

BAB II LANDASAN TEORI

1.1. Tinjauan Pustaka

Beberapa penelitian yang berkaitan dengan penerapan sistem pendukung bantuan operasional pendidikan dalam menentukan kelayakan berdasarkan jurnal penelitian terlihat pada Tabel 2.1:

Tabel 2.1 Tinjauan Pustaka

NO	Penulis (Tahun)	Judul	Metode	Hasil
1.	(Irawan and Wahyuni, 2019)	Sistem Pendukung Keputusan Penerimaan Siswa Baru Di SMK Negeri 1 Tapung Hulu Menggunakan Metode Simple Multi Attribut Rating Technique (SMART)	Metode Simple Multi Attribute Rating Technique	Sistem pendukung keputusan yang dibangun sangat membantu untuk mempercepat pengolahan data dalam pengambilan keputusan untuk menentukan calon siswa baru yang layak dan tidak layak untuk diterima.
2.	(Ismail, 2019)	Sistem Pendukung Keputusan Penerimaan Siswa Baru Menggunakan Algoritma Smart Pada Sekolah	Metode SMART	system penunjang keputusan penerimaan siswa baru menggunakan metode SMART., pekerjaan penentuan kelulusan menjadi mudah,
Tabel 2.1 Tinjauan Pustaka (Lanjutan)				
		Soppeng		akurat.
3	(Andani, 2019)	Penerapan Metode SMART	Metode SMART	Hasil dari penelitian ini adalah rekomendasi

NO	Penulis (Tahun)	Judul	Metode	Hasil
		Dalam Pengambilan Keputusan Penerima Beasiswa Yayasan AMIK Tunas Bangsa		penerima beasiswa. Dengan menggunakan metode SMART, maka rekomendasi yang dihasilkan lebih akurat
4	(Yani, Beny and Harti, 2019)	Perancangan Sistem Pendukung Keputusan Seleksi Penerimaan Beras Sejahtera (Rastra) Pada Kelurahan Tanjung Pinang Jambi	Metode SMART	Aplikasi dapat digunakan oleh dapat memberikan kemudahan memilih calon penerima yang layak mendapatkan bantuan beras sejahtera dengan cepat dan tepat
5	(Honggowibowo, 2015)	Sistem Pendukung Keputusan Penerimaan Calon Mahasiswa Baru Jalur Prestasi Di Sekolah Tinggi Teknologi Adisutjipto Menggunakan Simple Multi Attributerating Technique	Metode SMART	Hasil pengujian sistem, didapatkan kesimpulan bahwa Metode Simple Multi Attribute Rating Technique cukup efektif untuk diterapkan dalam menentukan penerimaan calon mahasiswa baru di STTA

Berdasarkan tabel penelitian terdahulu terdapat perbedaan dengan penelitian yaitu penelitian (Irawan and Wahyuni, 2019) meneliti tentang Sistem Pendukung Keputusan Penerimaan Siswa Baru Di SMK Negeri 1 Tapung Hulu Menggunakan Metode Simple Multi

Attribut Rating Technique (SMART) pada penelitian ini tidak membuat sistem pendaftaran siswa hanya membangun sistem untuk penilaian penerimaan siswa saja dan menggunakan metode pengembangan sistem *waterfall*. Sedangkan peneliti membangun sistem pendaftaran yang nantinya akan terhitung otomatis penilaian yang dilakukan dengan menggunakan metode SMART dan pengembangan sistem *extreme programming*.

Perbedaan penelitian (Ismail, 2019) meneliti tentang Sistem Pendukung Keputusan Penerimaan Siswa Baru Menggunakan Algoritma Smart Pada Sekolah Menengah Atas Negeri 2 Soppeng, pada penelitian ini sistem yang dibangun tidak melakukan pengupload-an berkas pendaftaran siswa dan perancangan sistem menggunakan UML. Sedangkan peneliti membangun sistem pendaftaran yang nantinya siswa dapat mengupload berkas, dan perancangan sistem menggunakan UML.

Perbedaan penelitian (Andani, 2019) meneliti tentang Penerapan Metode SMART Dalam Pengambilan Keputusan Penerima Beasiswa Yayasan AMIK Tunas Bangsa, pada penelitian ini tidak membuat sistem pendaftaran siswa hanya membangun sistem untuk penilaian penerimaan siswa saja dan kriteria yang digunakan dalam menentukan penerima beasiswa yayasan pada penelitian ini adalah IPK, penghasilan orang tua dan jumlah tanggungan. Sedangkan peneliti membangun sistem pendaftaran yang nantinya akan terhitung otomatis penilaian yang dilakukan dengan menggunakan metode SMART dan menggunakan kriteria nilai ijazah, prestasi, pendapatan orang tua dan kepribadian.

Selanjutnya penelitian (Yani, Beny and Harti, 2019) meneliti tentang Perancangan Sistem Pendukung Keputusan Seleksi Penerimaan Beras Sejahtera (Rastra) Pada Kelurahan Tanjung Pinang Jambi, pada penelitian ini tidak membuat sistem pendaftaran siswa hanya membangun sistem untuk penilaian penerimaan siswa saja dan menggunakan metode pengembangan sistem *waterfall*. Sedangkan peneliti membangun sistem pendaftaran yang

nantinya akan dihitung otomatis penilaian yang dilakukan dengan menggunakan metode SMART dan pengembangan sistem *extreme programming*.

Teakhir perbedaan penelitian (Honggowibowo, 2015) meneliti tentang Sistem Pendukung Keputusan Penerimaan Calon Mahasiswa Baru Jalur Prestasi Di Sekolah Tinggi Teknologi Adisutjipto Menggunakan *Simple Multi Attributerating Technique*, pada penelitian ini sistem yang dibangun tidak melakukan pendaftaran siswa dan pengupload-an berkas pendaftaran siswa serta perancangan sistem menggunakan UML. Sedangkan peneliti membangun sistem pendaftaran yang nantinya siswa dapat mengupload berkas, dan perancangan sistem menggunakan UML.

1.2. Sistem Pendukung Keputusan

Sistem Pendukung Keputusan (SPK) merupakan suatu ilmu untuk memecahkan masalah dan melakukan komunikasi terhadap masalah semi terstruktur dan tidak terstruktur, tidak seorangpun mutlak mengetahui keputusan bagaimana seharusnya dibuat. SPK dapat menyediakan informasi, prediksi dan mengarahkan pengguna informasi untuk melakukan pengambilan keputusan secara tepat dan baik (Vikasari, 2018).

Terdapat dua model pengambilan keputusan, yaitu model sistem tertutup dan model sistem terbuka (Davis and Hartono, 2013).

a. Model Sistem Tertutup

Model sistem tertutup dilandasi asumsi bahwa keputusan dapat diambil tanpa campur tangan dari lingkungan (luar) sistem, karena sistem pengambilan keputusan tidak dipengaruhi oleh lingkungan. Dalam hal ini sistem pengambilan keputusan dianggap:

- 1) Mengetahui semua alternatif tindakan untuk menanggapi permasalahan dengan segala konsekuensinya.
- 2) Memiliki metode untuk menyusun alternatif-alternatif sesuai prioritasnya.

- 3) Dapat memilih/menetapkan alternatif yang paling menguntungkan, misalnya dari segi laba, manfaat, dan lain-lain.

b. Model Sistem Terbuka

Model sistem terbuka dilandasi asumsi bahwa sistem pengambilan keputusan dan lingkungan memiliki hubungan saling pengaruh. Keputusan yang diambil akan berdampak terhadap lingkungan dan sebaliknya lingkungan juga berpengaruh terhadap sistem pengambilan keputusan. Dalam hal ini sistem pengambilan keputusan dianggap:

- 1) Hanya mengetahui sebagian saja dari alternatif-alternatif untuk menangani permasalahan dengan segala konsekuensinya.
- 2) Hanya dapat menyajikan sejumlah alternatif yang baik untuk menangani permasalahan, tetapi tidak dapat memilih/menetapkan alternatif yang paling menguntungkan.
- 3) Sekadar mempersilakan pemilihan alternatif terbaik untuk dilakukan oleh pihak diluar sistem sesuai dengan aspirasinya.

1.3. Penerimaan Siswa Baru

Penerimaan peserta didik baru adalah menerima peserta didik baru dengan melakukan pendaftaran dan persyaratan yang telah di beritahukan oleh lembaga pendidikan untuk memberikan pendidikan kepada masyarakat dalam rangka memperoleh hak-hak sebagai warga Negara.

Penerimaan siswa baru merupakan salah satu kegiatan yang pertama dilakukan, biasanya dengan mengadakan seleksi calon siswa. Pengelolaan siswa baru ini harus dilakukan secara terorganisir dan terencana, sehingga kegiatan pembelajaran dapat dilaksanakan pada hari pertama setiap tahun ajaran baru. Mati atau hidupnya suatu sekolah ditentukan oleh ada-tidaknya siswa yang bersekolah. Menurut Undang- Undang Nomor 20 Tahun 2002 Tentang Sistem Pendidikan Nasional Pasal 1 Ayat 4. Siswa adalah anggota masyarakat yang berusaha

mengembangkan potensi dirinya melalui proses pembelajaran yang tersedia pada jalur, jenjang, dan jenis pendidikan tertentu. Agar siswa dapat mengembangkan potensi dirinya melalui proses pembelajaran yang tersedia di sekolah, maka siswa harus di persiapkan secara baik dan terprogram. Penyiapan dan pembinaan kesiswaan dalam proses pendidikan tertuang dalam Undang-Undang Sistem Pendidikan Nasional (UUSPN), bahwa kesiswaan sebagai kader penerus perjuangan bangsa dan pembangunan nasional harus dipersiapkan sebaik-baiknya serta dihindarkan dari segala kendala yang merusaknya dengan memberikan bekal secukupnya dalam kepemimpinan Pancasila, pengetahuan dan keterampilan (Ismail, 2019).

1.4. Metode SMART

Metode SMART merupakan metode pengambilan keputusan multi kriteria yang dikembangkan oleh Edward pada tahun 1977. SMART merupakan teknik pengambilan keputusan multi kriteria ini didasarkan pada teori bahwa setiap alternatif terdiri dari sejumlah kriteria yang memiliki nilai-nilai dan setiap kriteria memiliki bobot yang menggambarkan seberapa penting ia dibandingkan dengan kriteria lain. Pembobotan ini digunakan untuk menilai setiap alternatif agar diperoleh alternatif terbaik. SMART menggunakan linear additive model untuk meramal nilai setiap alternatif. SMART merupakan metode pengambilan keputusan yang fleksibel. SMART lebih banyak digunakan karena kesederhanaanya dalam merespon kebutuhan pembuat keputusan dan caranya menganalisa respon. Analisa yang terlibat adalah transparan sehingga metode ini memberikan pemahaman masalah yang tinggi dan dapat diterima oleh pembuat keputusan. Pembobotan dan pemberian peringkat ini digunakan untuk menilai setiap alternatif agar diperoleh alternatif terbaik. Pembobotan pada SMART (*Simple Multi Attribute Rating Technique*) menggunakan skala antara 0 sampai 1, sehingga mempermudah perhitungan dan perbandingan nilai pada masing-masing alternatif (Salim And Nugroho, 2017). Adapun langkahnya sebagai berikut :

Pertama yaitu menentukan menentukan kriteria yang digunakan dalam menyelesaikan masalah pengambilan keputusan. Untuk menentukan kriteria-kriteria apa saja yang digunakan dalam sistem pengambilan keputusan ini diperlukan data-data dari pengambil keputusan atau pihak yang berwenang/kompeten terhadap masalah yang akan diselesaikan.. Langkah kedua yaitu menentukan alternatif yang akan digunakan untuk perhitungan. Langkah ketiga yaitu penentuan bobot kriteria yang dimana di beri interval 1-100 untuk tiap-tiap kriteria dengan prioritas terpenting. Langkah keempat yaitu melakukan normalisasi bobot kriteria yaitu menggunakan persamaan :

$$u(a_i) = \sum_{j=1}^m w_j u_i(a_i) \dots \dots \dots (1)$$

Keterangan:

$i = 1, 2, \dots, m$

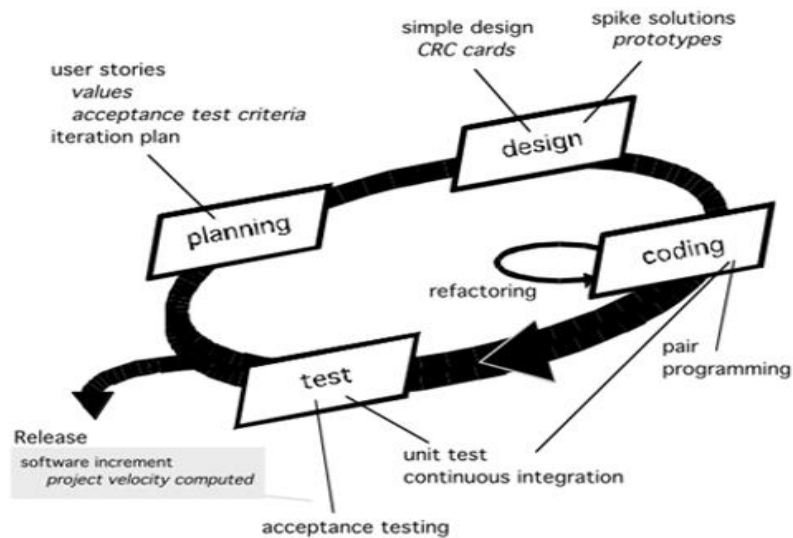
w_j = nilai pembobotan kriteria ke-j dan k kriteria

$u(a_i)$ = nilai utility kriteria ke-i untuk kriteria ke-i

Pemilihan keputusan adalah mengidentifikasi mana dari n alternatif yang mempunyai nilai fungsi terbesar.

1.5. Metode Pengembangan Sistem

Extreme Programming (XP) atau Pemograman Ekstreme yaitu suatu pendekatan yang paling banyak digunakan untuk pengembangan perangkat lunak cepat (Pressman, 2012). Pada pengembangan ini terdapat beberapa konteks kegiatan kerangka kerja, perencanaan, perancangan, pengkodean, dan pengujian.



Gambar 2. 1 Ilustrasi Proses *Extreme Prgamming*

Sumber: (Pressman, 2012)

Berikut ini adalah penjelasan tahapan *Extreme Programming* (Pressman, 2012), yaitu :

1. *Planning* (Perencanaan)

Kegiatan Perencanaan (disebut juga *planning game*) biasanya dimulai dengan mendengarkan suatu kegiatan yang bertujuan mengumpulkan kebutuhan-kebutuhan untuk memahami konteks bisnis dan perlunya keluaran-keluaran (*output*), fungsi utama, dan *fungsi*alitas.

Pada perencanaan terdapat *user stories values* yaitu story dengan value tertinggi akan dipindahkan dari jadwal dan diimplementasikan pertama, *acceptance test criteria iteration plan* melakukan perhitungan kecepatan project selama development, customer dapat menambah story, merubah value, membagi story atau menghapusnya.

2. *Design* (Perancangan)

Perancangan yang simple, menarik, dan sederhana selalu memberikan hasil yang lebih disukai daripada gambaran-gambaran yang lebih kompleks. Perancangan XP memberikan panduan implementasi untuk suatu cerita ketika ditulis, tidak kurang, tidak lebih.

Terdapat *simple design CRC Cards* untuk mengenali dan mengatur *object oriented class* sesuai dengan *software increment* dan *spike solutions prototypes* melakukan spesifikasi solusi dari *object oriented class*.

3. *Coding* (Pengkodean)

Pengkodean ini dilanjutkan setelah cerita yang telah dikembangkan dan rancangan yang telah dilakukan oleh tim perangkat lunak. Pengkodean ini tidak langsung mengarah ke kode-kode program. Tim akan mengembangkan serangkaian unit pengujian lalu beralih ke pengkodean.

Pada tahapan *pair programming* melakukan kerja sama untuk membuat code dari satu story. Dan *refactoring* adalah proses restrukturisasi kode program komputer yang ada tanpa mengubah perilaku eksternalnya.

4. *Pengujian* (Pengujian)

Unit pengujian yang harus dibuat dan kemudian dijalankan menggunakan kerangka kerja yang memungkinkan mereka untuk diotomatisasi sehingga dapat dijalankan dengan mudah dan dapat dijalankan berulang kali.

Pada tahapan pengujian yaitu *unit test continuous integration* yaitu tahapan pengujian code yang diintegrasikan dengan kerja lainnya dengan pengujian yang dilakukan oleh customer dan focus pada keseluruhan dan fungsional sistem, dan *acceptance testing* yaitu pengujian yang dilakukan *customer stories* yang akan diimplementasikan sebagai bagian dari *software realease*.

Selanjutnya terdapat tahapan *software increment project velocity computed* yaitu tahapan yang telah diimplementasikan dari *software realease* yang nantinya akan diterapkan dalam suatu sistem.



1.6. Bahasa Pemodelan Pengembangan Sistem (UML)

Bahasa Pemodelan Pengembangan Sistem (*Unified Modeling Language*) adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement* (kebutuhan), membuat analisis & desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek (Rosa & Shalahuddin, 2018).

1.6.1. Use Case Diagram

Use case diagram atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat (Rosa & Shalahuddin, 2018). *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat menjelaskan simbol-simbol yang ada pada diagram *use case* dapat dilihat pada gambar 2.2 di bawah ini:

Tabel 2. 2 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 menggunakan kata</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 merupakan orang, biasanya dinyatakan</p>

Tabel 2.2 Simbol Diagram Use Case (Lanjutan)

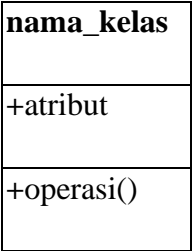





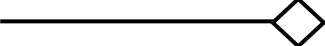
Simbol	Deskripsi
	menggunakan kata benda di awal frase nama <i>actor</i>
Asosiasi/ <i>association</i> —————	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan <i>actor</i>
Ekstensi/ <i>extend</i> <i><<extend>></i> →	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan 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 atau sebagai syarat dijalankan <i>use case</i> ini

Sumber : (Rosa & Shalahuddin, 2018)

1.6.2. 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 (Rosa and Shalahudin, 2018). menjelaskan simbol-simbol yang ada pada diagram kelas pada tabel *class diagram* 2.3.

Tabel 2. 3 Simbol Class Diagram





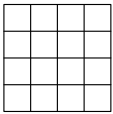


Simbol	Deskripsi
Kelas 	Kelas pada struktur sistem
Antarmuka/ <i>Interface</i>  nama_interface	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek
Asosiasi/ <i>asociation</i> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
Asosiasi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

Sumber : (Rosa & Shalahuddin, 2018)

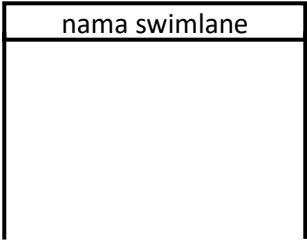
1.6.3. Activity Diagram

Activity diagram atau Diagram aktivitas menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis (Rosa & Shalahuddin, 2018), menjelaskan Simbol-simbol yang ada pada *activity diagram* dapat dilihat pada tabel 2.4 di bawah ini :

Tabel 2. 4 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 bergabung 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.

Tabel 2.4 Simbol Activity Diagram (Lanjutan)

Simbol	Deskripsi
<p><i>Swimlane</i></p> 	<p>Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi</p>

Sumber : (Rosa & Shalahuddin, 2018)

1.7. Alat Implementasi

1.7.1. 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.

1.7.2. 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 softwarentya.

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.

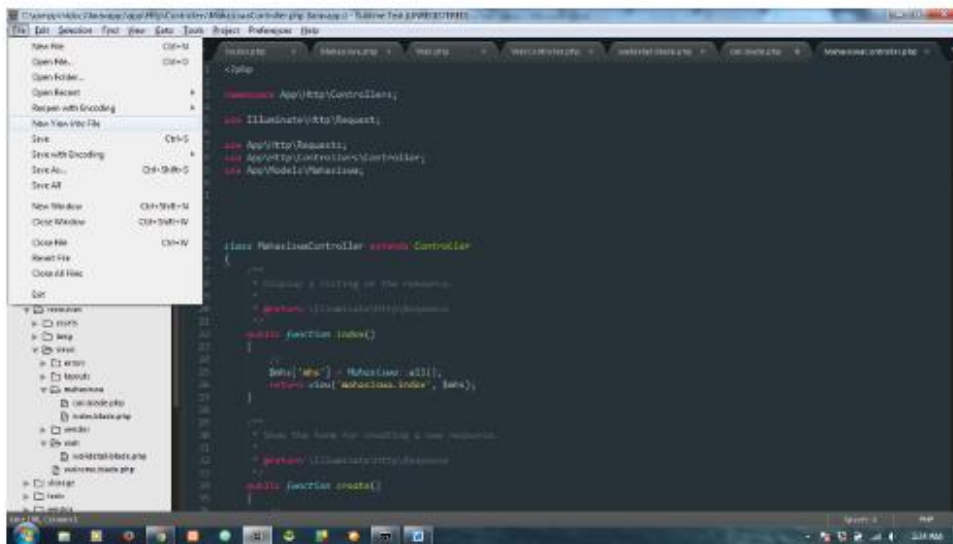
7. *Split Editing*

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.

Hasil pembuatan web yang dibangun menggunakan bahasa PHP dan aplikasi *sublime text* dengan tampilan text editor sebagai berikut :

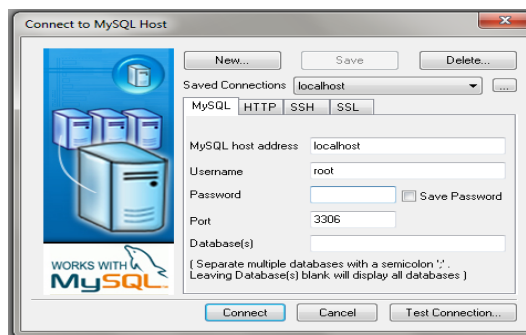


Gambar 2.2 Text Editor Sublime Text
Sumber : (Faridl, 2015)

1.7.3. MySQL

MySQL merupakan *My Structured Query Language* (MySQL) adalah program manajemen basis data atau pabrikan dan sering disebut Manajemen Database Sistem (DBMS). Ini adalah sifat *open source* DBMS. MySQL juga merupakan basis data yang diakses jaringan program, sehingga dapat digunakan untuk multiuser aplikasi (banyak pengguna). Keuntungan lain MySQL menggunakan bahasa query (permintaan) SQL standar. SQL adalah bahasa *query* terstruktur, SQL telah distandarisasi untuk semua program yang diakses *database* (Wadisman and Nozomi, 2019).

MySQL adalah nama *database* server. *Database* server adalah server yang berfungsi untuk menangani database. Database adalah suatu pengorganisasian data dengan tujuan memudahkan penyimpanan dan pengaksesan data. MySQL tergolong sebagai database relasional. pada model ini, data dinyatakan dalam bentuk dua dimensi yaitu secara khusus dinamakan tabel, tabel tersusun atas baris dan kolom seperti gambar berikut:



Gambar 2. 2 login koneksi SQLyog

Sumber : (Yanto, 2016)

Berikut informasi yang harus diisi untuk melakukan koneksi *database server* MySQL:

1. *MySQL Host Address* diisi dengan nama *host* atau *IP address* dari *MySQL server* yang akan dihubungkan. Nilai *defaultnya* adalah *localhost*.
2. *Username* diisi dengan nama pemakai untuk melakukan koneksi ke *MySQL server*. Nama pemakai yang dimasukkan adalah nama pemakai *MySQL*, bukan nama pemakai *FTP* atau *Web Server*. *User defaultnya* adalah *root*.
3. *Password* diisi dengan *password* untuk nama pemakai *MySQL*.

4. *Port* diisi dengan nomor port TCP/IP untuk melakukan koneksi ke MySQL *server*. Nomor *port defaultnya* adalah 3306.

Setelah koneksi ke *database server* MySQL berhasil, maka akan muncul jendela utama SQLyog seperti gambar berikut :



Gambar 2. 3 Jendela Utama SQLyog

Sumber : (Yanto, 2016)

Jendela utama SQLyog dibagi menjadi 3 *panel*, yaitu *object browser*, *SQL window*, dan *result window*:

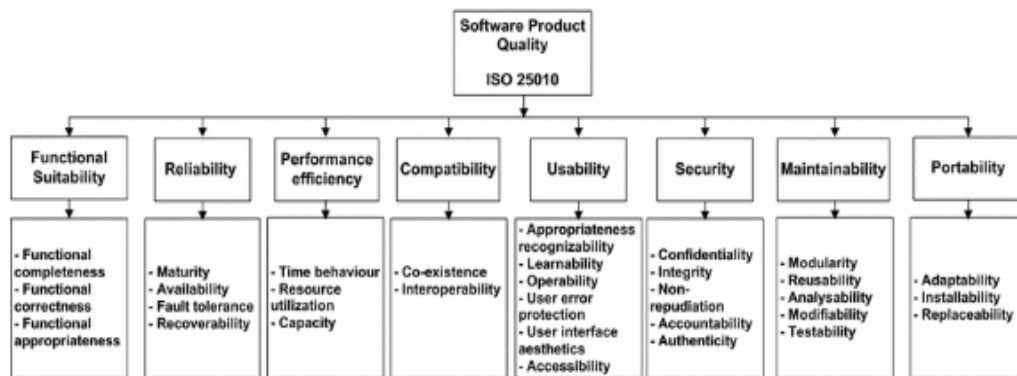
1. *Object browser* menampilkan secara detail semua *database*, tabel, kolom, dan *indeks* yang terdapat pada MySQL *server*.
2. *SQL Window* merupakan jendela tempat mengetikkan perintah SQL yang ingin dieksekusi.
3. *Result window* menampilkan hasil *query* yang dieksekusi. Jendela ini juga menampilkan informasi tentang *database*, tabel, kolom, dan *indeks* yang dipilih pada *object browser*.

1.8. Pengujian Sistem ISO 25010

Model ISO-25010 merupakan bagian dari *Software product Quality Requirements and Evaluation (SQuaRE)*, yang merupakan pengembangan dari model kualitas perangkat lunak sebelumnya yaitu ISO-9126. Dalam model ISO-25010 ini digunakan untuk melihat kualitas suatu perangkat lunak yang digunakan oleh perusahaan, instansi ataupun organisasi. Metode ISO 25010 ini dapat digunakan untuk mengevaluasi kualitas sistem perangkat lunak secara

spesifik berdasarkan dua dimensi umum, yaitu dimensi *product quality*, dimana prosesnya mengacu pada karakteristik intrinsik dari sebuah produk perangkat lunak, memiliki beberapa elemen antara lain meliputi *functional suitability*, *reliability*, *operability*, *performance efficiency*, *security*, *compatibility*, *maintainability* dan *transferability*. *Quality in use* dan *product quality* (Alfian, 2017). *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



Gambar 2. 4 Model kualitas produk ISO/IEC 25010

Sumber: (Kurniawan, Arifianto and Muharom, 2018)

Adapun untuk mengetahui gambaran kualitas model ISO-25010 yang terdiri dari dua dimensi umum, yaitu dimensi *product quality* dan *dimensi quality in use*. Adapun dimensi yang pertama terdapat beberapa faktor elemen diantaranya :

- 1) Fungsionalitas merupakan kemampuan perangkat lunak berada pada level di mana perangkat lunak dapat menyediakan fungsionalitas yang dibutuhkan ketika perangkat lunak digunakan pada kondisi tertentu, dalam hal ini perangkat lunak dapat memenuhi kelayakan suatu fungsi untuk melakukan pekerjaan tertentu bagi pengguna dan dapat memberikan hasil yang tepat serta akurasi untuk kebutuhan pengguna tingkat. Ciri-ciri tersebut terbagi menjadi beberapa ciri yaitu.

- a. Kelengkapan fungsional, sejauh mana fungsi yang disediakan mencakup semua tugas dan tujuan pengguna tertentu.
 - b. Ketepatan fungsional, sejauh mana produk atau sistem memberikan hasil yang benar sesuai kebutuhan.
 - c. Kesesuaian fungsional, sejauh mana fungsi yang diberikan mampu memfasilitasi penyelesaian tugas dan tujuan tertentu.
- 2) Kompatibilitas, sejauh mana produk, sistem atau komponen dapat bertukar informasi dengan produk, sistem atau komponen atau melakukan fungsi lain yang diperlukan secara bersamaan ketika berbagi lingkungan perangkat keras dan perangkat lunak yang sama. Ciri ini terbagi menjadi 2 ciri yaitu:
- a. Koeksistensi, sejauh mana produk atau sistem dapat melakukan fungsi yang diperlukan secara efisien sambil berbagi sumber daya dengan produk atau sistem lain tanpa merusak produk atau sistem.
 - b. Interoperabilitas, sejauh mana dua atau lebih produk, sistem atau komponen dapat bertukar informasi dan menggunakan informasi tersebut.
- 3) Kegunaan, sejauh mana suatu produk atau sistem dapat digunakan oleh pengguna tertentu untuk mencapai tujuan secara efektif, efisiensi, dan kepuasan tertentu dalam konteks penggunaan. Ciri ini terbagi menjadi beberapa ciri yaitu:
- a. Ketepatan pengenalan, sejauh mana pengguna dapat mengetahui apakah sistem atau produk memenuhi kebutuhan mereka.
 - b. Learnability, sejauh mana suatu produk atau sistem dapat digunakan oleh pengguna untuk mencapai tujuan tertentu dengan belajar menggunakan sistem atau produk tersebut secara efisien, efektif, bebas dari risiko dan kepuasan dalam konteks tertentu.

- c. Operabilitas, sejauh mana produk atau sistem mudah dioperasikan dan dikendalikan.
 - d. Perlindungan kesalahan pengguna, sejauh mana produk atau sistem melindungi pengguna dari membuat kesalahan.
 - e. Estetika antarmuka pengguna, sejauh mana antarmuka pengguna suatu produk atau sistem memungkinkan interaksi yang ramah pengguna dan memuaskan.
 - f. Aksesibilitas, sejauh mana suatu produk atau sistem dapat digunakan oleh semua kelompok untuk mencapai tujuan tertentu sesuai dengan konteks penggunaannya.
- 4) Reliabilitas Merupakan tingkat dimana perangkat lunak dapat bertahan pada tingkat tertentu ketika digunakan oleh pengguna dalam kondisi tertentu, dalam hal ini perangkat lunak dapat beroperasi dan siap pada saat dibutuhkan untuk digunakan dan juga dapat menahan tingkat kemampuan tertentu terhadap kegagalan, kesalahan dan pengembalian perangkat lunak sampai tingkat tertentu dalam memulihkan pengembalian data karena kegagalan atau kesalahan perangkat lunak. Ciri-ciri tersebut terbagi menjadi beberapa sub ciri, yaitu:
- a. Kematangan, sejauh mana suatu produk atau sistem dapat memenuhi kebutuhan dengan andal dalam keadaan normal.
 - b. Ketersediaan, sejauh mana produk atau sistem siap untuk dioperasikan dan dapat diakses pada saat perlu digunakan.
 - c. Toleransi kesalahan, sejauh mana produk atau sistem tetap berjalan sebagaimana mestinya meskipun ada kesalahan pada perangkat keras atau perangkat lunaknya.
 - d. Dapat dipulihkan, sejauh mana produk atau sistem dapat memulihkan data yang terpengaruh secara langsung dan mengatur ulang kondisi sistem sesuai keinginan ketika terjadi gangguan.

- 5) Keamanan, sejauh mana suatu produk atau sistem melindungi informasi dan data sehingga seseorang atau sistem lain dapat mengakses data sesuai dengan jenis dan tingkat otorisasi yang dimilikinya. Ciri-ciri tersebut terbagi menjadi beberapa ciri yaitu:
- a. Kerahasiaan, sejauh mana produk atau perangkat lunak memastikan data hanya dapat diakses oleh mereka yang berwenang untuk memiliki akses.
 - b. Integritas, sejauh mana produk atau perangkat lunak dapat mencegah akses tidak sah untuk mengubah data.
 - c. Non repudiation, sejauh mana peristiwa atau tindakan dapat dibuktikan telah terjadi, sehingga tidak ada penolakan atas peristiwa atau tindakan tersebut.
 - d. Akuntabilitas, sejauh mana tindakan suatu entitas dapat dilacak secara unik ke entitas tersebut.
 - e. Keaslian, sejauh mana identitas subjek atau sumber dapat dibuktikan menjadi yang diklaim.
- 6) Portabilitas, sejauh mana efektivitas dan efisiensi suatu sistem, produk atau komponen dapat dipindahkan dari satu perangkat keras, perangkat lunak atau digunakan dalam lingkungan yang berbeda. Ciri-ciri tersebut terbagi menjadi beberapa ciri yaitu:
- a. Adaptabilitas, sejauh mana produk atau sistem dapat secara efektif dan efisien diadaptasi ke perangkat lunak, perangkat keras, dan lingkungan yang berbeda.
 - b. Penginstalan, sejauh mana produk atau sistem berhasil diinstal atau dihapus di lingkungan tertentu.
 - c. Dapat diganti yaitu sejauh mana suatu produk atau sistem dapat menggantikan produk atau sistem lain yang ditentukan untuk tujuan yang sama di lingkungan yang sama.

- 7) Kinerja terhadap sejumlah sumber daya yang digunakan dalam kondisi tertentu, dalam hal ini efisiensi kinerja dapat memberikan reaksi dan waktu yang diperlukan ketika melakukan suatu tindakan dari suatu fungsi dan perangkat lunak dapat menggunakan jumlah sumber daya saat melakukan tindakan dari suatu fungsi. Kinerja relatif terhadap sumber daya yang digunakan dalam kondisi tertentu. Ciri ini terbagi menjadi beberapa sub ciri yaitu :
- a. Perilaku waktu, sejauh mana respon dan waktu pemrosesan produk atau sistem dapat memenuhi persyaratan saat menjalankan suatu fungsi.
 - b. Pemanfaatan sumber daya, sejauh mana jumlah dan jenis sumber daya yang digunakan oleh suatu produk atau sistem dapat memenuhi persyaratan saat menjalankan fungsinya.
 - c. Kapasitas, sejauh mana batas maksimum produk atau parameter sistem dapat memenuhi persyaratan.
- 8) Pemeliharaan adalah tingkat dimana perangkat lunak dapat dimodifikasi. Dalam hal ini, modifikasi adalah perbaikan, perubahan atau penyesuaian pada perangkat lunak untuk dapat mengubah lingkungan, persyaratan dan fungsionalitas tertentu. Selain itu, perangkat lunak dapat dianalisis untuk mengetahui penyebab kegagalan perangkat lunak dalam mengidentifikasi bagian-bagian yang dapat dimodifikasi. Sejauh mana efektivitas dan efisiensi suatu produk atau sistem dapat dicapai diobati. Ciri ini terbagi menjadi beberapa sub ciri yaitu.
- a. Modularitas, sejauh mana sistem terdiri dari komponen yang terpisah sehingga perubahan atau modifikasi salah satu komponen tersebut berdampak kecil pada komponen lainnya.
 - b. Dapat digunakan kembali, sejauh mana aset dapat digunakan lebih banyak oleh satu sistem atau digunakan untuk membangun aset lain.

- c. Analisis, tingkat efektivitas dan efisiensi untuk menilai dampak perubahan pada satu atau lebih bagian dari suatu produk atau sistem, untuk mendiagnosis kekurangan atau penyebab kegagalan produk, untuk mengidentifikasi bagian yang akan diubah.
- d. Dapat dimodifikasi sejauh mana suatu produk atau sistem dapat dimodifikasi secara efektif dan efisien tanpa mengurangi kualitas produk yang ada.
- e. Testabilitas, tingkat keefektifan dan efisiensi untuk membentuk kriteria pengujian suatu produk, sistem atau komponen dan pengujian dapat dilakukan untuk mengetahui apakah kriteria tersebut telah terpenuhi.

1.9. 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, 2017). Skala Likert umumnya menggunakan lima titik dengan label netral pada posisi tengah (ketiga). Skala Likert dapat dilihat pada Tabel 2.5

Gambar 2. 5 Skala Likert

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

Sumber : (Sugiyono, 2017)

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

$$Persentase = \frac{Skor\ Aktual\ (f)}{Skor\ Ideal\ (n)} \times 100\% \dots\dots\dots(2)$$

Persentase kelayakan yang diperoleh kemudian dibandingkan dengan Tabel konversi yang berpedoman pada acuan konversi nilai, dapat dilihat pada Tabel 2.6.

Gambar 2. 6 Skala Konversi Nilai

Persentase Pencapaian (%)	Interpretasi
$\geq 90\%$ - 100%	Sangat Baik
$\geq 70\%$ - $<90\%$	Baik
$\geq 70\%$ - $<80\%$	Cukup Baik
$\geq 60\%$ - $<70\%$	Kurang Baik
0% - 60%	Tidak Baik

Sumber : (Sugiyono, 2017)