

BAB II LANDASAN TEORI

2.1. Penelitian Terdahulu

Penelitian terdahulu yang digunakan dalam penelitian ini adalah :

Tabel 2.1. Penelitian Terdahulu

Nama Penulis	Judul	Tahun
Budi Hartono, Danang	Sistem Pemesanan dan Pembayaran Menggunakan Teknologi Quick Response Code (QR Code) Berbasis Web pada Kedai Cangkir Gubug	2021
Muhammad Dany Noor Hisyam, Tri Listyorini, Endang Supriyati	Purwarupa Sistem Pemesanan Menu Makanan Dan Minuman Menggunakan Qr-Code Berbasis Web	2022
Yedid Chris Diyanto, Muhammad Gemilang Al-Fatihah, Abdul Kadir Jailani, Herlinda	Perancangan Aplikasi Order Paperless Dengan Teknologi Qrcode Berbasis Web Service Pada Restoran	2023
Ndaru Ruseno, Reza Ahbati	Rancang Bangun Aplikasi Pemesanan Tiket Umrah Menggunakan Algoritma First In First Out (FIFO) Berbasis Website	2022
Ilham Rizky Widiyanto, Wowon Priatna, Hendarman Lubis	Algoritma First in First Out (FIFO) Untuk Perancangan Aplikasi Pemesanan Kaos Sablon	2023

Penelitian yang dilakukan Budi Hartono, Danang (2021) memberikan solusi yang buat adalah mengurangi tugas pramusaji tersebut agar pelayanan dapat

dilakukan secara merata dengan memanfaatkan teknologi QRcode untuk pemesanan menu dengan alat pemindainya adalah smartphone android. Pembayaran pun cukup menggunakan nama atau nomor meja dan dapat dilakukan di kasir atau pelanggan cukup request tagihan di mejanya. Dengan begitu pramusaji tidak perlu melakukan take order satu persatu dan pelanggan lebih mudah melakukan pembayaran..

Penelitian yang dilakukan Muhammad Dany Noor Hisyam, Tri Listyorini, Endang Supriyati (2022) Kombinasi web dan teknologi QR-Code dapat menciptakan sebuah sistem pemesanan menu makanan dan minuman yang dapat membantu tugas pramusaji tersebut agar pelayanan dapat dilakukan secara merata dengan memanfaatkan teknologi QR-Code untuk pemesanan menu dengan alat pemindainya adalah smartphone android. Pembayaran pun cukup menggunakan nama atau nomor meja dan dapat dilakukan di kasir atau pelanggan cukup requesttagihan di mejanya. Dengan begitu pramusaji tidak perlu melakukan take order satu persatu dan pelanggan lebih mudah melakukan pembayaran.

Penelitian yang dilakukan Yedid Chris Diyanto, Muhammad Gemilang Al-Fatihah, Abdul Kadir Jailani, Herlinda (2023) Perancangan Aplikasi Order Paperless Dengan Teknologi Qrcode Berbasis Web Service Pada Restoran tersebut sistem pemesanan yang Paperless telah diimplementasikan dengan QR Code menunjukkan hasil yang baik. Selain itu, sistem ini juga dapat menghemat waktu dan tenaga dalam melakukan verifikasi data pemesanan. Dengan adanya implementasi QR Code pada sistem pemesanan yang Paperless, diharapkan dapat meningkatkan kepuasan pelanggan dan efisiensi dalam proses pemesanan.

Penelitian yang dilakukan Ndaru Ruseno, Reza Ahbati (2022) Rancang Bangun Aplikasi Pemesanan Tiket Umrah Menggunakan Algoritma First In First Out (FIFO) Berbasis Website tersebut Algoritma ini menjelaskan tentang metode mengoptimalkan penentuan proses antrian pendaftaran. Penelitian ini menghasilkan sebuah sistem pendukung keputusan berupa hasil yang berdasarkan kriteria-kriteria yang sudah diproses. Pembuatan aplikasi ini menggunakan aplikasi bahasa pemrograman PHP dan database MySQL.

Penelitian yang dilakukan Ilham Rizky Widiyanto, Wowon Priatna, Hendarman Lubis (2023) Algoritma First in First Out (FIFO) Untuk Perancangan Aplikasi Pemesanan Kaos Sablon tersebut Hasil penelitian ini dapat diterapkan pada algoritma FIFO nya untuk antrian pelanggan dalam pemesanan kaos. Aplikasi pemesanan kaos diuji dengan metode white box dengan menjalankan test case pada empat lolos Semua tes lulus, sehingga Anda dapat menggunakan aplikasi pemesanan berdasarkan algoritma FIFO.

Perbedaan dengan penelitian yang dilakukan yaitu dalam penelitian ini menggunakan Teknologi QR-Code untuk pemesanan dan metode FIFO untuk proses antrian pemesanan pelanggan.

2.2. Tinjauan Pustaka

2.2.1. Sistem

Sistem adalah rangkaian dari dua atau lebih komponen-komponen yang saling berhubungan, yang berinteraksi untuk mencapai suatu tujuan. Sebagian besar sistem terdiri dari subsistem yang lebih kecil yang mendukung sistem yang lebih besar(Steinbart, 2019).

Sistem adalah suatu jaringan prosedur yang dibuat menurut pola yang terpadu untuk melaksanakan kegiatan pokok perusahaan(Makmur, Kadafi and Apria, 2019).

Berdasarkan pengertian diatas dapat disimpulkan sistem adalah kumpulan dari komponen-komponen yang saling berkaitan satu dengan yang lain untuk mencapai tujuan dalam melaksanakan suatu kegiatan pokok perusahaan.

2.2.2. Informasi

Informasi adalah data yang telah diorganisasi dan telah memiliki kegunaan dan manfaat(Agustin, 2019).

Informasi adalah data yang telah dikelola dan diproses untuk memberikan arti dan memperbaiki proses pengambilan keputusan. Sebagaimana perannya, pengguna membuat keputusan yang lebih baik sebagai kuantitas dan kualitas dari peningkatan informasi(Suharni and Sari, 2019)s.

Berdasarkan pengertian diatas dapat disimpulkan bahwa pengertian informasi adalah data yang diolah agar bermanfaat dalam pengambilan keputusan bagi penggunanya.

2.2.3. Pemesanan dan Pembayaran

Menurut **William J. Stevenson** (2020) Pemesanan sebagai "proses menghasilkan produk atau layanan berdasarkan permintaan pelanggan."

Dari definisi para ahli diatas dapat disimpulkan bahwa pemesanan adalah tindakan atau proses memesan produk atau jasa dengan tujuan untuk mendapatkan

barang atau layanan tersebut di waktu yang ditentukan. Ini biasanya melibatkan komunikasi antara pelanggan atau pemesan dengan penyedia barang atau jasa.

Menurut **Friedman** (2019) dalam bukunya "A Theory of the Consumption Function", mendefinisikan pembayaran sebagai "pengeluaran uang untuk memperoleh barang atau jasa." Definisi ini menyoroti hubungan antara uang dan perolehan barang atau jasa.

2.2.4. Metode FIFO

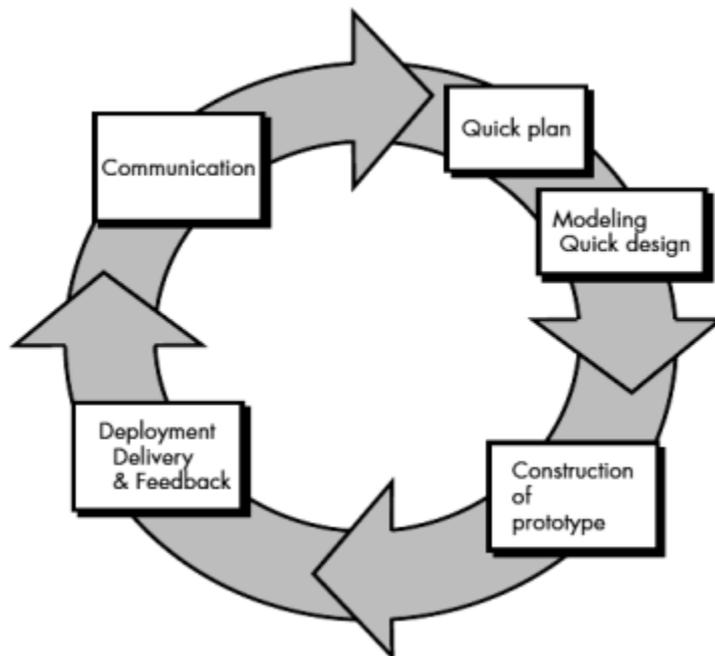
Menurut **Sari** (2020) Metode FIFO, atau First-In-First-Out, merupakan suatu pendekatan dalam pengelolaan dan penilaian persediaan yang didasarkan pada prinsip pertama masuk, pertama keluar. Artinya, barang atau item yang pertama kali masuk ke dalam stok akan menjadi yang pertama kali digunakan atau dijual. Metode ini diterapkan untuk memastikan bahwa barang yang lebih lama berada di dalam persediaan akan diprioritaskan penggunaannya atau penjualannya, sementara yang lebih baru akan tetap di simpanan. Konsep dasar dari metode FIFO terletak pada pemahaman urutan waktu barang masuk dan keluar dari persediaan. Dalam lingkungan manufaktur atau ritel, metode ini membantu mengelola persediaan dengan cara yang mengurangi risiko kerusakan atau usang pada barang yang terlalu lama disimpan. FIFO juga memberikan keuntungan dalam penilaian persediaan, karena nilai persediaan yang diakui dalam laporan keuangan akan mencerminkan biaya barang yang lebih akurat.

2.3. Metode Pengembangan Sistem

Metode prototype merupakan sebuah metode yang mengembangkan aplikasi dengan cara memberikan contoh penawaran sebuah rancangan kepada orang yang akan menjadi pemakai aplikasi dan memberikan evaluasi prototype

sebelum dilakukan penulisan syntak. Dalam penggunaan metode ini, calon pengguna atau pemakai juga ikut berperan dalam proses pembuatan aplikasi dan penggunaan prinsip-prinsip keahlian teknik untuk mendapatkan perangkat lunak yang ekonomis yang handal dan bekerja secara efisien (Indahningrum, 2020).

Prototype sering diwujudkan dalam bentuk user interface program aplikasi dan contoh-contoh reporting yang akan dihasilkan, sehingga pengguna sistem akan mempunyai gambaran tentang sisten yang akan digunakan(Syarif, 2020).



Gambar 2.1. Metode *Prototype* (Fadilah,Qintari et al.2019)

Tahapan dalam metode pengembangan sistem *prototype* yaitu :

1. ***Communication***

Tahap pertama dalam metode prototipe adalah mengidentifikasi kebutuhan awal sistem. Ini melibatkan berkomunikasi dengan pemangku kepentingan,

termasuk pengguna akhir, manajemen, dan pihak terkait lainnya, untuk memahami kebutuhan, tujuan, dan harapan mereka terhadap sistem yang akan dikembangkan. Dokumentasikan persyaratan awal ini dengan jelas.

2. *Quick Plan*

Langkah ini melibatkan penyusunan rencana awal yang mencakup gambaran umum tentang bagaimana proyek akan dijalankan, termasuk sumber daya yang diperlukan, jadwal kerja, dan tujuan proyek.

3. *Modelling Quick Design*

Di tahap ini, model awal sistem yang akan dikembangkan dibuat. Model ini mencakup fitur-fitur utama, antarmuka pengguna, dan fungsi-fungsi kunci yang direncanakan. Model ini berfungsi sebagai dasar untuk prototipe.

4. *Construction of Prototype*

Pada tahap ini, prototipe sistem sebenarnya dibangun. Prototipe ini mencerminkan konsep sistem dan mungkin berisi sebagian dari fitur dan fungsi yang direncanakan.

5. *Deployment Delivery dan Feedback*

Setelah prototipe telah dikonstruksi, ada tahap pengiriman atau implementasi. Prototipe ini dapat disampaikan kepada pemangku kepentingan atau pengguna akhir untuk diuji dan dievaluasi. Terakhir, umpan balik dari pengguna dan pemangku kepentingan diterima dan digunakan untuk membuat perbaikan pada prototipe. Umpan balik ini dapat digunakan untuk merevisi prototipe atau sebagai masukan untuk pengembangan selanjutnya.

2.4. UML

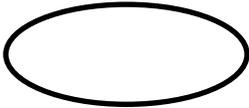
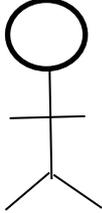
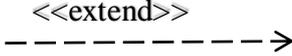
Unified Modelling Language (UML) Menurut (Ariani Sukamto & Shalahuddin, 2019), Unified Modelling Language (UML) adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan requirement, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek. UML digunakan untuk memodelkan suatu sistem (bukan hanya perangkat lunak) yang menggunakan konsep berorientasi objek. Dan juga untuk menciptakan suatu bahasa pemodelan yang dapat digunakan baik oleh manusia maupun mesin. UML menyediakan notasi notasi yang membantu memodelkan sistem dari berbagai perspektif UML tidak hanya digunakan dalam pemodelan perangkat lunak, namun hampir dalam semua bidang yang membutuhkan pemodelan (Feby Prasetya et al., 2022).

2.4.1. Use Case Diagram

Use Case atau diagram use case merupakan pemodelan untuk kelakuan (behavior) sistem informasi yang akan dibuat. Use Case menggambarkan interaksi antara satu atau lebih aktor dan sistem informasi yang akan dibuat. Secara garis besar, use case digunakan untuk mendefinisikan fungsi yang tersedia di dalam sistem informasi dan siapa yang berhak menggunakannya (Ariani Sukamto & Shalahuddin, 2019).

Berikut ini adalah simbol-simbol yang ada pada use case diagram :

Tabel 2.1. Simbol-simbol *Use Case Diagram*

No	Simbol	Keterangan
1	<p><i>Use Case</i></p> 	<p>Fungsional yang disediakan system sebagai unit-unit yang saling bertukar pesan antar unit atau antar aktor biasanya dinyatakan dengan menggunakan kata kerja di awal frase name usecase</p>
2	<p><i>Aktor/ actor</i></p> 	<p>aktor / actor Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang tapi aktor belum tentu merupakan orang;biasanya</p>
3	<p><i>Asosiasi/ assiciation</i></p> 	<p>Komuniasi antara aktor dan use case yang berpartisipasi pada use case atau use case memiliki intreraksi dengan aktor</p>
4	<p><i>Ektensii/ extend</i></p> 	<p>Relasi use case tambahan sebuah use case dimana use case yang ditambahkan dapat berdiri sendiri walau tanpa usecase tambahan itu, mirip dengan prinsip inheritance pada pemrograma berorientasi objek,biasanya use case tambahan memiliki nama depan yang sama dengan use case yang ditambahkan</p>

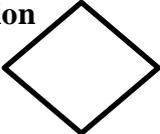
5	<p>Generalisasi/Generalization</p> 	<p>Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah use case dimana fungsi yang satunya adalah fungsi yang lebih umum dari yang lainnya.</p>
6	<p>Menggunakan/Include/uses</p> 	<p>Relasi use case tambahan ke sebuah use case di mana use case yang ditambahkan memerlukan use case ini untuk menjalankan fungsinya atau sebagai syarat dijalankan use case ini</p>

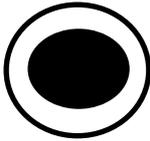
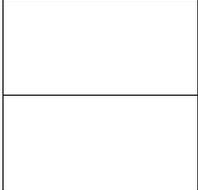
Sumber : ((Ariani Sukamto & Shalahuddin, 2019).

2.4.2. Activity Diagram

Diagram Activity atau activity diagram digunakan untuk menggambarkan workflow (aliran kerja) atau aktivitas dari suatu sistem atau proses bisnis atau menu di dalam perangkat lunak (Ariani Sukamto & Shalahuddin, 2019). Berikut adalah simbol-simbol yang ada pada diagram aktivitas :

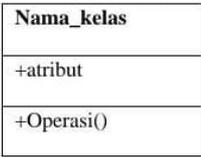
Tabel 2.2. Simbol-simbol Activity Diagram

No	Simbol	keterangan
1	<p>Status awal</p> 	<p>Status awal aktivitas sistem, sebuah diagram aktivitas memiliki status awal</p>
2	<p>Percabangan / decision</p> 	<p>Aktivitas yang di lakukan system aktivitas biasanya diawali dengan kata kerj</p>

3		Penggabungan/joinAsosiasi penggabungan dimana lebih daPri satu aktivitas digabungkan menjadi satu
4	Status akhir 	Status akhir yang dilakukan sistem, sebuah diagramaktivitas memiliki sebuah status akhir.
5	Swimlane Name Swimlane 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

2.4.3. Class Diagram

Diagram kelas atau class diagram adalah menggambarkan struktur system informasi dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Diagram kelas dibuat agar pembuatan rogram atau programmer membuat kelas-kelas sesuai rancangan di dalam diagram kelas antara dokumentasi perancang dan perangkat lunak sinkron (Ariani Sukamto & Shalahuddin, 2019). Berikut ini adalah simbol-simbol yang ada pada class diagram

No	Simbol	Keterangan
1	<p>Kelas</p> 	Kelas pada struktur sistem
2	<p>Antarmuka / <i>interface</i></p>  <p><i>Nama_interfase</i></p>	Sama dengan konsep interface dalam pemrograman berorientasi objek.
3	<p>Asosiasi / <i>association</i></p> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
4	<p>Asosiasi berarah / <i>directed association</i></p> 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
5	<p>Generalisasi</p> 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum

Tabel 2.3. Simbol-simbol *Class Diagram*

No	Simbol	Keterangan
		ke khusus)
6	Kebergantungan/ <i>dependency</i> ----->	Relasi antar kelas dengan makna kebergantungan antar kelas
7	Agregasi / <i>aggregation</i> —————◆	Relasi antarkelas dengan makna semua-bagian (whole-part)

Sumber : (Ariani Sukamto & Shalahuddin, 2019)

2.5. *Blackbox Testing*

Metode *Black Box Testing* adalah salah satu teknik pengujian perangkat lunak yang difokuskan pada pengujian fungsionalitas suatu sistem tanpa memerhatikan detail internal kode sumber atau struktur program. Dalam metode ini, pengujian dilakukan dengan cara memperlakukan sistem sebagai kotak hitam di mana Anda hanya tahu apa yang dimasukkan ke dalam sistem (input) dan apa yang seharusnya dikeluarkan (output), tanpa perlu mengetahui bagaimana sistem mencapai hasil tersebut.

Berikut beberapa poin penting terkait dengan metode Black Box Testing:

1. Tidak Memeriksa Kode Sumber Metode Black Box Testing tidak memeriksa kode sumber atau detail implementasi internal. Sebagai pengujian yang berfokus pada perilaku eksternal sistem, pengujian ini cocok untuk pengujian fungsionalitas dan kesesuaian dengan persyaratan.
2. Berdasarkan Spesifikasi Eksternal: Pengujian Black Box didasarkan pada spesifikasi eksternal dan persyaratan perangkat lunak. Ini termasuk

dokumen persyaratan, desain sistem, atau spesifikasi fungsional yang telah ditentukan sebelumnya.

3. Fokus pada Fungsi dan Input-Output: Pengujian Black Box lebih memperhatikan input yang diberikan ke sistem dan output yang dihasilkan. Tujuannya adalah untuk memastikan bahwa sistem menghasilkan output yang benar berdasarkan input yang benar, sesuai dengan persyaratan yang telah ditetapkan.

2.6. ISO 25010

Model ini merupakan bagian dari *Software product Quality Requirements and Evaluation (SQuaRE)*, dimana model ini berkaitan dengan model kualitas perangkat lunak yang merupakan pengembangan dari model sebelumnya. Pada model ini terdapat beberapa sub-karakteristik tambahan dan beberapa sub-karakteristik yang dipindahkan ke karakteristik lain. Berikut ini merupakan karakteristik atau faktor kualitas internal dan eksternal yang terdapat pada model ISO-25010.

Tabel 2.4. ISO 25010

Faktor	Sub Faktor
<i>Functional Suitability</i>	<ol style="list-style-type: none"> 1. <i>Appropriateness</i> 2. <i>Accuracy</i> 3. <i>Functional Suitability Compliance</i>
<i>Reliability</i>	<ol style="list-style-type: none"> 1. <i>Availability</i> 2. <i>Fault Tolerance</i> 3. <i>Recoverability</i> 4. <i>Reliability Compliance</i>
<i>Performance Efficiency</i>	<ol style="list-style-type: none"> 1. <i>Time-behaviour</i> 2. <i>Resource-utilisation</i> 3. <i>Performance Efficiency Compliance</i>
<i>Operability</i>	<ol style="list-style-type: none"> 1. <i>Appropriateness Recognisability</i> 2. <i>Learnability</i> 3. <i>Ease of Use</i> 4. <i>Helpfulness</i> 5. <i>Attractiveness</i>

	6. <i>Technical Accessibility</i> 7. <i>Operability Compliance</i>
<i>Security</i>	1. <i>Confidentiality</i> 2. <i>Integrity</i> 3. <i>Non-repudiation</i> 4. <i>Accountability</i> 5. <i>Authenticity</i> 6. <i>Security Compliance</i>
<i>Compatibility</i>	1. <i>Replaceability</i> 2. <i>Co-existence</i> 3. <i>Interoperability</i> 4. <i>Compatibility Compliance</i>
<i>Maintainability</i>	1. <i>Modularity</i> 2. <i>Reusability</i>
<i>Functional Suitability</i>	3. <i>Appropriateness</i> 4. <i>Accuracy</i> 5. <i>Functional Suitability Compliance</i>
<i>Reliability</i>	1. <i>Availability</i> 2. <i>Fault Tolerance</i> 3. <i>Recoverability</i> 4. <i>Reliability Compliance</i>
<i>Maintainability</i>	1. <i>Modularity</i> 2. <i>Reusability</i> 3. <i>Analyzability</i> 4. <i>Changeability</i> 5. <i>ModificationStability</i> 6. <i>Testability</i> 7. <i>Maintainability Complianc</i>
<i>Transferability</i>	1. <i>Portability</i> 2. <i>Adaptability</i> 3. <i>Installability</i> 4. <i>Transferability Compliance</i>

Berikut ini merupakan pengertian dari masing-masing faktor dan sub-faktor yang terdapat pada model ISO-25010, antara lain :

1. *Functional Suitability*

Functional suitability merupakan tingkat dimana perangkat lunak dapat menyediakan fungsionalitas yang dibutuhkan ketika perangkat lunak digunakan pada kondisi yang spesifik.

2. *Reliability*

Reliability merupakan tingkatan dimana perangkat lunak dapat bertahan pada tingkatan tertentu ketika digunakan oleh pengguna pada kondisi yang spesifik.

3. *Performance Efficiency*
Performance efficiency merupakan tingkat dimana perangkat lunak dapat memberikan kinerja yang tepat terhadap sejumlah sumber daya yang digunakan pada kondisi tertentu.
4. *Operability*
Operability merupakan tingkat dimana perangkat lunak dapat dimengerti, dipelajari, digunakan, dan menarik perhatian pengguna ketika digunakan pada kondisi tertentu.
5. *Security*
Security merupakan perlindungan terhadap perangkat lunak dari berbagai ancaman, akses atau penggunaan dari pengguna yang tidak dikenal.
6. *Compatibility*
Faktor ini merupakan kemampuan dari dua atau lebih komponen perangkat lunak untuk dapat melakukan pertukaran informasi dan melakukan fungsi yang dibutuhkan ketika digunakan pada hardware atau lingkungan perangkat lunak yang sama.
7. *Maintainability*
Maintainability merupakan tingkat dimana sebuah perangkat lunak dapat dimodifikasi. Modifikasi ini termasuk perbaikan, perubahan atau penyesuaian perangkat lunak untuk dapat berubah pada lingkungan, kebutuhan, dan fungsionalitas yang spesifik.
8. *Transferability*
Transferability merupakan tingkat dimana perangkat lunak dapat berpindah dari lingkungan yang satu ke lingkungan yang lain.

2.7. Skala Likert

Skala *Likert* merupakan skala yang didesain untuk menilai sejauh mana responden setuju atau tidak dengan pernyataan pada skala dengan susunan sebagai berikut ini :

Tabel 2.5. Skala *Likert*

Jawaban	Skor
Sangat Setuju	5
Setuju	4
Ragu-Ragu	3
Tidak Setuju	2
Sangat Tidak Setuju	1

Dengan skala *Likert*, maka variabel yang akan diukur dijabarkan menjadi indikator variabel. Kemudian indikator tersebut dijadikan titik tolak untuk menyusun instrument-instrumen berupa pernyataan.

Pada tahap pengolahan data dalam penelitian ini menggunakan jenis analisis deskriptif yang merupakan jenis penelitian yang menggambarkan fakta-fakta yang ada untuk selanjutnya diolah menjadi data. Data tersebut kemudian dianalisis untuk memperoleh suatu kesimpulan. Analisis statistik deskriptif digunakan untuk menggambarkan bagaimana tingkat kualitas *prototype* yang dibuat.

Langkah-langkah yang dilakukan dalam analisis statistik deskriptif tersebut adalah sebagai berikut (Narimawati, 2007):

1. Setiap indikator yang dinilai oleh responden, diklasifikasikan dalam lima alternatif jawaban dengan menggunakan skala ordinal yang menggambarkan peringkat jawaban.
2. Dihitung total skor setiap variabel/subvariabel = jumlah skor dari seluruh indikator variabel untuk semua responden.
3. Dihitung skor setiap variabel/subvariabel = rata-rata dari total skor.
4. Untuk mendeskripsikan jawaban responden, juga digunakan statistik deskriptif seperti distribusi frekuensi dan tampilan dalam bentuk tabel atau grafik.
5. Untuk menjawab deskripsi tentang variabel penelitian ini, digunakan rentang kriteria penelitian sebagai berikut :

$$SkorTotal = \frac{skorAktual}{skorIdeal} \times 100 \%$$

Skor aktual adalah jawaban seluruh responden atas kuesioner yang telah diajukan. Skor ideal adalah skor atau bobot tertinggi atau semua responden diasumsikan memilih jawaban dengan skor tertinggi. Penjelasan bobot nilai skor aktual dapat dilihat pada tabel berikut :

Tabel 2.6. Kriteria Presentase Tanggapan Responden

% Jumlah Skor	Kriteria
20,00 % - 36,00 %	Tidak Baik
36,01 % - 52,00 %	Kurang Baik
52,01 % - 68,00 %	Cukup
68,01 % - 84,00 %	Baik
84,01 % - 100 %	Sangat Baik