

BAB II

TINJAUAN PUSTAKA

2.1 Sistem

Pada dasarnya, sistem adalah sekumpulan elemen yang saling terkait atau terpadu yang dimaksudkan untuk mencapai suatu tujuan. Sistem adalah dua atau lebih komponen yang saling berhubungan dan berintraksi membentuk kesatuan kelompok sehingga menghasilkan satu tujuan [1].

2.2 Informasi

McFadden, dan kawan-kawan mendefinisikan informasi sebagai data yang telah diproses sedemikian rupa sehingga meningkatkan pengetahuan seseorang yang menggunakan data tersebut. Shannon dan Weaver, dua orang insinyur listrik melakukan pendekatan secara matematis untuk mendefinisikan informasi (Kroenke). Menurut mereka, informasi adalah jumlah ketidakpastian yang dikurangi ketika sebuah pesan diterima. Artinya, dengan adanya informasi, tingkat kepastian menjadi meningkat. Informasi merupakan sebuah data yang dikelola menjadi sesuatu yang lebih bernilai tinggi bagi penerima guna untuk membantu membuat sebuah pengambilan Keputusan [2].

2.3 Sistem Informasi

Sebenarnya yang dimaksud sistem informasi tidak harus melibatkan komputer. Sistem informasi yang menggunakan komputer biasa disebut sistem informasi berbasis komputer (*Computer Based Information System* atau CBIS). Dalam praktik, istilah sistem informasi lebih sering dipakai tanpa berbasis komputer, walaupun dalam kenyataannya komputer merupakan bagian yang penting. Di buku ini, yang dimaksudkan dengan sistem informasi adalah sistem informasi berbasis komputer. Ada beragam definisi sistem informasi, yaitu :

- a. Alter, sistem informasi adalah kombinasi antar prosedur kerja, informasi, orang dan teknologi informasi yang diorganisasikan untuk mencapai tujuan dalam sebuah organisasi.
- b. Bodnar dan Hopwoo, sistem informasi adalah kumpulan perangkat keras dan perangkat lunak yang dirancang untuk mentransformasikan data ke dalam bentuk informasi yang berguna.
- c. Gelinas, Oram dan Wiggins, sistem informasi adalah suatu sistem buatan manusia yang secara umum terdiri atas sekumpulan komponen berbasis komputer dan manual yang dibuat untuk menghimpun, menyimpan dan mengelola data serta menyediakan informasi keluaran kepada para pemakai.
- d. Hall, Sistem informasi adalah sebuah rangkaian prosedur formal, dimana data dikelompokkan, diproses menjadi informasi dan didistribusikan kepada para pemakai.
- e. Turban, McLean dan Wetherbe, Sebuah sistem informasi mengumpulkan, memproses, menyimpan, menganalisis dan menyebarkan informasi untuk tujuan yang spesifik.
- f. Wilkinson, Sistem informasi adalah kerangka kerja yang mengkoordinasikan sumber daya (manusia dan komputer) untuk mengubah masukan (*input*) menjadi keluaran (informasi) guna mencapai sasaran-sasaran perusahaan. Sistem informasi adalah gugus elemen dalam pembuatan dan penyampaian informasi dalam sebuah perusahaan atau organisasi. Keandalan suatu sistem informasi dalam suatu organisasi tergantung pada hubungan antara elemen-elemen tersebut, sehingga dapat memberikan data yang berguna bagi berbagai institusi yang terlibat [3].

2.4 UMKM

UMKM adalah kegiatan usaha yang dijalankan oleh perseorangan atau individu, rumah tangga, atau badan usaha skala kecil. Biasanya bisnis UMKM digolongkan melalui pendapatan per tahun, jumlah karyawan, dan aset yang dimiliki. Pemerintah telah mengatur seputar pengelolaan UMKM ini dalam UU Nomor 20 Tahun 2008 tentang Usaha Mikro, Kecil, dan Menengah [4].

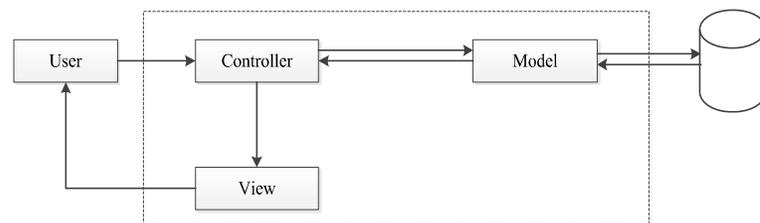
Kriteria UMKM yang terakhir terletak pada hasil penjualan tahunan. Menurut PP No. 7 Tahun 2021 hasil penjualan tahunan atau omset usaha dari masing-masing badan usaha adalah sebagai berikut:

1. Usaha Mikro: Maksimal Rp2 Milyar
2. Usaha Kecil: Rp2 Milyar – Rp15 Milyar
3. Usaha Menengah: Rp15 Milyar – 50 Milyar

2.5 CodeIgniter

CodeIgniter adalah *Framework* untuk bahasa pemrograman *PHP*, yang dibuat Rick Ellis pada tahun 2006. *CodeIgniter* memiliki banyak fitur yang membantu para pengembang *PHP* untuk dapat membuat aplikasi secara mudah dan cepat serta memiliki sifat yang fleksibel dapat mengembangkan dalam perangkat *web*, dekstop maupun *mobile*” [5].

CodeIgniter memiliki konsep atau pola *Model-View-Controller* (MVC) sehingga kode-kode dapat di sederhanakan.



Gambar 2. 1 *Arsitektur MVC*

2.6 Web based

Web based adalah aplikasi yang dibuat berbasis *web* yang membutuhkan *web server* dan browser untuk menjalankannya [6].

Dengan membuat sistem berbasis *web based* ada beberapa hal yang penting dan harus kita pikirkan sebelum membangun sistem tersebut, diantaranya :

1. Tidak membutuhkan *Hardware* dengan spesifikasi yang tangguh untuk menjalankan aplikasinya.

2. *Server* yang dibutuhkan cukup diinstallkan tools pendukung saja agar klien mudah menjalankan aplikasi
3. Infrastruktur jaringan yang dibutuhkan juga cukup besar karena aplikasi yang dibuat dapat diakses dari jaringan luar (internet).
4. Aplikasi berbasis *web based* dapat diakses dari berbagai perangkat dengan syarat menggunakan *web browser* saja sudah dapat mengaksesnya.
5. Jika aplikasi yang sudah jadi ingin di *update*, sangat mudah untuk melakukannya karena tidak membutuhkan membuka keseluruhan aplikasi.

2.7 PHP

PHP adalah bahasa *server-side-scripting* yang menyatudengan *HTML* untuk membuat halaman *web* yang dinamis [7].

PHP Hypertext Preprocessor adalah bahasa pemrograman *web server-side* yang bersifat open source. *PHP* merupakan *script* yang terintegrasi dengan *HTML* dan berada pada *server* (*server side HTML embedded scripting*). *PHP* adalah *script* yang digunakan untuk membuat halaman *website* yang dinamis. Dinamis berarti halaman yang akan ditampilkan dibuat saat halaman itu diminta oleh *client*. Mekanisme ini menyebabkan informasi yang diterima *client* selalu yang terbaru/*up to date*. Semua *script PHP* dieksekusi pada *server* dimana *script* tersebut dijalankan. Dengan menggunakan program *PHP*, sebuah *website* akan lebih interaktif dan dinamis. Sehingga *PHP* merupakan bahasa pemrograman yang digunakan oleh pengembang untuk membuat sistem *website* dengan kumpulan bahasa *HTML* dan *script* lainnya.

2.8 Mysql

Mysql adalah singkatan dari *Structure Query Language* yang digunakan untuk mendefinisikan structure data, memodifikasi data pada basis data, menspesifikasi batasan keamanan (*security*), hingga pemeliharaan data. (*Mysql*, 2018),

Mendefinisikan *mysql* adalah *RDBMS* yang cepat dan mudah digunakan, serta sudah banyak digunakan untuk berbagai kebutuhan.

Mysql merupakan bahasa standar yang paling banyak digunakan untuk mengakses database relasional dan merupakan aplikasi yang dapat dipergunakan secara bebas [8].

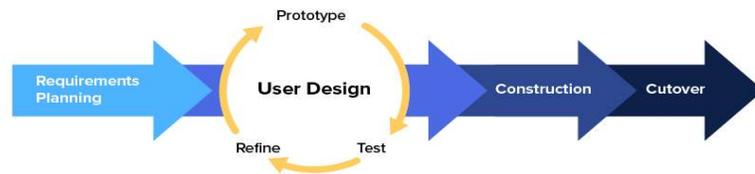
2.9 Metode Pengembangan Sistem

Metode pengembangan sistem merupakan metode yang digunakan sebagai alur proses dalam pengembangan, sehingga penelitian dapat dikembangkan sesuai tahapan dari metode pengembang sistem.

2.9.1 *Rapid Application Development (RAD)*

Metode yang berfokus pada pengembangan aplikasi secara cepat, melalui pengulangan dan *feedback* berulang-ulang. *RAD* diajukan oleh *IBM* pada tahun 1980 sampai 1990-an, ketika permintaan terhadap aplikasi semakin meningkat. Dengan banyaknya *demand*, orang-orang di dunia *IT* harus mencari solusi untuk memenuhi permintaan tersebut. Metode ini merupakan semacam cikal bakal *agile project management*, karena bisa mengikuti *pace* bisnis yang terus berkembang dan juga kebutuhan pasar yang terus meningkat. Pengembangan *Software* pada umumnya seperti *waterfall* model membutuhkan perencanaan yang terbilang cukup kaku. Klien atau pelanggan seakan ‘dipaksa’ untuk menyetujui banyak hal di awal, tetapi mereka tidak bisa melihat proses pembuatannya.

Keuntungan utama menjalankan *Rapid Application Development* adalah jangka waktu pengembangan lebih cepat. Hal ini dikarenakan *feedback* dari pelanggan cepat didapatkan dan semua perubahan yang dilakukan akan sesuai hasil tersebut. Akan tetapi, salah satu kekurangan *RAD* adalah kamu membutuhkan tim berisikan *developer* yang benar-benar memiliki *skill* tinggi dan juga metode ini hanya bisa digunakan untuk proyek yang bisa termodulasi [9].



Gambar 2. 2 *Rapid Application Development (RAD)*
Sumber: (Rosa and Shalahuddin, 2019)

1. Kelebihan Model *RAD*

Kelebihan metodologi *RAD* :

- a. Penghematan waktu dalam keseluruhan fase proyek dapat dicapai.
- b. *RAD* mengurangi seluruh kebutuhan yang berkaitan dengan biaya proyek dan sumberdaya manusia.
- c. *RAD* sangat membantu pengembangan aplikasi yang berfokus pada waktu penyelesaian proyek.
- d. Perubahan desain sistem dapat lebih berpengaruh dengan cepat dibandingkan dengan pendekatan SDLC tradisional.
- e. Sudut pandang user disajikan dalam sistem akhir baik melalui fungsi-fungsi sistem atau antarmuka pengguna.
- f. *RAD* menciptakan rasa kepemilikan yang kuat di antara seluruh pemangku kebijakan proyek.

2. Kelemahan Model *RAD*

Kelemahan pada pengembangan tersebut dapat dilihat berdasarkan kesesuaian pengembangan yang dilakukan, berikut adalah kelemahan metode pengembang sistem *RAD* :

- a. Dengan metode *RAD*, penganalisis berusaha mempercepat proyek dengan terburu-buru.
- b. Kelemahan yang berkaitan dengan waktu dan perhatian terhadap detail. Aplikasi dapat diselesaikan secara lebih cepat, tetapi tidak

mampu mengarahkan penekanan terhadap permasalahan-permasalahan perusahaan yang seharusnya diarahkan.

- c. *RAD* menyulitkan *programmer* yang tidak berpengalaman menggunakan prangkat ini di mana *programmer* dan *analyst* dituntut untuk menguasai kemampuan-kemampuan baru sementara pada saat yang sama mereka harus bekerja mengembangkan sistem

2.10 Tahap Penelitian

Tahapan dalam penelitian sebagai langkah-langkah penelitian yang harus dikerjakan, berikut adalah tahapan penelitian *Rapid Application Development*.

1. Tahap *Requirements Project*

RAD dimulai dengan menentukan kebutuhan sebuah proyek (*project requirements*). Pada tahap ini, tim perlu menentukan kebutuhan yang ingin dipenuhi dari sebuah proyek. Kebutuhan ini tidak perlu spesifik. Tapi, sifatnya benar-benar umum dan jumlahnya bisa banyak. Baru dari situ, tim akan menentukan mana kebutuhan yang perlu diprioritaskan. Setelah mendapatkan kebutuhan yang jelas, barulah tim menentukan hal-hal yang lebih detail. Misalkan seperti tujuan, timeline, dan budget yang diperlukan.

2. Tahap Membuat *Prototype*

Hal yang selanjutnya dilakukan adalah membuat *prototype*.

Developer secepat mungkin akan membuat *prototype* dari aplikasi yang diinginkan. lengkap dengan fitur dan fungsi yang berbeda-beda. Tujuannya, sekedar untuk mengecek apakah *prototype* yang dibuat sudah sesuai dengan kebutuhan klien. Meski begitu, tahap ini bisa saja dilakukan berulang-ulang. Kadang juga melibatkan user untuk testing dan memberikan *feedback*. Proses ini memungkinkan tim mempelajari error yang mungkin muncul ke depannya. Ini berguna untuk mengurangi *error* dan *debugging*. Lewat tahapan ini, tim *developer* memiliki modal untuk membuat aplikasi yang mudah dipakai, stabil, tidak sering *error*, dan desainnya pun oke.

3. Proses Pengembangan dan Pengumpulan *Feedback*

Setelah tahu aplikasi seperti apa yang ingin dibuat, *developer* mengubah *prototype* ke bentuk aplikasi versi beta sampai dengan final. Jadi, bisa dibilang tahap *RAD* inilah yang cukup intens. *Developer* terus-menerus melakukan coding aplikasi, melakukan testing sistem, dan integrasi dengan bagian-bagian lainnya. Karena itulah, *developer* menggunakan *tools* dan *framework* yang mendukung *RAD* agar cepat. Apalagi proses ini terus diulang sambil terus mempertimbangkan *feedback* dari klien. Baik itu soal fitur, fungsi, *Interface*, sampai keseluruhan aspek dari produk yang dibuat. Nah, kalau prosesnya berjalan lancar, *developer* akan melanjutkan ke langkah berikutnya. Yaitu, finalisasi produk atau implementasi. Kalau pun tidak, proses ini kemungkinan akan terus diulang. Pun, kalau apes-apesnya aplikasi tidak menjawab kebutuhan, *developer* akan kembali ke proses *prototyping*.

4. Tahap *Implementation* (implementasi).

Merupakan tahap pengujian terhadap aplikasi yang dikembangkan. Tahap ini *programmer* mengembangkan desain menjadi suatu program kemudian dilakukan proses pengujian untuk memeriksa kesalahan sebelum diaplikasikan.

2.11 Alat Perancang Sistem

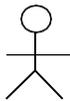
Unified Modelling Language adalah bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung. Berikut ini merupakan penjelasan tentang masing-masing diagram yang ada pada *Unified Modelling Language* [10].

2.11.1 *Use Case Diagram*

Use case mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut [10].

Berikut simbol-simbol yang akan digunakan dalam menggambarkan *Use Case Diagram* dapat dilihat pada Tabel 2.1 :

Tabel 2. 1 Simbol-simbol *Use Case Diagram*

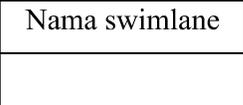
No	Simbol	Keterangan Fungsi
1.	Aktor 	Aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu.
2.	<i>Use Case</i> 	<i>Use Case</i> adalah deskripsi dari urutan aksi- aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
3.	Asosiasi 	Asosiasi adalah apa yang menghubungkan antara objek satu dengan objek yanglainnya.
4.	Generalisasi 	Generalisasi adalah hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada diatasnya atau sebaliknya dari bawah ke atas.
5.	<i>Defendancy</i> 	<i>Defendancy</i> (ketergantungan) adalah hubungan dimana perubahan yang terjadi pada suatu elemen defenden (mandiri) akan mempengaruhi elemen yang bergantung padanya (independen).

Sumber: (Rosa dan Salahuddin, 2019)

2.11.2 *Activity Diagram*

Activity Diagram menggambarkan aliran kerja atau aktivitas dari sebuah sistem atau proses bisnis atau menggambarkan aktivitas sistem bukan apa yang dilakukan *actor* [10]. Berikut simbol-simbol yang akan digunakan dalam menggambarkan *activity diagram* dapat dilihat pada Tabel 2.2:

Tabel 2. 2 Simbol *Activity Diagram*

No.	Simbol	Keterangan
1.		Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
2.		Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3.		Percabangan (<i>Decision</i>) merupakan asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
4.		Penggabungan (<i>Join</i>) merupakan asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
5.		<i>Swimlane</i> Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas.
6.		Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.

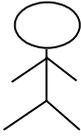
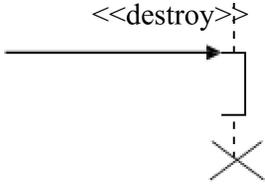
Sumber: (Rosa dan Salahuddin, 2019)

2.11.3 *Sequence Diagram*

Diagram *sequence* menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dengan *message* yang dikirimkan dan diterima antar objek [10]. Oleh karena itu untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Membuat diagram sekuen juga dibutuhkan untuk melihat skenario yang ada pada *use case*. Banyaknya diagram sekuen yang harus digambar adalah minimal sebanyak pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use case* yang telah didefinisikan interaksi jalannya pesan sudah dicakup dalam diagram sekuen sehingga semakin banyak *use*

case yang didefinisikan maka diagram sekuen yang harus dibuat juga semakin banyak. Berikut adalah simbol-simbol yang ada pada *Sequence Diagram* pada Tabel 2.3.

Tabel 2. 3 Simbol-simbol *Sequence Diagram*

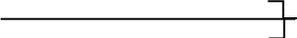
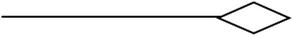
No.	Simbol	Deskripsi
1	<p>Aktor</p>  <p>Atau <u>Nama aktor</u></p> <p>Tanpa waktu aktif</p>	<p>Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan dalam menggunakan kata benda di awal frase nama aktor</p>
2.	<p>Garis hidup /<i>lifeline</i></p> 	<p>Menyatakan kehidupan suatu objek</p>
3.	<p>Objek</p> <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 0 auto;"> <p><u>Nama objek: nama kelas</u></p> </div>	<p>Menyatakan objek yang berinteraksi peran</p>
4	<p>Pesan tipe <i>destroy</i></p> 	<p>Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaliknya jika ada <i>create</i> maka ada <i>Destroy</i></p>

Sumber: (Rosa dan Salahuddin, 2019)

2.11.4 Class Diagram

Class diagram mengembangkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem [10]. Berikut simbol-simbol yang akan digunakan dalam menggambarkan *Class Diagram* dapat dilihat pada Tabel 2.4:

Tabel 2. 4 Simbol *Class Diagram*

No.	Simbol	Deskripsi
1.		Kelas pada struktur sistem.
2.	<p>Antar Muka/Interface</p>  <p>Nama_Interface</p>	Sama dengan konsep interface dalam pemrograman berorientasi objek.
3.	<p>Asosiasi / Asociation</p> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan symbol
4.	<p>Asosiasi Berarah / Directed Association</p> 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan symbol.
5.	<p>Generalisasi</p> 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)
6.	<p>Ketergantungan / dependency</p> 	Relasi antar kelas dengan makna ketergantungan antar kelas.
7.	<p>Agregasi / aggregation</p> 	Relasi antar kelas dengan maksna semua bagian (<i>whole-part</i>)

Sumber: (Rosa dan Salahuddin, 2019)

2.12 Metode Pengujian Sistem

Metode pengujian sistem merupakan metode yang digunakan untuk melakukan testing pada sistem yang dibangun sehingga di peroleh hasil berupa sistem yang sesuai fungsinya.

2.12.1 *Black Box*

Pengujian kotak hitam dilakukan dengan membuat kasus uji yang bersifat mencoba semua fungsi dengan memakai perangkat lunak apakah sesuai dengan spesifikasi yang dibutuhkan. *Blackbox* testing yaitu menguji perangkat lunak dari spesifikasi fungsional tanpa menguji desain dan kode program. Dari definisi para ahli diatas dapat disimpulkan *Blackbox* Testing adalah metode perancangan data uji yang didasarkan pada spesifikasi perangkat lunak.

2.12.2 Tahapan Pengujian Sistem

Tahapan pengujian sistem digunakan untuk mengetahui proses pengujian yang akan dilakukan, berikut adalah tahapan pengujian *Black Box*:

1. *Descision Table*

Decision Table merupakan cara yang tepat untuk memodelkan logika yang cukup rumit, seperti diagram alur, *if-then-else* dan *switch* laporan kasus. Dalam kondisi ini mengaitkan dengan tindakan untuk melakukan, akan tetapi banyak kasus melakukannya dengan cara yang lebih elegan.

2. *All-Pairs Testing*

All-Pairs Testing atau disebut *pairwise testing* merupakan metode pengujian perangkat lunak kombinatorial yang digunakan untuk setiap pasangan parameter yang masuk kedalam sistem atau algoritma yang ada pada perangkat lunak.

3. *State Transition Table*

State Transition Table merupakan teori automata dan logika skuensial, pada *table* yang menunjukkan *state* dalam pengujian. Pada dasarnya sebuah *table state* merupakan *table* kebenaran yang digunakan untuk beberapa *input* dan *output* termasuk dengan *state* berikutnya dengan kondisi yang sebenarnya terjadi.

4. *Equivalence Partitioning*

Equivalence Partitioning merupakan teknik yang membagi data masukan dari beberapa unit perangkat lunak menjadi beberapa partisi data dari mana *test case* dapat diturunkan. Pada prinsipnya, uji kasus ini dirancang untuk menutupi setiap partisi minimal.

Teknik ini digunakan untuk mendefinisikan kasus pengujian yang mengungkap kelas kesalahan, sehingga mengurangi jumlah pengujian yang harus dilakukan.

5. *Boundary Values Analysis*

Boundary Value Analysis merupakan Pengujian yang dirancang untuk mencakup perwakilan dari batas Nilai-nilai batas. Pada nilai-nilai di sebuah partisi kesetaraan atau sebesar nilai terkecil di kedua sisi tepi.

2.13 Kamus Data

Kamus data adalah katalog fakta tentang data dan kebutuhan-kebutuhan informasi dari suatu sistem informasi. Kamus data adalah kumpulan daftar elemen data yang mengalir pada sistem perangkat lunak sehingga masukan (*input*) dan keluaran (*output*) dapat dipahami secara umum (memiliki standar cara penulisan). Jadi Kesimpulannya “Kamus Data adalah kumpulan atau daftar katalog fakta tentang data dan kebutuhan informasi” [11].

2.14 Penelitian Terkait

Pada era teknologi informasi yang terus berkembang pesat, penerapan sistem informasi telah menjadi landasan utama dalam meningkatkan efisiensi dan kualitas layanan di berbagai sektor. Oleh karena itu, penelitian terkait dengan pemasaran berbasis *website* menjadi langkah penting dalam upaya peningkatan kualitas pelayanan kepada pelanggan. Adapun penelitian yang terkait pada sistem informasi ini terlihat pada tabel 2.5

Tabel 2. 5 Penelitian Terkait

No	Peneliti	Judul	Tahun	Hasil Penelitian	Metode
1	[12]	SISTEM INFORMASI BERBASIS <i>WEB</i> SEBAGAI MEDIA PEMASARAN UMKM DI DESA NGRIMBI	2022	Pembentukan dari sistem informasi berbasis <i>web</i> untuk UMKM di Desa Ngrimbi bertujuan sebagai media pemasaran, sehingga dapat memperluas pasar dari penjualan produk milik para pelaku UMKM.	<i>Waterfall.</i>
2	[13]	Rancang Bangun Sistem Informasi Produk Unggulan Daerah Usaha Kecil Menengah	2018	Sistem ini memungkinkan pengguna untuk mengetahui dan mengenal produk unggulan daerah usaha kecil menengah berdasarkan kategori jenis produk dan berdasarkan kabupaten/kota. Sistem ini akan lebih memudahkan para anggota UKM untuk mempromosikan hasil produk unggulannya.	Metode <i>Rapid Application Development (RAD).</i>

No	Nama Peneliti	Judul	Tahun	Hasil Penelitian	Metode
3	[14]	Perancangan <i>Website</i> Program Studi Sistem Informasi Institut Informatika Dan Bisnis Darmajaya Bandar Lampung	2018	memberikan informasi kepada Mahasiswa, Alumni dan Dosen dengan cepat dan akurat, untuk mengetahui tentang semua informasi yang berhubungan dengan Prodi SI.	<i>RUP (Rational Unified Process)</i>
4	[15]	Pelatihan Dan Pemanfaatan E Commerce Sebagai Media Pemasaran Produk UMKM Di Desa Tales Kecamatan Ngadiluwih, Kabupaten Kediri	2019	Sistem penjualan online untuk umkm	Transaksi elektronik E-Commerce
5	[16]	Sistem Penjualan Pada Toko <i>Online</i> Dengan Metode <i>Rapid Application Development (RAD)</i>	2018	Metode <i>RAD</i> dapat digunakan dalam pembangunan sistem informasi penjualan online.	Metode <i>Rapid Application Development (RAD)</i>