

BAB II

LANDASAN TEORI

1.1. Penelitian Terdahulu

Berikut ini daftar penelitian terdahulu untuk mendukung penelitian yang dilakukan penulis:

Budiman dan Nugraha, (2019) melakukan penelitian dengan judul Aplikasi Raport *Online* Berbasis Web Menggunakan Framework CodeIgniter dengan studi kasus di SMK Angkasa 1 Margahayu. Permasalahan yang sering terjadi dalam pengolahan nilai di SMK Angkasa 1 Margahayu saat ini yaitu setiap guru mata pelajaran tidak bisa melihat nilai rapor dan juga orang tua siswa yang berhalangan hadir pada saat penyerahan rapor tidak akan bisa melihat hasil belajar anaknya serta penyimpanan data yang belum terorganisir. Penelitian ini bertujuan untuk membangun suatu aplikasi nilai yang mempermudah pengecekan, pencatatan dan laporan data nilai siswa yang terkomputerisasi. Metodologi penelitian yang digunakan yaitu metode deskriptif. Metode pengembangan sistem yang digunakan adalah dengan menggunakan metode OOSE, bahasa pemrograman yang digunakan adalah bahasa PHP dengan *framework* CodeIgniter. Penelitian ini telah menghasilkan sebuah aplikasi rapor online untuk mengolah nilai yang membantu kerja dari para guru dan wali kelas dan dapat mempermudah pengguna untuk melakukan proses pengolahan nilai agar dapat diolah secara efektif dan efisien sehingga bisa langsung diakses serta informasinya dapat tersampaikan dengan baik.

Putra dan Ariansidi, (2019) melakukan penelitian dengan judul Rancang Bangun Sistem Informasi Pengolahan Nilai Rapor Berbasis Web Pada SMK Negeri 1 Kotaraja. Proses perekapan nilai pada SMK Negeri 1 Kotaraja masih

menggunakan cara konvensional dalam pengolahan nilai yaitu, dengan mengandalkan formulir data nilai siswa yang diberikan oleh guru mata pelajaran yang masih ada kemungkinan formulir tersebut hilang, terselip, tercecer dan rentan terjadi kesalahan. Metode yang dipakai dalam pengumpulan dan pendukung penelitian ini yaitu dengan metodologi pengembangan perangkat lunak waterfall dengan melakukan observasi, wawancara, dokumentasi, dan kepustakaan. Penelitian ini berawal dari tahap mengidentifikasi masalah, kemudian mengumpulkan data dan mengelolanya, tahap berikutnya adalah menganalisis sistem dan merancang sistem, selanjutnya melakukan implementasi berdasarkan rancangan dari sistem yang dibuat, sistem ini dirancang dengan menggunakan bahasa pemrograman PHP dan MySQL sebagai databasenya, setelah sistem dibuat maka akan dilakukan pengujian sistem menggunakan *blackbox testing*.

Juliantri, Florentinus dan Wibawanto, (2017) melakukan penelitian dengan judul Pengembangan *E-Rapor* Kurikulum 2013 Berbasis Web di SMK Negeri 1 Slawi. SMK Negeri 1 Slawi merupakan sekolah rujukan kurikulum 2013 dan mendapatkan penghargaan salah satu SMK yang Indeks Integritas Ujian Nasional (IIUN) Tertinggi se Indonesia. Metode penilaian yang autentik menjadikan format penulisan rapor pada kurikulum 2013 menjadi lebih kompleks, penelitian ini dimaksudkan untuk mengembangkan model rapor berbasis Web *e-Rapor*, serta untuk mengetahui tingkat validitas, kepraktisan dan keefektifan dari Sistem yang dikembangkan. Metode yang digunakan adalah *Research & Development* (R&D) dan model pengembangan sistem *waterfall*, Hasil uji perangkat lunak e-rapor telah siap digunakan. Sesuai dengan validitas *e-rapor* dinyatakan valid dengan nilai 83,08% dengan kategori sangat baik. Tingkat kepraktisan dapat dilihat dari

indikator keterlaksanaan dengan skor 81,83% (sangat terlaksana). Tingkat keefektifan dilihat dari respon guru dengan skor 84,64% (sangat baik) sehingga dapat disimpulkan bahwa hasil pengembangan model *e-rapor* berbasis web adalah valid, praktis dan efektif.

Gunawan dan Triantoro, (2017) melakukan penelitian dengan judul Sistem Informasi Pengolahan Rapor Kurikulum 2013. Sistem pengolahan rapor yang berjalan di SMKN 2 Purwokerto masih banyak kekurangan dan kendala. Selama ini hanya ada beberapa orang saja yang bisa mengisi nilai rapor menggunakan sistem yang ada, dikarenakan seorang guru merasa takut terjadi kesalahan ketika pengisian nilai. Kesalahan seperti menghapus informasi data siswa, menjadikan pekerjaan yang membutuhkan waktu sehari-hari dalam pengisian rapor seluruh siswa di SMKN 2 Purwokerto. Metodologi yang digunakan dalam pengembangan sistem ini adalah pertama menggunakan teknik wawancara untuk mendapatkan data yang diperlukan dengan cara melakukan komunikasi dengan pelanggan dan kasir serta manajemen Orion Sport Center. Kedua studi kepustakaan untuk mengumpulkan data yang dilakukan dengan cara membaca buku, makalah dan hal lain yang dapat menunjang penelitian serta data-data yang diperoleh dari Orion Sport Center Purwokerto. Ketiga pengembangan sistem dikembangkan dengan metode *Waterfall*. Standar pengujian yang digunakan dalam penelitian ini yaitu ISO/IEC 25010.

Nursahid, Riasti dan Purnama, (2015) melakukan penelitian dengan judul Pembangunan Sistem Informasi Penilaian Hasil Belajar Siswa Sekolah Menengah Atas (SMA) Negeri 2 Rembang Berbasis Web. Sistem Informasi Penilaian Hasil Belajar Sekolah Menengah Atas (SMA) Negeri 2 Rembang saat ini masih

konvensional dengan mencatat pada buku daftar nilai kemudian direkap hasilnya secara manual. Mulai dari nilai kognitif yang meliputi nilai ulangan harian, ulangan tengah semester, ulangan semester, nilai psikomotorik, nilai afektif yang diolah kemudian hasilnya ditulis di leger secara manual. Leger tersebut kemudian menjadi acuan dalam mengisi rapor yang juga manual dalam pengisiannya. Hal ini menyebabkan proses memakan waktu lama dan tingkat kesalahan yang besar, selain itu juga tingkat kecepatan akses data (laporannya) jika dibutuhkan sewaktu-waktu dibutuhkan menjadi terlambat. Tujuan dari penelitian ini yaitu pembuatan sistem informasi yang dapat digunakan untuk penilaian hasil belajar. Metode yang digunakan dalam penelitian ini adalah observasi, wawancara, kepustakaan, analisis, perencanaan, perancangan atau desain, pembangunan, uji coba sistem serta implementasi sistem. Sistem ini dikembangkan dengan menggunakan bahasa pemrograman PHP dan *database* MySQL.

1.2. Nilai

Nilai rapor (hasil belajar) merupakan kumpulan nilai akhir dari semua mata pelajaran yang ditempuh siswa pada suatu semester tahun ajaran tertentu (Nursahid, Riasti dan Purnama, 2015). Penilaian adalah proses pengumpulan dan pengolahan informasi untuk mengukur pencapaian hasil belajar peserta didik (Kemendikbud, 2016).

Nilai hasil belajar peserta didik pada pendidikan dasar dan pendidikan menengah meliputi aspek sikap, pengetahuan dan keterampilan.

1. Penilaian sikap merupakan kegiatan yang dilakukan oleh pendidik untuk memperoleh informasi deskriptif mengenai perilaku peserta didik. Penilaian

ini dilakukan melalui observasi/pengamatan dan teknik penilaian lain yang relevan, dan pelaporannya menjadi tanggungjawab wali kelas atau guru kelas.

2. Penilaian pengetahuan merupakan kegiatan yang dilakukan untuk mengukur penguasaan pengetahuan peserta didik. Penilaian ini dilakukan melalui tes tertulis, tes lisan, dan penugasan sesuai dengan kompetensi yang dinilai.
3. Penilaian keterampilan merupakan kegiatan yang dilakukan untuk mengukur kemampuan peserta didik menerapkan pengetahuan dalam melakukan tugas tertentu. Penilaian ini dilakukan melalui praktik, produk, proyek, portofolio, dan/atau teknik lain sesuai dengan kompetensi yang dinilai.

Penilaian hasil belajar oleh pendidik bertujuan untuk memantau dan mengevaluasi proses, kemajuan belajar, dan perbaikan hasil belajar peserta didik secara berkesinambungan.

1.3. Hasil Belajar

Hasil belajar secara umum dipandang sebagai perwujudan nilai-nilai yang diperoleh siswa melalui proses belajar mengajar. Hal ini menunjukkan bahwa hasil belajar adalah penguasaan hubungan-hubungan antara bagian-bagian informasi yang telah diperoleh sehingga siswa dapat menampilkan pengalaman dan penguasaan bahan pelajaran yang dipelajari (Lubis, 2012).

1.4. Kurikulum 2013

Pada tahun pelajaran 2014/2015 telah mulai diberlakukan Kurikulum 2013 di seluruh Indonesia yang merupakan pembaharuan dan penyempurnaan Kurikulum 2006. Karakteristik dasar Kurikulum 2013 adalah terletak pada pendekatan yang digunakan dalam pengembangan kurikulum tersebut. Kurikulum 2013 menekankan pendekatan saintifik pada jenjang pendidikan dasar hingga menengah. Implementasi memiliki tujuan untuk meningkatkan kualitas sumber daya manusia dan meningkatkan daya saing bangsa seiring dengan perkembangan ilmu pengetahuan, teknologi, dan seni (Setiadi, 2016).

Salah satu aspek yang mengalami perkembangan dibanding kurikulum sebelumnya adalah penilaian. Pada Kurikulum 2013, Standar Penilaian Pendidikan adalah kriteria mengenai lingkup, tujuan, manfaat, prinsip, mekanisme, prosedur, dan instrumen penilaian hasil belajar peserta didik yang digunakan sebagai dasar dalam penilaian hasil belajar peserta didik pada pendidikan dasar dan pendidikan menengah. Penilaian adalah proses pengumpulan dan pengolahan informasi untuk mengukur pencapaian hasil belajar peserta didik. Pembelajaran adalah proses interaksi antar peserta didik, antara peserta didik dengan pendidik dan sumber belajar pada suatu lingkungan belajar (Kemendikbud, 2016).

Pada Kurikulum 2013, penilaian diatur dalam Permendikbud Nomor 66 Tahun 2013 tentang Standar Penilaian Pendidikan meliputi penilaian otentik, penilaian diri, penilaian berbasis portofolio, ulangan harian, ulangan tengah semester, ulangan akhir, ujian tingkat kompetensi, ujian mutu tingkat kompetensi, ujian nasional dan ujian sekolah/madrasah. Penilaian ini merupakan penilaian hasil belajar yang dilakukan oleh pendidik, satuan pendidikan dan pemerintah. Pada

Kurikulum 2013, penilaian lebih tegas dan menyeluruh dibanding dengan pelaksanaan penilaian pada Kurikulum 2006. Pelaksanaan penilaian pada Kurikulum 2013 secara eksplisit meminta agar guru-guru di sekolah seimbang dalam melakukan penilaian di tiga ranah domain, yaitu kognitif, afektif dan psikomotor sesuai dengan tujuannya yang hendak diukur. Penekanan penilaian menyeluruh terhadap ketiga aspek memberikan perubahan besar dibanding kurikulum sebelumnya (Setiadi, 2016)..

1.5. Sistem

Sistem dapat didefinisikan sebagai kumpulan komponen-komponen yang bekerjasama untuk tujuan bersama. Misalnya bidang organisasi keuangan, operasi dan pemasaran memiliki tujuan yang sama untuk mencapai tujuan perusahaan secara keseluruhan. Hal ini dapat dilihat bahwa dalam sistem, data digunakan sebagai *input* untuk diproses yang menghasilkan informasi sebagai *output* (Irawan, 2018). Berikut ini merupakan karakteristik sistem:

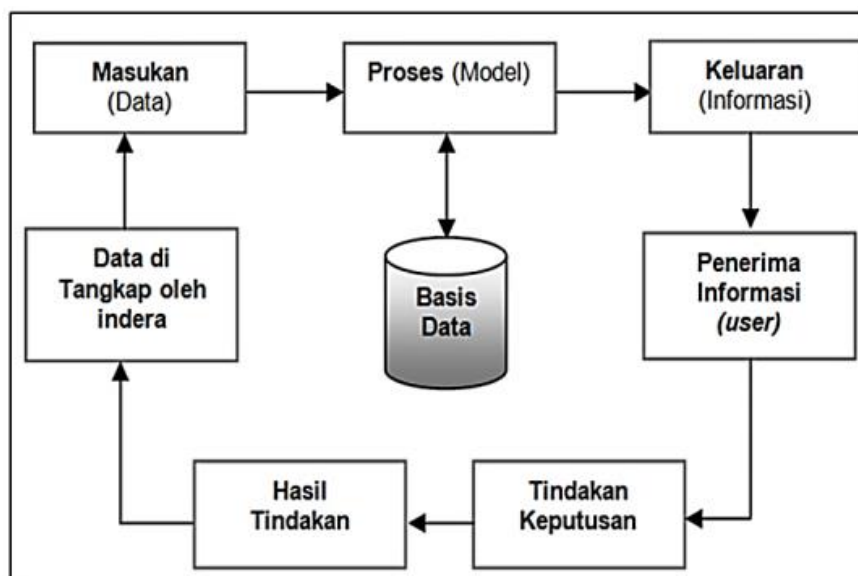
1. Komponen sistem (*Components*)
2. Batasan Sistem (*Boundary*)
3. Lingkungan Luar Sistem (*Environment*)
4. Penghubung Sistem (*Interface*)
5. Masukan Sistem (*Input*)
6. Keluaran Sistem (*Output*)
7. Pengolah Sistem (*Process*)
8. Sasaran Sistem (*Objective*)

1.6. Perancangan Sistem

Perancangan sistem merupakan tahapan dalam membangun sebuah sistem setelah tahap analisis sistem dan siklus pengembangan sistem. Tahapan ini mendefinisikan kebutuhan-kebutuhan fungsional dan menggambarkan suatu sistem yang akan dibangun. Dalam merancang suatu sistem dapat menggunakan pemodelan secara terstruktur dengan menggunakan grafik atau diagram (Andalia and Setiawan, 2015). Rancangan sistem dibuat untuk mempermudah dalam implementasi sistem (Monica, Borman and Satya, 2017).

1.7. Informasi

Sebuah informasi dapat didefinisikan sebagai suatu hasil dari pengolahan data dalam suatu bentuk yang lebih berguna dan lebih berarti bagi penerimanya yang menggambarkan suatu kejadian-kejadian yang nyata yang digunakan untuk pengambilan keputusan (Herliana and Rasyid, 2016). Informasi adalah data yang telah diproses dengan suatu cara untuk memberikan arti dan memperbaiki pengambilan keputusan (Mara and Adrian, 2017). Berikut gambar siklus informasi:



Gambar 2.1 Siklus Informasi (Mariana, Ayu and Susanto, 2019)

Kualitas informasi bergantung pada lima hal (Andalia and Setiawan, 2015)

yaitu sebagai berikut:

1. Akurat

Sebuah informasi harus akurat karena dari sumber informasi hingga penerima informasi kemungkinan banyak terjadi gangguan yang dapat mengubah atau merusak informasi tersebut. Informasi dikatakan akurat apabila informasi tersebut tidak menyesatkan, bebas dari kesalahan kesalahan dan harus jelas mencerminkan maksudnya.

2. Tepat Waktu

Informasi yang dihasilkan dari suatu proses pengolahan data, datangnya tidak boleh terlambat (usang). Informasi yang terlambat tidak akan mempunyai nilai yang baik, karena informasi merupakan landasan dalam pengambilan keputusan.

3. Relevan

Informasi dikatakan berkualitas jika relevan bagi pemakainya. Hal ini bahwa informasi tersebut harus bermanfaat bagi pemakainya. Relevansi informasi untuk tiap-tiap orang satu dengan yang lainnya berbeda.

4. Lengkap

Informasi lengkap jika tidak meninggalkan aspek-aspek penting dari kejadian yang merupakan dasar masalah.

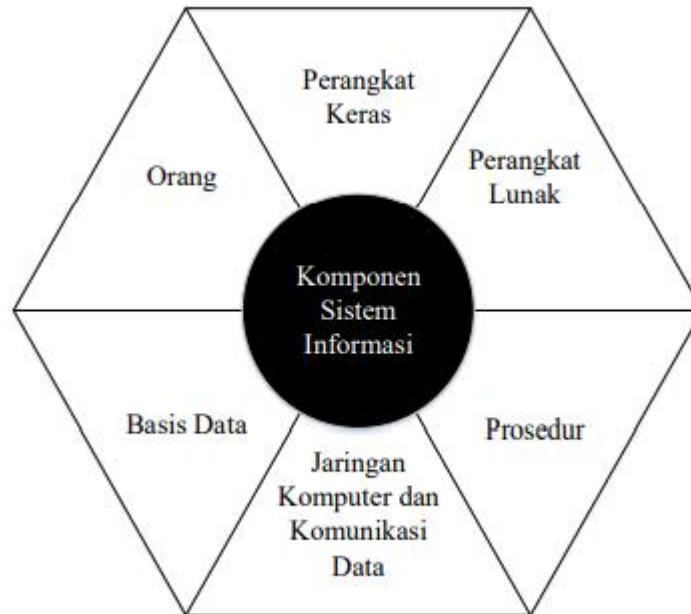
5. Mudah dipahami

Informasi dapat dipahami jika disajikan dalam bentuk yang dapat dipakai dan jelas.

1.8. Komponen Sistem Informasi

Sistem informasi memberikan nilai tambah terhadap proses, produksi, kualitas, manajemen, pengambilan keputusan dan pemecahan masalah serta keunggulan kompetitif yang berguna bagi kegiatan bisnis (Sulistiani and Damayanti, 2017). Adapun komponen dari sistem informasi adalah:

1. Perangkat keras (*hardware*): mencakup peranti-peranti fisik seperti komputer dan printer.
2. Perangkat lunak (*software*) atau program: sekumpulan instruksi yang memungkinkan perangkat keras untuk dapat memproses data.
3. Prosedur (*procedure*): sekumpulan aturan yang dipakai untuk mewujudkan pemrosesan data dan pembangkitan keluaran yang dikehendaki.
4. Orang (*brainware*): semua pihak yang bertanggung jawab dalam pengembangan sistem informasi, pemrosesan dan penggunaan keluaran sistem informasi.
5. Basis data (*database*): sekumpulan tabel, hubungan dan lain-lain yang berkaitan dengan penyimpanan data.
6. Jaringan komputer dan komunikasi data: sistem penghubung yang memungkinkan sumber (*resources*) dipakai secara bersama atau diakses oleh sejumlah pemakai.



Gambar 2.2 Komponen Sistem Informasi (Swara, Kom dan Pebriadi, 2016)

1.9. Kurikulum

Kurikulum adalah seperangkat rencana dan pengaturan mengenai tujuan, isi, dan bahan pelajaran serta cara yang digunakan sebagai pedoman penyelenggaraan kegiatan pembelajaran untuk mencapai tujuan pendidikan tertentu. Kurikulum adalah sejumlah bidang studi yang harus ditempuh oleh siswa dari awal sampai akhir program pembelajaran untuk memperoleh ijazah (Ismail *et al.*, 2019).

1.10. Web

Web atau situs adalah sistem dimana informasi dalam bentuk teks, suara, gambar dan lain-lain yang disimpan di *server-server* yang terdapat di seluruh dunia (Bakhri, 2019). *Web page* adalah sebuah halaman yang ditampilkan dalam sebuah *website*. Halaman *markup* dalam sebuah web disebut HTML (*Hyper Text Markup Language*) yang dapat diakses melalui HTTP (*Hyper Text Transfer Protocol*). Ada dua jenis web yaitu web statis dan web dinamis. *Hypertext Markup Language*

(HTML) merupakan bahasa standar yang digunakan oleh *browser* untuk membuat halaman dan dokumen pada sebuah Web yang kemudian dapat diakses dan dibaca layaknya sebuah artikel. HTML saat ini merupakan standar Internet yang didefinisikan dan dikendalikan penggunaannya oleh *World Wide Web Consortium*.

HTTP singkatan dari *Hypertext Transfer Protocol* adalah suatu protokol yang digunakan untuk mengirim dokumen atau halaman dalam WWW atau World Wide Web. Sedangkan pengertian HTTP menurut kamus besar adalah protokol jaringan untuk didistribusikan, kolaboratif, sistem informasi *hypermedia*. HTTP adalah dasar dari komunikasi data untuk WWW. WWW merupakan kumpulan *server* web dari seluruh dunia yang mempunyai kegunaan untuk menyediakan data dan informasi untuk dapat digunakan bersama. Informasi informasi tersebut tidak hanya berupa *text*, dapat juga berupa gambar, suara, dan video.

1.11. MySQL

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (bahasa Inggris: *database management sistem*) atau DBMS yang *multithread*, *multi-user*, dengan sekitar 6 juta instalasi di seluruh dunia (Standisyah and N.S, 2015).

MySQL adalah RDBMS (*Relational Database Management Sistem*) yang didistribusikan secara gratis dibawah lisensi GPL. Dimana setiap orang bebas untuk menggunakan MySQL, namun tidak boleh dijadikan produk turunan yang bersifat komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam *database* sejak lama, yaitu SQL (*Structured Query Language*). SQL adalah sebuah konsep pengoperasian *database*, terutama untuk pemilihan atau seleksi dan

pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis. Keandalan suatu sistem *database* (DBMS) dapat diketahui dari cara kerja *optimizer*-nya dalam melakukan proses perintah-perintah SQL, yang dibuat oleh user maupun program-program aplikasinya. Sebagai *database server*, MySQL dapat dikatakan lebih unggul dibandingkan *database server* lainnya dalam *query* data. Hal ini terbukti untuk *query* yang dilakukan oleh *single user*, kecepatan *query* MySQL bisa sepuluh kali lebih cepat dari PostgreSQL dan lima kali lebih cepat dibandingkan Interbase.

1.12. Framework CodeIgniter

Framework dapat diartikan sebagai alat yang digunakan untuk membantu dan memudahkan pembuatan aplikasi dalam konteks ini adalah aplikasi Web. *Framework* juga dapat didefinisikan sebagai kumpulan *script* (terutama *class* dan *function*) yang dapat membantu *developer* menangani berbagai masalah seperti koneksi ke *database*, pemanggilan *variable* dan fungsi-fungsi lainnya sehingga *developer* dapat lebih fokus dan lebih cepat membangun aplikasi (Fernando, Tanaamah and Wijaya, 2017).

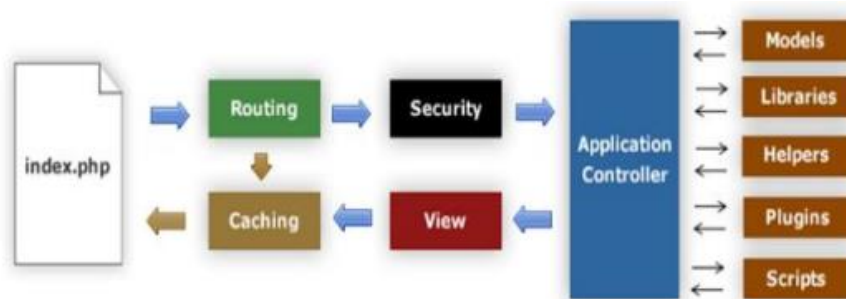
Codeigniter adalah *framework* web untuk bahasa pemrograman PHP, yang dibuat oleh Rick Ellis pada tahun 2006, penemu dan pendiri EllisLab yaitu suatu tim kerja yang berdiri pada tahun 2002 yang bergerak di bidang pembuatan *software* dan *tool* untuk para pengembang web (Romadhon and Desmulyati, 2019).

Codeigniter memiliki banyak fitur yang membantu para pengembang PHP untuk dapat membuat aplikasi web secara mudah dan cepat. *Codeigniter*

mengizinkan para pengembang untuk menggunakan *framework* secara parsial maupun secara keseluruhan. Adapun keunggulan dari *Codeigniter* sebagai berikut:

1. *Codeigniter* adalah *framework* yang bersifat *free* dan *open-source*.
2. *Codeigniter* memiliki ukuran kecil dibandingkan dengan *framework* lain.
3. Aplikasi yang dibuat menggunakan *Codeigniter* bisa berjalan cepat.
4. *Codeigniter* menggunakan pola design *Model-View-Controller* (MVC) sehingga satu file tidak terlalu berisi banyak kode.
5. *Codeigniter* dapat diperluas sesuai dengan kebutuhan.
6. *Codeigniter* terdokumentasi dengan baik. Informasi tentang pustaka kelas dan fungsi yang disediakan oleh *Codeigniter* dapat diproses melalui dokumentasi yang disertai di dalam paket distribusinya.

Alur dari aplikasi yang ditulis menggunakan *Codeigniter* digambarkan seperti dibawah ini:

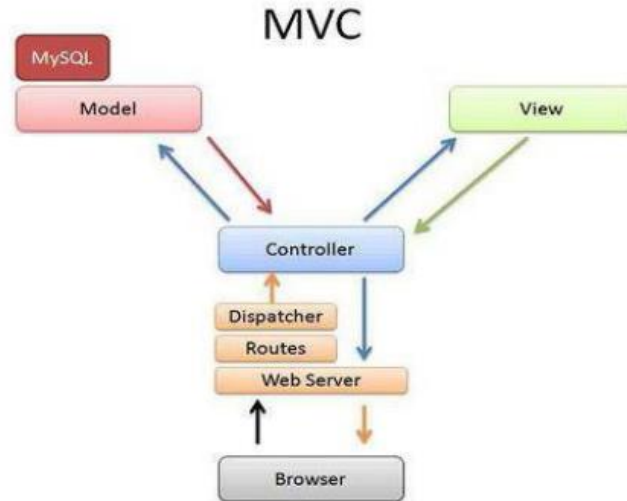


Gambar 2.3 Arsitektur *Codeigniter* (Romadhon and Desmulyati, 2019)

Konsep MVC:

Codeigniter merupakan konsep M-V-C (Model-View-Controller) yang memungkinkan pemisahan antara *layer application-logic* dan *presentation*. Dengan konsep ini kode PHP, query Mysql, Javascript dan CSS dapat saling dipisahkan sehingga ukuran file menjadi lebih kecil dan lebih mudah dalam perbaikan

kedepannya atau *maintenance* (Mara and Adrian, 2017). Interaksi ketiga bagian objek yang disebutkan, dapat digambarkan berikut:



Gambar 2.4 Model View Controller (Letsoin, 2013)

1.13. *Unified Modeling Language* (UML)

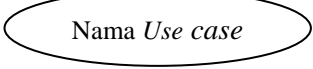
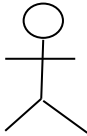

Rosa dan Shalahuddin, (2019) menjelaskan bahwa *Unified Modeling Language* (UML) merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung. UML merupakan suatu kumpulan teknik terbaik yang telah terbukti sukses dalam memodelkan sistem yang besar dan kompleks. UML tidak hanya digunakan dalam proses pemodelan perangkat lunak, namun hampir dalam semua bidang yang membutuhkan pemodelan. UML menurut Rosa dan Shalahuddin (2019), UML adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan requirement, membuat analisis dan desain.

1. *Use case Diagram*

Menurut Rosa dan Shalahuddin (2019), *use case* diagram merupakan permodelan untuk tingkah laku sistem informasi yang akan dibuat. *Use case*



mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. *Use case* juga dapat digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem dan siapa saja yang berhak atas perintah-perintah yang ada pada sistem tersebut. *Use case* harus dengan penamaan yang singkat dan mudah dimengerti. Simbol dan deskripsi *use case* diagram dapat dilihat pada tabel 2.1 berikut:

Tabel 2. 1 Simbol dan Deskripsi Use Case Diagram

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</p>
<p>Aktor/<i>Actor</i></p> 	<p>Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informaasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang biasanya dinyatakan menggunakan kata benda diawal frase nama <i>actor</i>.</p>
<p><i>Extend</i> /Ekstensi</p> <p><<<i>extend</i>>></p> 	<p>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</i></p>

Sumber: Rosa dan Shalahuddin, (2019)

Tabel 2. 1 Simbol dan Deskripsi Use Case Diagram (Lanjutan)

	<p><i>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.</p>
<p><i>Generalization/Generalisasi</i></p> 	<p>Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.</p>
<p><i>Association/Asosiasi</i></p> 	<p>Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi interaksi dengan <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.</p>

Sumber: Rosa dan Shalahuddin, (2019)

2. Class Diagram

Rosa dan Shalahuddin, (2019) menjelaskan bahwa 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. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas. Diagram kelas dibuat agar pembuat program membuat kelas-kelas sesuai rancangan didalam diagram kelas agar antara dokumentasi perancangan dan perangkat lunak

sinkron. Kelas-kelas yang ada pada struktur sistem harus dapat melakukan fungsi-fungsi sesuai dengan kebutuhan sistem sehingga pembuat perangkat lunak dapat membuat kelas-kelas didalam program perangkat lunak sesuai dengan perancangan diagram kelas. Susunan struktur kelas yang baik pada diagram kelas sebaiknya memiliki jenis-jenis kelas sebagai berikut:

1. Kelas Main

Kelas main adalah kelas yang memiliki fungsi awal dieksekusi ketika sistem dijalankan.

2. Kelas yang menangani tampilan sistem (*view*)

Kelas yang mendefinisikan dan mengatur tampilan ke pemakai.

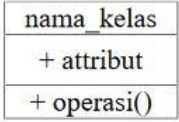



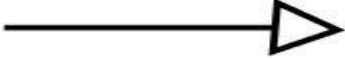


3. Kelas yang diambil dari pendefinisian *use case* (*controller*)

Kelas yang menangani fungsi-fungsi yang harus ada diambil dari pendefinisian *use case*, kelas ini biasanya disebut dengan kelas proses yang menangani proses bisnis pada perangkat lunak.

4. Kelas yang diambil dari pendefinisian data (*model*)

Kelas yang digunakan untuk memegang data menjadi sebuah kesatuan yang diambil maupun akan disimpan ke basis data. Semua tabel yang dibuat di basis data dapat dijadikan kelas, namun untuk tabel dari hasil relasi atau atribut multivalued pada ERD dapat dijadikan kelas tersendiri dapat juga tidak asalkan pengaksesanya dapat dipertanggungjawabkan. Simbol dan deskripsi *class* diagram dapat dilihat pada tabel 2.2 berikut:

Tabel 2. 2 Simbol dan Deskripsi *Class Diagram*



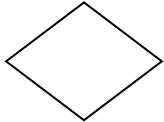
Simbol	Deskripsi
<p>Kelas</p> 	<p>Kelas pada struktur sistem.</p>
<p>Antarmuka / <i>Interface</i></p> 	<p>Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.</p>
<p>Asosiasi / <i>Association</i></p> 	<p>Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan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

Sumber: Rosa dan Shalahuddin, (2019)



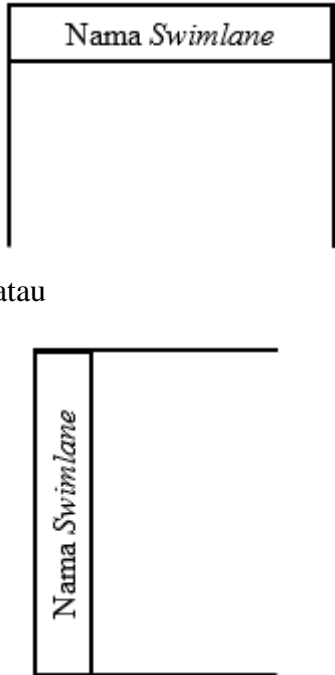
3. Activity Diagram

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak (Rosa and Shalahuddin, 2019). Perlu diperhatikan bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. Adapun fungsi dari *activity* diagram adalah menggambarkan proses bisnis dan urutan aktifitas dalam sebuah proses, memperlihatkan urutan aktifitas proses pada sistem, *activity* diagram dibuat berdasarkan beberapa usecase pada *use case* diagram. Simbol dan deskripsi *activity* diagram dapat dilihat pada tabel 2.3 berikut:

Tabel 2. 3 Simbol dan Deskripsi 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 pilhan aktivitas lebih dari satu.

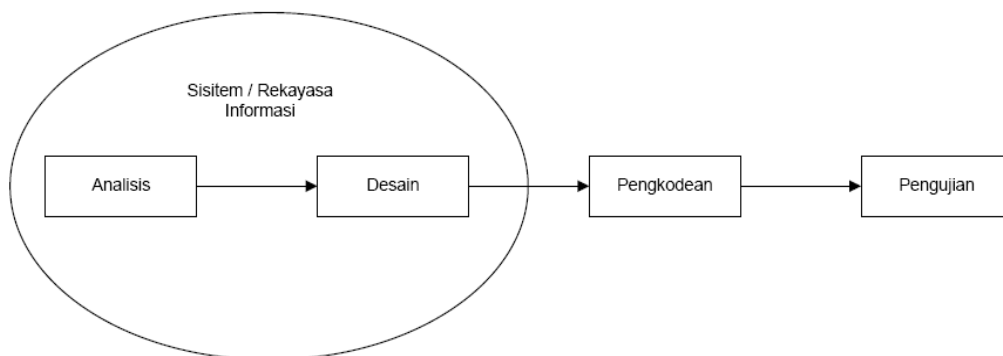
Tabel 2.3 Simbol dan Deskripsi Activity Diagram (Lanjutan)

<p>Penggabungan/<i>Join</i></p> 	<p>Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu</p>
<p>Status akhir</p> 	<p>Status akhir yang di lakukan sistem, sebuah diagram aktivitas pasti memiliki status akhir</p>
 <p>atau</p>	

Sumber: Rosa dan Shalahuddin, (2019)

1.14. Waterfall

Model SDLC air terjun (*waterfall*) sering 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 dimulai dari analisis, desain, pengkodean, pengujian dan tahap pendukung (*support*). Berikut ini adalah gambar model air terjun:



Gambar 2.5 Tahap Model Waterfall (Rosa and Shalahuddin, 2019)

Tahap dalam siklus pengembangan sistem menggunakan model waterfall sebagai berikut:

1. Analisis kebutuhan perangkat lunak

Proses pengumpulan kebutuhan dilakukan secara intensif untuk memesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh *user*. Spesifikasi kebutuhan perangkat lunak pada tahap ini perlu untuk di dokumentasikan.

2. Desain

Desain perangkat lunak adalah proses multilangkah yang fokus pada desain pembuatan program termasuk struktur data, aritektur perangkat lunak, representasi antarmuka, dan prosedur pengkodean. Tahap ini mentranslasi kebutuhan perangkat lunak dari tahap analisa kebutuhan ke representasi

desain agar dapat diimplementasikan mejadi program pada tahap selanjutnya. Desain perangkat lunak yang dihasilkan pada tahap ini juga perlu di dokumentasikan.

3. Pembuatan kode program

Desain harus ditranslasikan ke dalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain.

4. Pengujian

Pengujian fokus pada perangkat lunak secara dari segi logik dan fungsional dan memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.

5. Pendukung (*support*) atau pemeliharaan (*maintenance*)

Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirimkan ke *user*. Perubahan bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan baru. Tahap pendukung atau pemeliharaan dapat mengurangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan perangkat lunak yang sudah ada, tapi tidak untuk membuat perangkat lunak baru.

1.15. Pengujian Sistem

Pengujian adalah suatu set aktifitas yang direncanakan dan sistematis untuk menguji atau mengevaluasi kebenaran yang diinginkan (Rosa and Shalahuddin, 2013). Aktifitas pengujian terdiri dari satu set atau sekumpulan langkah dimana

dapat menempatkan desain kasus uji yang spesifik dan metode pengujian. Secara umum pola pengujian perangkat lunak adalah sebagai berikut:

1. Pengujian dimulai dari level komponen hingga integrasi antarkomponen menjadi sebuah sistem.
2. Teknik pengujian berbeda-beda sesuai dengan berbagai sisi atau unit uji dalam waktu yang berbeda-beda pula bergantung pada pengujian pada bagian mana yang dibutuhkan.
3. Pengujian dilakukan oleh pengembang perangkat lunak, dan jika untuk proyek besar, pengujian bisa dilakukan oleh tim uji yang tidak terkait dengan tim pengembang perangkat lunak (*independent test group* (ITG)).
4. Pengujian dan penirkutan (*debugging*) merupakan aktifitas yang berbeda, tapi penirkutan (*debugging*) harus diakomodasi pada berbagai strategi pengujian. Pengujian lebih fokus untuk mencari adanya kesalahan (*error*) baik dari sudut pandang orang secara umum atau dari sudut pandang pengembang tanpa harus menemukan lokasi kesalahan pada kode program. Penirkutan (*debugging*) adalah proses mencari lokasi kesalahan (*error*) pada kode program sehingga dapat segera diperbaiki oleh pembuat program (*programmer*).

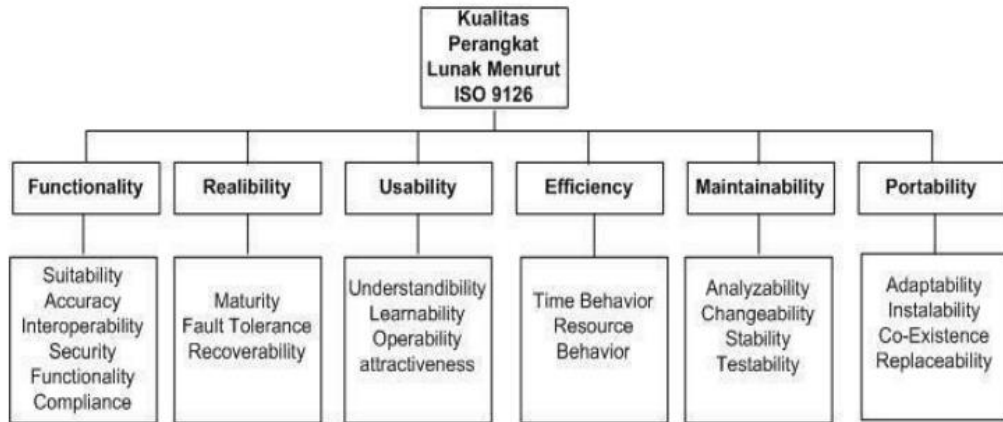
1.16. ISO 9126

Salah satu tolak ukur kualitas perangkat lunak adalah ISO 9126, yang dibuat oleh International Organization for Standardization (ISO) dan International Electrotechnical Commission (IEC). ISO 9126 mendefinisikan kualitas produk perangkat lunak, model, karakteristik mutu, dan metrik terkait yang digunakan untuk mengevaluasi dan menetapkan kualitas sebuah produk software. Standar ISO

9126 telah dikembangkan dalam usaha untuk mengidentifikasi atribut-atribut kunci kualitas untuk perangkat lunak komputer. Menurut (Abran dkk., 2008), ISO 9126 adalah standar internasional yang diterbitkan oleh ISO untuk evaluasi kualitas perangkat lunak dan merupakan pengembangan dari ISO 9001. Faktor kualitas menurut ISO 9126 meliputi enam karakteristik kualitas sebagai berikut (Al-Qutaish, 2010):

1. *Functionality* (Fungsionalitas). Kemampuan perangkat lunak untuk menyediakan fungsi sesuai kebutuhan pengguna, ketika digunakan dalam kondisi tertentu.
2. *Reliability* (Kehandalan). Kemampuan perangkat lunak untuk mempertahankan tingkat kinerja tertentu, ketika digunakan dalam kondisi tertentu.
3. *Usability* (Kebergunaan). Kemampuan perangkat lunak untuk dipahami, dipelajari, digunakan, dan menarik bagi pengguna, ketika digunakan dalam kondisi tertentu.
4. *Efficiency* (Efisiensi). Kemampuan perangkat lunak untuk memberikan kinerja yang sesuai dan relatif terhadap jumlah sumber daya yang digunakan pada saat keadaan tersebut.
5. *Maintainability* (Pemeliharaan). Kemampuan perangkat lunak untuk dimodifikasi. Modifikasi meliputi koreksi, perbaikan atau adaptasi terhadap perubahan lingkungan, persyaratan, dan spesifikasi fungsional.

6. *Portability* (Portabilitas). Kemampuan perangkat lunak untuk ditransfer dari satu lingkungan ke lingkungan lain.



Gambar 2.6 Model Kualitas Perangkat Lunak Model ISO 9126

(Al-Qutaish, 2010)

Masing-masing karakteristik kualitas perangkat lunak model ISO 9126 dibagi menjadi beberapa sub-karakteristik kualitas. Karakteristik Kualitas Perangkat Lunak Model ISO 9126 dapat dilihat pada tabel 2.4 berikut:

Tabel 2.4 Karakteristik Kualitas Perangkat Lunak Model ISO 9126

Karakteristik	Sub Karakteristik	Deskripsi
<i>Functionality</i>	<i>Suitability</i>	Kemampuan perangkat lunak untuk menyediakan serangkaian fungsi yang sesuai untuk tugas-tugas tertentu dan tujuan pengguna
	<i>Accuracy</i>	Kemampuan perangkat lunak dalam memberikan hasil yang presisi dan benar sesuai dengan kebutuhan
	<i>Security</i>	Kemampuan perangkat lunak untuk mencegah akses yang tidak

**Tabel 2.4 Karakteristik Kualitas Perangkat Lunak Model ISO 9126
(Lanjutan)**

		diinginkan, menghadapi penyusup (hacker) maupun otoritas dalam modifikasi data
	<i>Interoperability</i>	Kemampuan perangkat lunak untuk berinteraksi dengan satu atau lebih sistem tertentu
	<i>Compliance</i>	Kemampuan perangkat lunak dalam memenuhi standar dan kebutuhan sesuai peraturan yang berlaku
<i>Reliability</i>	<i>Maturity</i>	Kemampuan perangkat lunak untuk menghindari kegagalan sebagai akibat dari kesalahan dalam perangkat lunak.
	<i>Fault tolerance</i>	Kemampuan perangkat lunak untuk mempertahankan kinerjanya jika terjadi kesalahan perangkat lunak
	<i>Recoverability</i>	Kemampuan perangkat lunak untuk membangun kembali tingkat kinerja ketika terjadi kegagalan sistem, termasuk data dan koneksi jaringan
<i>Usability</i>	<i>Understandability</i>	Kemampuan perangkat lunak dalam kemudahan untuk dipelajari

**Tabel 2.4 Karakteristik Kualitas Perangkat Lunak Model ISO 9126
(Lanjutan)**

	<i>Learnability</i>	Kemampuan perangkat lunak dalam kemudahan untuk dipelajari
	<i>Operability</i>	Kemampuan perangkat lunak dalam kemudahan untuk dioperasikan
	<i>Attractiveness</i>	Kemampuan perangkat lunak dalam menarik pengguna
<i>Efficiency</i>	<i>Time behavior</i>	Kemampuan perangkat lunak dalam memberikan respon dan waktu pengolahan yang sesuai saat melakukan fungsinya
	<i>Resource behavior</i>	Kemampuan perangkat lunak dalam menggunakan sumber daya yang dimilikinya ketika melakukan fungsi yang di tentukan
<i>Maintainability</i>	<i>Analyzability</i>	Kemampuan perangkat lunak dalam mendiagnosis kekurangan atau penyebab kegagalan
	<i>Changeability</i>	Kemampuan perangkat lunak untuk dimodifikasi tertentu
	<i>Stability</i>	Kemampuan perangkat lunak untuk meminimalkan efek tak terduga dari modifikasi perangkat lunak

**Tabel 2.4 Karakteristik Kualitas Perangkat Lunak Model ISO 9126
(Lanjutan)**

	<i>Testability</i>	Kemampuan perangkat lunak untuk dimodifikasi dan divalidasi perangkat lunak lain
<i>Portability</i>	<i>Adaptability</i>	Kemampuan perangkat lunak untuk diadaptasikan pada lingkungan yang berbeda-beda
	<i>Instalability</i>	Kemampuan perangkat lunak untuk diinstal dalam lingkungan yang berbeda-beda
	<i>Coexistence</i>	Kemampuan perangkat lunak untuk berdampingan dengan perangkat lunak lainnya dalam satu lingkungan dengan berbagi sumber daya
	<i>Replaceability</i>	Kemampuan perangkat lunak untuk digunakan sebagai pengganti perangkat lunak lainnya

Sumber: Al-Qutaish, (2010)

Adapun alasan penggunaan ISO 9126 karena ISO 9126 memiliki enam karakteristik pendukung yang dapat digunakan sebagai acuan dalam menilai maupun memberikan masukan terhadap kualitas perangkat lunak yang akan dibangun yang akan menghasilkan nilai uji yang terukur.

1.17. Skala Likert

Skala Likert adalah pengukuran digunakan untuk mengukur sikap, pendapat, dan persepsi seseorang atau sekelompok orang tentang fenomena sosial (Megawaty dan Putra, 2020). Skala likert mempunyai empat atau lebih butir-butir pertanyaan yang dikombinasikan sehingga membentuk sebuah skor atau nilai yang merepresentasikan sifat individu atau rataan, dari semua butir pertanyaan yang dapat digunakan. Terdapat dua bentuk pertanyaan dalam skala likert, yaitu bentuk pertanyaan positif untuk mengukur skala positif dan bentuk pertanyaan negatif untuk mengukur skala negatif. Pertanyaan positif diberi skor 5, 4, 3, 2 dan 1; sedangkan bentuk pertanyaan negatif diberi skor 1, 2, 3, 4 dan 5 (Pranatawijaya dkk., 2019). Berikut ini merupakan skala penskoran likert:

Tabel 2.5 Skala Penskoran Likert

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