

BAB II LANDASAN TEORI

2.1. Tinjauan Pustaka

Dalam penelitian ini akan digunakan lima tinjauan pustaka yang nantinya dapat mendukung penelitian, berikut ini merupakan tinjauan studi yang diambil yaitu :

1. Aulia (2020) meneliti tentang Sistem Pengawasan Terhadap Hakim Konstitusi Dalam Mewujudkan Independensi Hakim. Pengawasan internal terhadap MK yang ada saat ini sebagaimana diatur dalam Peraturan MK Nomor 2 Tahun 2014 Tentang Majelis Kehormatan MK, masih memiliki beberapa kekurangan seperti kewenangan pengawasan yang dimiliki oleh Dewan Etik bersifat pasif, hal tersebut dirasa sangatlah longgar mengingat peran Dewan Etik sebagai early warning system, dan keberadaan Dewan Etik yang berada dilingkungan Internal MK. Dewan Etik mengusulkan pembentukan Majelis Kehormatan MK, yang kemudian MK melakukan pemilihan terhadap keanggotaan Majelis Kehormatan, hal tersebut juga dirasa kurang tepat, berkenaan dengan keadaan MK membentuk suatu Majelis yang dipilih oleh dirinya sendiri untuk mengawasi dan melakukan penegakan terhadap dirinya sendiri. Terkait dengan pembentukan Majelis Kehormatan, dilihat perlu adanya penyempurnaan komposisi keanggotaan pada Majelis Kehormatan, dengan melihat bahwa terdapat unsur yang sama antara Dewan Etik dan Majelis Kehormatan.
2. Tuharyadi, Budiman and Rolan (2021) meneliti tentang Rancang Bangun Sistem Informasi Pengawasan Data Pemilih (Sepedah) Pada Bawaslu Kota Jakarta Timur. Tujuan penelitian ini untuk menghasilkan data pemilih secara akurat dengan menggunakan aplikasi perangkat lunak sesuai dengan standar ISO 91 26 dari sisi *Functionality*, *Security*, *Usebility*, *Efficiency* dan *Maintanibility*. Metode Penelitian dalam Rancang Bangun Sistem Informasi Pengawasan Data Pemilih (SEPEDAH) meliputi studi kasus, analisis kebutuhan, design sistem, pemrograman

sistem, testing, implementasi, evaluasi dan penulisan laporan. Untuk perancangan aplikasi menggunakan unified modeling language (UML). Proses pengerjaan penelitian ini dibuat secara interaktif mulai dari literasi kasus sampai dengan tahap implementasi dan pengujian perangkat lunak.

3. Dinata (2016) meneliti tentang Sistem Informasi Pengawasan Taman Kota Pada Dinas Pasar, Kebersihan Dan Pertamanan Kabupaten Aceh Utara. Penggunaan Teknologi Informasi di Dinas Pasar, Kebersihan dan Pertamanan Kabupaten Aceh Utara, memberikan dampak positif bagi kemajuan dinas tersebut. Sistem komputerisasi menciptakan suatu bagian yang sangat terpenting dari kemajuan teknologi dan kemudahan komunikasi saat ini. Untuk itu penulis ingin mengembangkan kecanggihan sistem komputer yang dapat meringankan kerja dari Dinas Pasar, Kebersihan dan Pertamanan. Penulis merancang dan membuat sistem informasi pengawasan taman kota pada Dinas Pasar, Kebersihan dan Pertamanan Kabupaten Aceh Utara sesuai dengan kebutuhan yang diharapkan oleh dinas tersebut. Hasil yang diperoleh dari sistem ini sangat membantu dan mempermudah kinerja dari pegawai di dinas tersebut dalam memantau kondisi pasar dan kebersihan taman kota di Kabupaten Aceh Utara.

4. Munggaran and Simpony (2016) meneliti tentang Sistem Informasi Pengawasan Pemilihan Umum. Pemilihan Umum merupakan mekanisme utama dalam tahapan penyelenggaraan negara dan pembentukan pemerintahan. Proses pelaksanaan Pemilihan Umum tidak terlepas dari berbagai permasalahan yang timbul dari masyarakat, peserta Pemilu, hingga penyelenggara Pemilu. Sistem pengawasan pemilihan umum yang belum efektif menjadi masalah bagi pengawas yang masih menggunakan sistem manual. Dikatakan belum efektif karena masih banyak kendala yang harus dihadapi, perlu sistem pendataan yang lebih efektif

dan efisien untuk dapat menjangkau pengawasan yang lebih luas. Dengan masih digunakannya sistem secara offline, maka pengawas diharuskan untuk mendatangi langsung ke tempat lokasi ini apabila ingin mencari informasi dan melakukan pendataan pengawasan secara tertulis. Itu semua sangat memerlukan waktu yang cukup lama dan mengeluarkan biaya yang tidak sedikit. Adanya website pendataan pengawas online ini, masalah yang ada pada pendataan dalam pengawasan pemilihan umum tersebut dapat terselesaikan dengan baik. Selain semakin meluasnya jangkauan pengawasan, dengan adanya website ini bukti pendataan merupakan data yang sangat valid untuk dijadikan dokumen masukan sistem pengawasan karena pendataan tersebut diinput langsung melalui form laporan oleh pengawas. Pendataan secara online yang dapat diakses pengawas dimanapun dan kapanpun.

5. Palit, Rindengan and Lumenta (2019) meneliti tentang Rancangan Sistem Informasi Keuangan Gereja Berbasis Web Di Jemaat GMIM Bukit Moria Malalayang. Perubahan pada era revolusi industri 4.0 mendasari sektor publik untuk berperan dalam menyederhanakan sistem informasi guna meningkatkan efisiensi dalam penyelenggaraan pemerintah pusat maupun daerah. Penyelenggaraan pemerintah diawasi oleh lembaga pengawasan yakni Inspektorat Jenderal, yang berpusat secara hierarki melaporkan hasil pengawasan kepada Menteri Dalam Negeri dan Presiden. Sementara, pada pemerintah daerah diawasi oleh Inspektorat Provinsi yang bertanggungjawab melaporkan kepada Gubernur. Mendukung revolusi industry 4.0 Inspektorat Jenderal Kemendagri, telah membangun sistem informasi pengawasan di situs lhp.sistemdatabase.online. Pembangunan sistem yang sudah berjalan dari tahun 2018 hingga 2019 akan tetapi, mengalami permasalahan dan memerlukan desain untuk mempermudah pengawasan dan pelaporan. Serta sinkronisasi data antara Inspektorat Jenderal Kementerian Dalam Negeri dan Inspektorat daerah baik provinsi maupun kabupaten/kota.

Data yang dimaksud adalah data temuan hasil pengawasan Inspektorat Kemendagri dan penyelesaian temuan tersebut. Penyelesaian dimaksudkan menjalani fungsi Inspektorat Kemendagri sebagai konsultan. Dan penelitian dilakukan guna menganalisis permasalahan yang terjadi guna perbaikan sistem informasi pengawasan yang lebih sederhana. Tujuannya, mendapatkan data tunggal yang dikoordinasikan antara pemerintah pusat dan daerah. Penelitian dilakukan menggunakan pendekatan post positivsm dengan teori komunikasi pemerintahan. Data dilengkapi melalui wawancara secara mendalam guna perbaikan sistem dan membentuk desain baru pada sistem informasi pengawasan.

2.2. Sistem Informasi

Sistem dapat didefinisikan sebagai kumpulan dari elemen-elemen berupa data, jaringan kerja dari prosedur-prosedur yang saling berhubungan, sumber daya manusia, teknologi baik hardware dan software yang saling berinteraksi sebagai kesatuan untuk mencapai tujuan atau sasaran tertentu yang sama (Maniah and Haminidin, 2017).

sistem adalah sekumpulan unsur atau elemen dan komponen-komponen yang saling berkaitan dan saling mempengaruhi dalam melakukan kegiatan bersama untuk mencapai suatu tujuan tertentu (Fernanda dan Fernando, 2017)

Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya(Wati, Rahmanto and Fernando, 2019)

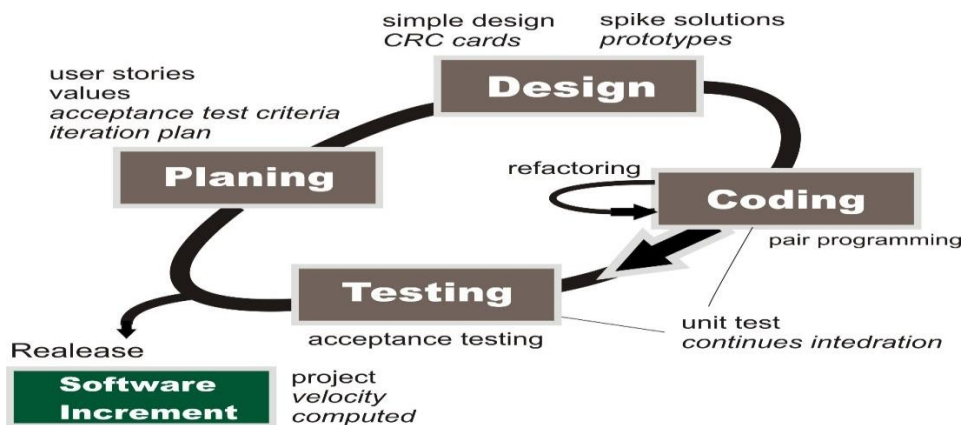
Informasi merupakan kumpulan data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerima. Tanpa suatu informasi, suatu sistem tidak akan berjalan dengan lancar dan akhirnya bisa mati. Suatu organisasi tanpa adanya suatu informasi maka organisasi tersebut tidak bisa berjalan dan tidak bisa beroperasi (Kristanto, 2018).

Sistem informasi ialah suatu sistem yang menyajikan informasi yang bertujuan untuk mengambil suatu keputusan dan menjalankan kegiatan operasional suatu organisasi dan memberikan suatu keunggulan kompetitif (Awaludin, Bahri and Muslih, 2019)

Sistem informasi adalah suatu sistem yang dibuat oleh manusia yang terdiri dari komponen-komponen dalam organisasi untuk mencapai suatu tujuan yaitu menyajikan informasi (Adrian and Pramono, 2017)

2.3. Metode Pengembangan Sistem *Extreme Programming*

Extreme Programming (XP) adalah metodologi pengembangan perangkat lunak yang ditujukan untuk meningkatkan kualitas perangkat lunak dan tanggap terhadap perubahan kebutuhan pelanggan. Jenis pengembangan perangkat lunak semacam ini dimaksudkan untuk meningkatkan produktivitas dan memperkenalkan pos pemeriksaan dimana persyaratan pelanggan baru dapat diadopsi. Tahapan-tahapan dari *Extreme Programming* terdiri dari *planning* seperti memahami kriteria pengguna dan perencanaan pengembangan, *designing* seperti perancangan *prototype* dan tampilan, *coding* termasuk pengintegrasian, dan yang terakhir adalah *testing* (Pressman, 2012)



Gambar 2. 1 Model *Extreme Programming* (XP)
Sumber : (Lestari, Samsugi and Abidin, 2020)

Proses *Extreme Programming* (XP) menurut (Pressman, 2012)

1. *Planning* : Tahap *planning* dimulai dengan membuat *user stories* yang menggambarkan output, fitur, dan fungsi - fungsi dari *software* yang akan dibuat. *User stories* tersebut kemudian diberikan bobot seperti prioritas dan dikelompokkan untuk selanjutnya dilakukan proses *delivery* secara incremental.
2. *Design* : *Design* di *Extreme Programming* mengikuti prinsip *Keep It Simple (KIS)*. Untuk *design* yang sulit, *Extreme Programming* akan menggunakan *Spike Solution* dimana pembuatan design dibuat langsung ke tujuannya. *Extreme Programming* juga mendukung adanya *refactoring* dimana *software system* diubah sedemikian rupa dengan cara mengubah stuktur kode dan menyederhanakannya namun hasil dari kode tidak berubah.
3. *Coding* : Proses *coding* pada *Extereme Programming* diawali dengan membangun serangkaian unit test. Setelah itu pengembangan akan berfokus untuk mengimplementasikannya. Dalam *Exterime Programming* diperkenalkan istilah Pair Programming dimana proses penulisan program dilakukan secara berpasangan. Dua orang *Programmer* saling bekerjasama di satu komputer untuk menulis program. Dengan melakukan ini akan didapat *real-time problem solving* dan *real-time quality assurance*.
4. *Testing* : Tahap ini dilakukan pengujian kode pada unit test. Dalam *Extreme Programming*, diperkenalkan *Extreme Programming acceptance test* atau biasa disebut *customer test*. Tes ini dilakukan oleh customer yang berfokus kepada fitur dan fungsi sistem secara keseluruhan. *Acceptance test* ini berasal dari *user stories* yang telah diimplementasikan.

2.4. Karakteristik Pemodelan Berorientasi Objek

Berikut ini adalah beberapa karakteristik dasar yang harus dipahami tentang metodologi berorientasi objek:

1. Kelas (*Class*)

Kelas adalah kumpulan objek-objek dengan karakteristik yang sama. Sebuah kelas akan mempunyai sifat (atribut), kelakuan (operasi/metode), hubungan (relationship) dan arti.

2. Objek (*Object*)

Objek adalah abstraksi dan sesuatu yang mewakili dunia nyata seperti benda, manusia, satuan organisasi, tempat, kejadian, struktur, status, atau hal-hal lain yang bersifat abstrak.

3. Metode (*Method*)

Operasi atau metode atau method pada sebuah kelas hampir sama dengan fungsi atau prosedur pada metodologi structural. Sebuah kelas boleh memiliki lebih dari satu metode atau operasi.

4. Atribut (*attribute*)

Atribut dari sebuah kelas adalah variable global yang dimiliki sebuah kelas. Atribut dapat berupa nilai atau elemen-elemen data yang dimiliki oleh objek dalam kelas objek. Atribut dipunyai secara individual oleh sebuah objek, misalnya berat, jenis, nama, dan sebagainya.

5. Abstraksi (*abstraction*)

Prinsip untuk mempersentasikan dunia nyata yang kompleks menjadi satu bentuk model sederhana dengan mengabaikan aspek-aspek lain yang tidak sesuai dengan permasalahan.

6. Enkapsulasi (*encapsulation*)

Pembungkusan atribut data dan layanan (operasi-operasi) yang dipunyai objek untuk menyembunyikan implementasi dan objek sehingga objek lain tidak mengetahui cara kerjanya.

7. Pewarisan (*inheritance*)

Mekanisme yang memungkinkan satu objek mewarisi sebagian atau seluruh definisi dan objek lain sebagai bagian dari dirinya.

8. Antarmuka (*interface*)

Antarmuka atau interface sangat mirip dengan kelas, tapi hanya atribut kelas dan memiliki metode yang dideklarasikan tanpa isi.

9. *Reusability*

Pemanfaatan kembali objek yang sudah didefinisikan untuk suatu permasalahan pada permasalahan lain yang melibatkan objek tersebut.

10. Generalisasi dan Spesialisasi

Menunjukkan hubungan antara kelas dan objek yang umum dengan kelas dan objek yang khusus. Misalnya kelas yang lebih umum (generalisasi) adalah kendaraan darat dan kelas khususnya (spesialisasi) adalah mobil, motor, dan kereta.

11. Komunikasi Antar Objek

Komunikasi antar objek dilakukan lewat pesan (message) yang dikirimkan dari satu objek ke objek lainnya.

12. Polimorfisme (*Polymorphism*)

Kemampuan suatu objek untuk digunakan dengan banyak tujuan yang berbeda dengan nama yang sama sehingga menghemat baris program.

13. Package

Package adalah sebuah kontainer atau kemasan yang dapat digunakan untuk mengelompokkan kelas-kelas sehingga memungkinkan beberapa kelas yang bernama sama disimpan dalam package yang berbeda.

2.5. *Unified Modeling Language (UML)*

Bahasa Pemodelan Pengembangan Sistem (*Unified Modeling Language*) adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis & desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek (Rosa and Shalahudin, 2018) (Mukti, Widiyanto and Eridani, 2021). Beberapa jenis diagram *UML* antara lain sebagai berikut:

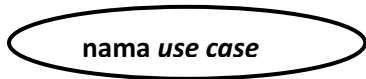

A. *Use Case Diagram*





Menurut (Rosa & Shalahudin, 2018).

“Use case diagram 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. simbol-simbol yang ada pada diagram use case dapat dilihat pada gambar 2.1 di bawah ini:

Tabel 2. 1 Simbol Diagram *Use Case*

Simbol	Deskripsi
<p><i>Use Case</i></p> 	<p>Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i></p>
<p>Aktor/<i>actor</i></p> 	<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</p>

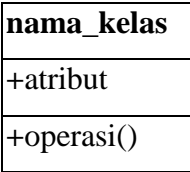



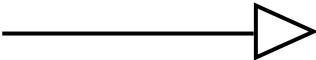

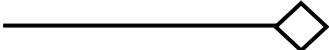
Simbol	Deskripsi
	merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama <i>actor</i>
Asosiasi/ <i>association</i> 	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan <i>actor</i>
Ekstensi/ <i>extend</i>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan
<< <i>extend</i> >> 	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
Generalisasi/ <i>generalization</i> 	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
Menggunakan/ <i>Include/uses</i> << <i>include</i> >> 	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya

Sumber: (Rosa and Shalahudin, 2018).

B. *Class Diagram*

Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi. simbol-simbol yang ada pada diagram kelas pada tabel *class diagram 2.2* di bawah ini:

Tabel 2. 2 Simbol *Class Diagram*

Simbol	Deskripsi
<p>Kelas</p> 	Kelas pada struktur sistem
<p>Antarmuka/<i>Interface</i></p>  <p>nama_interface</p>	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek
<p>Asosiasi/<i>asociation</i></p> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
<p>Asosiasi berarah/<i>directed association</i></p> 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya disertai dengan <i>multiplicity</i>
<p>Generalisasi</p> 	Relasi antar kelas dengan makna generalisasi-spesialisasi(umum khusus)
<p>Kebergantungan/<i>dependecy</i></p> 	Relasi antar kelas dengan makna kebergantungan antar kelas
<p>Agregasi/<i>agregation</i></p> 	Relasi antar kelas dengan makna semua bagian (<i>whole-part</i>)





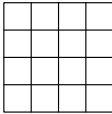


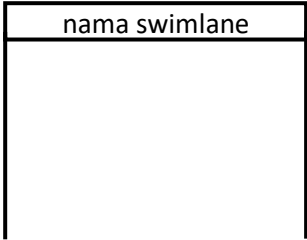
Sumber: (Rosa and Shalahudin, 2018).

C. *Activity Diagram*

Activity diagram atau Diagram aktivitas menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas

yang dapat dilakukan oleh sistem. simbol-simbol yang ada pada *activity diagram* dapat dilihat pada tabel 2.3 di bawah ini :

Tabel 2. 3 Simbol *Activity Diagram*

Simbol	Deskripsi
Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
Percabangan/ <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
Penggabungan/ <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
Tabel 	Suatu file komputer dari mana data bisa dibaca atau direkam selama kejadian bisnis
Dokumen 	Menunjukkan dokumen sumber atau laporan
Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
<i>Swimlane</i> 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

Sumber: (Rosa and Shalahudin, 2018).

2.6. MySQL

MySQL merupakan RDBMS (atau server database) yang mengelola database dengan cepat menampung dalam jumlah sangat besar dan dapat di akses oleh banyak user, MySQL adalah sebuah *software open source* yang digunakan untuk membuat sebuah *database*, berdasarkan pendapat yang dikemukakan di atas dapat ditarik kesimpulan bahwa MySQL adalah suatu *software* atau program yang digunakan untuk membuat sebuah database yang bersifat *open source*(Firman et al, 2016)

2.7. XAMPP

XAMPP adalah perangkat lunak bebas, yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program. Fungsinya adalah sebagai server yang berdiri sendiri (*localhost*), yang terdiri atas program *Apache* HTTP Server, *MySQL database*, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl. Nama XAMPP merupakan singkatan dari X (sistem operasi apapun), *Apache*, *MySQL*, *PHP* dan *Perl*. Program ini tersedia dalam GNU *General Public License* dan bebas, merupakan web server yang mudah digunakan yang dapat melayani tampilan halaman web yang dinamis (Kusbianto, Ardiansyah and Hamadi, 2017).

Berikut ini Penjelasan dari bagian-bagian XAMPP:

1. X , XAMPP bisa dijalankan di 4 OS besar yang sering digunakan oleh pengguna komputer saat ini. Dan 4 OS tersebut tidak lain dan tidak bukan adalah Windows, Linux, Mac OS dan Solaris.
2. A (*Apache*) merupakan aplikasi web server. Apache ini bersifat opensource yang berarti gratis dan bisa diedit oleh penggunanya. Tugas utama Apache adalah menghasilkan halaman web yang benar kepada user berdasarkan kode PHP yang dituliskan oleh pembuat halaman web. jika diperlukan juga berdasarkan kode PHP yang dituliskan,maka dapat saja suatu

database diakses terlebih dahulu (misalnya dalam MySQL) untuk mendukung halaman web yang dihasilkan.

3. M (MySQL), merupakan aplikasi database server. Perkembangannya disebut SQL yang merupakan kepanjangan dari *Structured Query Language*. SQL merupakan bahasa terstruktur yang digunakan untuk mengolah database. MySQL dapat digunakan untuk membuat dan mengelola database beserta isinya. Kita dapat memanfaatkan MySQL untuk menambahkan, mengubah, dan menghapus data yang berada dalam database.
4. P (PHP), bahasa pemrograman web. Bahasa pemrograman PHP merupakan bahasa pemrograman untuk membuat web yang bersifat server-side scripting. PHP memungkinkan kita untuk membuat halaman web yang bersifat dinamis. Sistem manajemen basis data yang sering digunakan bersama PHP adalah MySQL. namun PHP juga mendukung sistem manajemen *database Oracle, Microsoft Access, Interbase, d-base, PostgreSQL*, dan sebagainya.
5. P (Perl), bahasa pemrograman, Dua diantara karakteristik utama perl adalah penanganan teks dan berbagai jalan pintas untuk menyelesaikan persoalan-persoalan umum. Perl sangat populer di gunakan dalam program-program CGI (*Common Gateway Interface*) dan protokol internet lainnya.

2.8. Sublime Text

Sublime text adalah teks editor berbasis *Python*, sebuah teks editor yang elegan, kaya fitur, *cross platform*, mudah dan simpel yang cukup terkenal di kalangan developer (pengembang), penulis dan desainer. Para programmer biasanya menggunakan sublime text untuk menyunting source code yang sedang ia kerjakan. Sampai saat ini sublime text sudah mencapai versi 3(Faridl, 2015). *Sublime text* mempunyai beberapa keunggulan-keunggulan yang dapat membantu

pengguna dalam membuat sebuah *web development*. Berikut keunggulan-keunggulan fitur yang dimiliki Sublime Text, adalah :

1. *Multiple Selection*

Multiple Selection mempunyai fungsi untuk membuat perubahan pada sebuah kode pada waktu yang sama dan dalam baris yang berbeda. *Multiple selection* ini juga merupakan salah satu fitur unggulan dari Sublime Text 3. Kita dapat meletakkan kursor pada kode yang akan di ubah/edit, lalu tekan Ctrl+klik atau blok kode yang akan diubah kemudian Ctrl+D setelah itu kita dapat merubah kode secara bersamaan.

2. *Command Pallete*

Command Pallete mempunyai fungsi yang berguna untuk mengakses file shortcut dengan mudah. Untuk mencari file tersebut kita dapat tekan Ctrl+Shift+P, kemudian cari perintah yang kita inginkan.

3. *Distraction Free*

Mode Fitur ini mempunyai fungsi untuk merubah tampilan layar menjadi penuh dengan menekan SHIFT + F11. Fitur ini sangat dibutuhkan ketika pengguna ingin fokus pada pekerjaan yang sedang dikerjakannya.

4. *Find in project* Fitur ini kita dapat mencari dan membuka file di dalam sebuah project dengan cepat dan mudah. Hanya dengan menekan Ctrl+P anda dapat mencari file yang diinginkan.

5. *Plugin API*

Switch Sublime Text mempunyai keunggulan dengan plugin yang berbasis Python Plugin API. Teks editor ini juga mempunyai plugin yang sangat beragam, dan ini dapat memudahkan pengguna dalam mengembangkan softwarena.

6. *Drag and Drop*

Dalam teks editor ini pengguna dapat menyeret dan melepas file teks ke dalam editor yang akan membuka tab baru secara otomatis.

7. *Split Editing*

Di dalam fitur ini pengguna dapat mengedit file secara berdampingan dengan klik File->New menu into file

8. *Multi Platform*

Sublime Text juga mempunyai keunggulan dalam berbagai platform. *Sublime text* sendiri sudah tersedia dalam berbagai platform sistem operasi, yaitu Windows, Linux, dan MacOS.

2.9. Pengujian ISO 25010

ISO/IEC 25010 merupakan model kualitas sistem dan perangkat lunak yang menggantikan ISO/IEC 9126 (Iqbal, 2016). *Product quality* ini juga digunakan untuk tiga model kualitas yang berbeda untuk produk perangkat lunak antara lain:

1. Kualitas dalam model penggunaan
2. Model kualitas produk
3. Data model kualitas

Adapun dimensi yang pertama terdapat beberapa faktor elemen diantaranya :

- 1) *Functionality* (Fungsionalitas). Kemampuan perangkat lunak untuk menyediakan fungsionalitas yang dibutuhkan ketika perangkat lunak digunakan pada kondisi spesifik tertentu dalam hal ini perangkat lunak dapat memenuhi kelayakan dari sebuah fungsi untuk melakukan pekerjaan yang spesifik bagi pengguna dan dapat memberikan hasil yang tepat dan ketelitian terhadap tingkat kebutuhan pengguna.
- 2) *Reliability* Merupakan tingkatan dimana perangkat lunak dapat bertahan pada tingkatan tertentu ketika digunakan oleh pengguna pada kondisi yang spesifik dalam hal ini

perangkat lunak dapat beroperasi dan siap ketika dibutuhkan untuk digunakan dan juga dapat bertahan pada tingkat kemampuan tertentu terhadap kegagalan, kesalahan serta perangkat lunak kembali pada tingkat tertentu dalam mengembalikan pengembalian data yang disebabkan kegagalan atau kesalahan pada perangkat lunak.

- 3) *Performance efficiency* Merupakan tingkatan dimana perangkat lunak dapat memberikan kinerja terhadap sejumlah sumber daya yang digunakan pada kondisi tertentu dalam hal ini *performance efficiency* dapat memberikan reaksi dan waktu yang dibutuhkan ketika melakukan aksi dari sebuah fungsi dan perangkat lunak dapat menggunakan sejumlah sumber daya ketika melakukan aksi dari sebuah fungsi.
- 4) *Usability* Perangkat lunak dapat dimengerti, dipelajari, digunakan, dan menarik pengguna ketika digunakan dalam hal ini perangkat lunak mudah dipelajari oleh pengguna, perangkat lunak dapat digunakan dan dioperasikan oleh pengguna.
- 5) *Security* Merupakan perlindungan terhadap perangkat lunak dari berbagai ancaman atau keganjalan dalam hal ini perangkat lunak memiliki perlindungan terhadap data atau informasi dari pengguna dan merupakan dari kelengkapan, ketepatan dari sejumlah *asset* yang telah dijaga sehingga aksi atau tindakan yang dilakukan telah terbukti dan hal tersebut tidak dapat ditolak.
- 6) *Compatibility* Faktor ini merupakan kemampuan dari dua atau lebih komponen perangkat lunak dapat melakukan pertukaran informasi dan melakukan fungsi yang dibutuhkan ketika digunakan pada *hardware* atau lingkungan perangkat lunak yang sama.
- 7) *Maintainability* Merupakan tingkat dimana sebuah perangkat lunak dapat dimodifikasi. Dalam hal ini modifikasi adalah perbaikan, perubahan atau penyesuaian perangkat lunak untuk dapat berubah pada lingkungan, kebutuhan dan fungsionalitas yang spesifik.

Selain itu perangkat lunak dapat dianalisis untuk mengetahui apa yang menyebabkan kegagalan pada perangkat lunak untuk mengidentifikasi bagian yang dapat dimodifikasi.

- 8) *Transferability*. Merupakan kemudahan dimana sistem atau komponen dapat berpindah dari lingkungan satu ke lingkungan yang lain dalam hal ini perangkat lunak dapat beradaptasi dengan cepat pada spesifikasi lingkungan yang berbeda tanpa menerapkan aksi atau cara lain dari pada memberikan tujuan tertentu terhadap perangkat lunak yang telah ada.

2.10. Skala Pengukuran

Skala pengukuran yang digunakan adalah skala Likert, skala yang didasarkan pada penjumlahan sikap responden dalam merespon pernyataan berkaitan indikator-indikator suatu konsep atau variable yang sedang diukur (Sugiyono, 2018). Skala Likert umumnya menggunakan lima titik dengan label netral pada posisi tengah (ketiga). Skala Likert dapat dilihat pada Tabel 2.4 :

Tabel 2. 4 Skala Likert

Jawaban	Skor
Sangat Setuju	5
Setuju	4
Netral	3
Tidak Setuju	2
Sangat Tidak Setuju	1

Sumber : (Sugiyono, 2018)

Hasil penilaian responden akan dihitung *persentase* kelayakannya dengan menggunakan perhitungan, dapat dilihat dibawah ini:

$$Persentase = \frac{Skor\ Aktual\ (f)}{Skor\ Ideal\ (n)} \times 100\%$$

Persentase kelayakan yang diperoleh kemudian dibandingkan dengan Tabel konversi yang berpedoman pada acuan konversi nilai, dapat dilihat pada Tabel 2.5

Tabel 2. 5 Skala Konversi Nilai

Persentase Pencapaian (%)	Interpretasi
$90 \leq x$	Sangat Baik
$80 \leq x < 90$	Baik
$70 \leq x < 80$	Cukup
$60 \leq x < 70$	Kurang
$X < 60$	Sangat Kurang

Sumber : (Sugiyono, 2018)

Keterangan: x = persentase hasil pengujian