinggi semua jenis bahasa pemrograman.