

BAB II

LANDASAN TEORI

2.1 Perancangan Sistem

Sebuah sistem pada dasarnya merupakan sekelompok faktor yang erat hubungannya satu dengan yang lain, yang berperan bersama-sama untuk menggapai tujuan tertentu. Sederhananya, sebuah sistem merupakan suatu kumpulan ataupun himpunan dari faktor, komponen, ataupun variabel yang terorganisir serta saling berhubungan dan bergantung satu sama lain. Perancangan sistem bertujuan untuk mengelola data, menyimpan, menghimpun kerangka kerja yang bertujuan untuk mencapai sasaran yang telah ditetapkan sebelumnya [4].

Analisis dan Perancangan Sistem Informasi adalah sebuah proses yang kompleks yang digunakan untuk mengembangkan dan memelihara sistem informasi ini dibuat berdasarkan tujuan, struktur, dan proses yang dimiliki oleh suatu organisasi dan organisasi ini dapat berupa perusahaan, departemen atau sebuah kelompok sehingga jaringan kerja dari suatu prosedur yang dibuat dapat melaksanakan pokok perusahaan dan menyelesaikan tujuan dan sasaran yang dimaksud [5].

2.2 Definisi Sistem Pencatatan

Pencatatan adalah pembuatan suatu catatan kronologis peristiwa setiap hari dengan metode yang sistematis serta terstruktur. Agar perusahaan dapat mempertahankan eksistensinya dan terus berkembang diperlukan pengelolaan bisnis yang baik dan disertai dengan inovasi-inovasi produk utama yang dihasilkan perusahaan tersebut. Pada perusahaan yang berorientasi pada laba, penjualan merupakan kegiatan utama untuk mencapai sebuah kesuksesan tujuan dalam perusahaan [6].

Pencatatan merupakan proses untuk mengetahui data persediaan yang telah di catat agar bisnis menjadi lebih efisien. Pencatatan pengolahan pada data barang dapat meningkatkan efektifitas perusahaan yaitu bertujuan untuk memudahkan karyawan dalam melakukan bagian pengontrolan persediaan barang. Pencatatan

juga menyediakan informasi informasi yang cepat, tepat dan akurat mengenai data maupun laporan serta memudahkan proses di penyampaian laporan informasi ke kantor pusat. [7].

2.3 Definisi Sistem Persediaan

Persediaan adalah aktivitas yang tersedia untuk dijual dalam kegiatan normal persediaan. Dengan persediaan, perusahaan dapat memenuhi permintaan pelanggan dengan tepat waktu sehingga perusahaan dapat tetap eksis dalam mencapai tujuannya. Persediaan merupakan salah satu faktor penting dalam kegiatan usaha, baik perusahaan dagang maupun manufaktur. Perusahaan dagang maupun manufaktur masing-masing memiliki kegiatan berbeda-beda namun mempunyai tujuan yang sama yakni untuk memenuhi kebutuhan konsumen. [8].

Persediaan merupakan produk yang dibeli untuk tujuan dijual kembali sehingga mendapat keuntungan semaksimal mungkin. Pengendalian persediaan barang pada suatu perusahaan merupakan suatu masalah yang sering dihadapi, dengan pengharapan suatu barang sanggup didapatkan pada tempat dan waktu dengan estimasi pengeluaran biaya yang tergolong terjangkau. Persediaan barang membutuhkan sejumlah waktu untuk proses pemesanan barang untuk pengadaan barang tersebut. [9].

2.4 Definisi Pembelian dan Penjualan

Pembelian adalah kegiatan agar kelancaran transaksi penjualan di dalam perusahaan dapat terjamin. Pembelian memungkinkan perusahaan mengalokasikan sumber daya yang dibutuhkan secara efektif dan efisien. Dalam sebuah perusahaan dagang pembelian meliputi pembelian aktiva produktif, pembelian barang dagangan serta pembelian barang dan jasa lain dalam rangka kegiatan usaha. Pembelian dapat dilakukan secara kredit maupun tunai dan pada umumnya dilakukan kepada beberapa supplier [10].

Penjualan merupakan kegiatan dengan cara menyusun suatu rencana yang sistematis agar dapat memenuhi kebutuhan dan keinginan pembeli atau konsumen guna mencapai penjualan yang menguntungkan. Definisi penjualan merupakan suatu kegiatan transaksi yang dilakukan oleh 2 (dua) belah pihak/lebih dengan

menggunakan alat pembayaran yang sah. Penjualan juga merupakan salah satu sumber pendapatan seseorang atau suatu perusahaan yang melakukan transaksi jual & beli, dalam suatu perusahaan apabila semakin besar penjualan maka akan semakin besar pula pendapatan yang diperoleh seseorang atau perusahaan tersebut [11].

2.5 UMKM

UMKM atau Usaha Mikro, Kecil, dan Menengah adalah kumpulan entitas (subjek atau orang, tujuan, proses, sistem, lingkungan.) yang memberikan nilai bisnis kepada konsumen melalui proses penciptaan dan pengiriman ke dalam suatu jaringan. Sementara itu beberapa lembaga memberikan definisi UMKM secara beragam, diantaranya Badan Pusat Statistik, Lementarian Negara Koperasi dan Usaha Kecil Menengah, Keputusan Menteri Keuangan, Undang-undang no 20 tahun 2008. Definisi UMKM dijelaskan berbeda-beda, namun secara garis besar dikelompokkan dalam tiga parameter yaitu kegiatan usaha berdasarkan jumlah tenaga kerja, aset yang dimiliki dan penghasilan penjualan [12].

Usaha Mikro Kecil Menengah (UMKM) merupakan salah satu wujud dari kegiatan bisnis/usaha atau perekonomian oleh lembaga atau perorangan. Sebagaimana dijelaskan dalam definisi ontologi binis, UMKM merupakan kumpulan dari entitas (misal subyek/orang, tujuan, proses, sistem, lingkungan) yang menawarkan suatu nilai perusahaan kepada konsumen melalui proses membuat dan men-delivery dalam suatu jaringan [13].

2.6 Metode Prototyping

Prototyping merupakan model sebuah pengembangan perangkat lunak yang digunakan developer untuk menggambarkan suatu program kepada pengguna agar dapat lebih memahami dan menentukan program yang sesuai untuk digunakan [14]. Model *prototyping* sendiri merupakan sebuah evolusi dari model pengembangan perangkat lunak sebelumnya yaitu SDLC (*System Development Life Cycle*) [15]. *Prototyping* sendiri pada mulanya berbentuk sebuah *mockup* yang kemudian akan dilakukan evaluasi oleh pengguna. Setelah dilakukan tahap evaluasi *mockup* maka tahap selanjutnya adalah membuat sebuah program berdasarkan hasil *mockup* yang

telah dievaluasi oleh pengguna [16]. Kunci utama dari penggunaan model *prototyping* terletak pada keterlibatan pengguna secara penuh yang akan menguntungkan bagi developer dan pengguna itu sendiri [17]. Berikut ini adalah tahapan-tahapan dalam melakukan *prototyping*, yaitu:



Gambar 2. 1 Tahapan Metode *Prototyping*

Gambar 2.1 ditampilkan tahapan metode pada proses *prototyping* dari proses *communication*, *quick plan*, *modeling quick design*, *construction of prototype*, *deployment delivery & feedback*.

Tahap pertama melakukan komunikasi dengan cara melakukan wawancara dan melakukan analisis kebutuhan pengguna berdasarkan hasil wawancara, untuk tahap kedua yaitu *quick design* pada tahapan ini yaitu membuat desain sederhana yang akan memberikan gambaran singkat tentang aplikasi atau sistem yang akan dibangun, setelah desain cepat atau desain sederhana disetujui selanjutnya adalah pemabngunan *prototype* yang sebenarnya, pada tahap selanjutnya yaitu mengimplementasikan rancangan yang telah dibangun, tahapan terakhir adalah memberikan hasil *prototype* kepada pihak perusahaan, dimana pihak pengguna dapat melakukan evaluasi terhadap rancangan *prototype* yang telah dibuat kemudian pihak perusahaan melakukan *feedback* [18].

Model *prototyping* sendiri memiliki kelebihan dan kekurangan. Kelebihannya, yaitu [19]:

1. Dapat membantu dalam melakukan akomodasi perangkat lunak yang kriteria kebutuhannya belum terpenuhi.
2. Kemudahan dalam mengimplementasikan kepada setiap perangkat lunak.
3. Proses pengembangan sistem yang lebih efisien.
4. Pengguna memiliki peran aktif dalam melakukan pengembangan sistem.

Sedangkan kekurangannya, yaitu:

1. Terkadang pengguna tidak menyadari bahwa *prototyping* bukanlah gambaran keseluruhan dari suatu program dan belum adanya kemampuan untuk *maintenance* dalam jangka.
2. Kualitas dari suatu program nampak tidak sesuai dikarenakan *developer* ingin menyelesaikan proyek dengan waktu yang singkat.

Peneliti menggunakan metode *prototyping* karena metode *prototyping* adalah metode yang pas untuk membangun aplikasi yang baru karena dengan metode ini dapat mendeteksi kesalahan dengan lebih cepat dan *feedback* pengguna juga yang cepat sehingga menghasilkan solusi yang lebih baik dan sesuai dengan keinginan pengguna.

2.7 Unified Model Leveling (UML)

UML (*Unified Modeling Language*) adalah bahasa pemodelan secara visual yang digunakan untuk mendokumentasikan serta penganalisaan desain dalam melakukan pengembangan sistem berorientasi objek. Perancangan perangkat lunak (program-program aplikasi) sesungguhnya merupakan salah satu pekerjaan yang paling banyak menuntut keahlian dan ketrampilan manusia, yaitu kemampuan dalam hal analisis dan perancangan, kemampuan teknis, serta kemampuan manajerial [20].

Unified Modelling Language (UML) adalah suatu alat untuk memvisualisasikan dan mendokumentasikan hasil analisa dan desain yang berisi sintak dalam memodelkan sistem secara visual. Sejarah UML sendiri terbagi dalam

dua fase; sebelum dan sesudah munculnya UML. Dalam fase sebelum, UML sebenarnya sudah mulai diperkenalkan sejak tahun 1990an namun notasi yang dikembangkan oleh para ahli analisis dan desain berbeda-beda, sehingga dapat dikatakan belum memiliki standarisasi [21].

UML (*Unified Modeling Language*) memiliki diagram-diagram yang digunakan dalam pembuatan aplikasi berorientasi objek, diantaranya:

2.7.1 Use Case Diagram

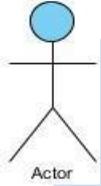
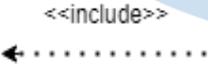
Use Case Diagram adalah salah satu diagram yang membentuk UML (*Unified Modeling Language*), berfungsi untuk mengidentifikasi kasus penggunaan, serangkaian tindakan yang satu sistem dapat berjalan dalam skenario, dan aktor, yang mewakili faktor eksternal (pengguna atau sistem eksternal) yang berinteraksi dengan sistem. Use case mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibangun. [22].

Use case merupakan teknik menangkap kebutuhan-kebutuhan fungsional dari sistem baru atau sistem yang diubah. Setiap *use case* terdiri dari satu atau lebih skenario yang menerangkan bagaimana sistem berinteraksi dengan pengguna atau sistem yang lain untuk mencapai suatu sasaran bisnis tertentu. Dalam teknik ini tidak diterangkan cara kerja sistem secara internal maupun implementasinya. Yang ditunjukkan adalah langkah-langkah yang dilakukan pengguna dalam menggunakan perangkat lunak. [23].

Berikut adalah simbol-simbol yang terdapat pada diagram *use case* [24]:

Tabel 2. 1 *Use Case Diagram*

Simbol	Deskripsi
	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor

<p>Aktor / <i>actor</i></p> 	<p>Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang: biasanya dinyatakan menggunakan kata benda diawal <i>frase</i> nama aktor</p>
<p>Asosiasi / <i>assosiation</i></p> 	<p>Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor</p>
<p>Ekstensi / <i>extend</i></p> 	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dinamakan <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu; mirip dengan prinsip <i>intherince</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>Include</i></p> 	<p>Menspesifikasikan bahwa usecase sumber secara eksplisit</p>
<p><i>System</i></p> 	<p>Menspsifikasikan paket yang menampilkan sistem secara terbatas</p>
<p><i>Generalization</i></p> 	<p>Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku data dari objek yang ada diatasnya objek induk</p>

Sumber: [24]

Pada tabel 2.1 merupakan daftar symbol dan deskripsi setiap simbol yang ada pada *use case diagram* seperti *use case*, aktor, asosiasi, dan *extend*, *include system* dan *generalization*.

2.7.2 Activity Diagram

Activity Diagram yaitu sebuah gambaran aktifitas proses bisnis yang dimiliki perangkat oleh sebuah perangkat lunak. Yang perlu diperhatikan dari *Activity diagram* yaitu menggambarkan aktivitas yang dapat dilakukan oleh sistem, bukan apa yang dilakukan oleh aktor. *Activity Diagram* juga digunakan untuk mendefinisikan atau mengelompokkan aluran tampilan dari sistem [25].

Activity diagram menggambarkan aliran fungsionalitas dalam suatu sistem informasi. Secara lengkap, *activity diagram* mendefinisikan dimana *workflow* dimulai, dimana berakhirnya, aktifitas apa yang terjadi selama *workflow*, dan bagaimana urutan kejadian aktifitas tersebut. *Activity diagram* juga menyediakan pendekatan untuk proses pemodelan paralel. Bagi mereka yang akrab dengan analisis dan desain struktur tradisional, diagram ini menggabungkan ide-ide yang mendasari diagram alir data dan diagram alur sistem [26].

Berikut adalah simbol-simbol yang terdapat pada *activity diagram* [24]:

Tabel 2. 2 *Activity Diagram*

Simbol	Deskripsi
	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal
	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja

Percabangan / <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
Penggabungan / <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir

Sumber: [21]

Pada tabel 2.2 merupakan daftar simbol beserta deskripsi dari setiap simbol yang digunakan pada *activity diagram* diantaranya status awal, aktivitas, percabangan, penggabungan dan status akhir.

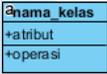
2.7.3 Class Diagram

Diagram kelas UML adalah inti utama dari analisis dan desain OO di mana model lain diturunkan. Terkadang, karena faktor waktu dan biaya yang terlibat dalam rekonstruksi diagram kelas, desainer memodifikasinya secara langsung. Akibatnya, inkonsistensi dapat terjadi dalam kasus ini antara dokumen persyaratan pengguna dan model formal, yang pada gilirannya menyebabkan tidak hanya masalah serius dalam fase pemeliharaan aplikasi, tetapi juga mempengaruhi pembuatan prototipe dan penggunaan kembali perangkat lunak baru dengan persyaratan serupa [27].

Class Diagram adalah salah satu pemodelan yang cukup penting dalam UML, fungsinya adalah untuk membuat sebuah *logical models* dari sebuah sistem. Sebuah *class diagram* akan menunjukkan bagaimana skema dari arsitektur sebuah sistem yang sedang dirancang. *Class diagram* digambarkan dengan *class* yang berisi atribut dan *method*, setiap *class* akan dihubungkan dengan sebuah garis disebut Asosiasi [28].

Class memiliki sesuatu yang disebut atribut dan operasi. Berikut adalah simbol-simbol yang ada pada *class diagram* [24]:

Tabel 2.3 *Class Diagram*

Simbol	Deskripsi
Kelas 	Kelas pada struktur sistem
Antarmuka / <i>Interface</i> 	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek
Asosiasi / <i>Association</i> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
Asosiasi berarah / <i>Directed Association</i> 	Relasi antarkelas dengan makna kelas satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>

Sumber: [24]

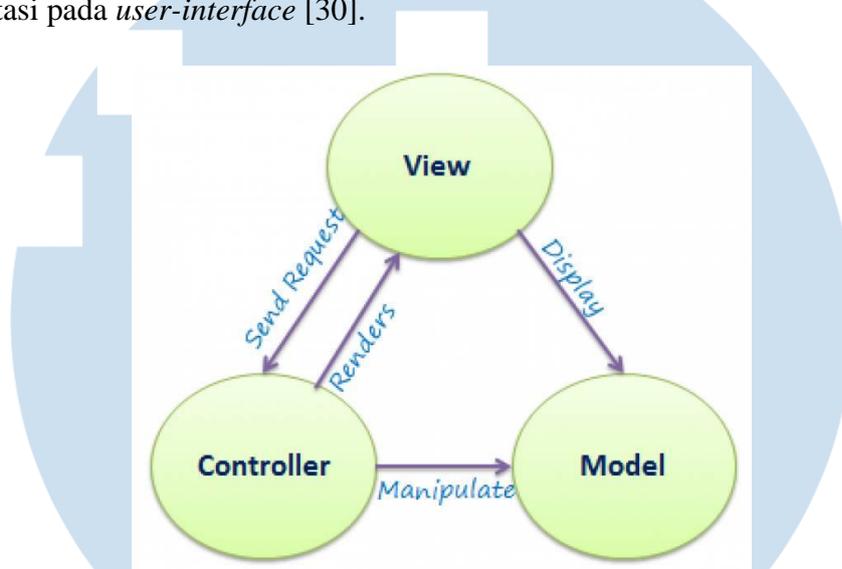
Pada tabel 2.3 menampilkan daftar simbol beserta deskripsi dari setiap simbol yang ada pada *class diagram* diantaranya kelas, antarmuka, asosiasi dan asosiasi berarah.

2.8 CodeIgniter

CodeIgniter adalah sebuah *framework* pengembangan aplikasi dengan menggunakan bahasa pemrograman PHP. *Framework* ini berbasis *website* yang menggunakan PHP lebih sistematis. MVC merupakan konsep dasar dari Codeigniter. MVC merupakan singkatan dari *Model View Controller*. MVC sebenarnya adalah sebuah teknik pemrograman dengan cara memisahkan alur bisnis, penyimpanan data, dan antar muka atau secara sederhana adalah memisahkan antara desain, data dan proses [29].

Codeigniter adalah sebuah *framework* yang bersifat *open-source* yang digunakan untuk membangun aplikasi secara dinamis. CodeIgniter menyediakan banyak library untuk dapat digunakan sehingga dapat mempermudah dalam

pengembangan website. Sebelum mengetahui tentang CodeIgniter, maka harus diketahui konsep dasar tentang MVC. MVC (*Model, View, Controller*) adalah teknik pemrograman yang memisahkan desain, data, dan proses untuk presentasi pada *user-interface* [30].



Gambar 2. 2 Konsep MVC

Gambar 2.2 merupakan konsep MVC diantaranya yaitu *Model, View* dan *Controller* dimana dalam setiap bagian tersebut memiliki fungsi masing – masing untuk presentasi pada *user-interface*.

2.9 Hypertext Preprocessor (PHP)

Bahasa pemrograman PHP merupakan bahasa untuk membuat website yang bersifat server-side scripting. Selain bersifat dinamis PHP juga dapat dijalankan diberbagai sistem operasi seperti Windows, Linux, dan Mac Os. PHP dapat memanfaatkan database untuk menghasilkan halaman web yang dinamis. Sistem manajemen yang biasa digunakan yaitu MySQL, Oracle, dan PostgreSQL. Ketika dipanggil dari web browser, program yang ditulis dengan PHP akan di-parsing di dalam web server oleh interpreter PHP dan diterjemahkan ke dalam dokumen HTML, yang selanjutnya akan ditampilkan kembali web server [31].

Kekuatan yang paling utama dimiliki oleh PHP adalah pada konektivitasnya dengan sistem database di dalam web. Keunggulan lainya dari PHP yaitu mendukung layanan komunikasi protokol yaitu IMAP, SNMP, NNTP dan HTTP. Tujuan utama menggunakan bahasa pemrograman PHP adalah merancang sebuah

website. Program PHP diterjemahkan oleh web server sehingga menghasilkan kode HTML yang dikirim ke browser untuk ditampilkan [32]. Terdapat kelebihan dari PHP, yaitu [33]:

1. Memiliki proses pengembangan yang lebih mudah
2. Termasuk bahasa pemrograman yang memiliki banyak referensi, sehingga mudah dipahami dan diimplementasikan
3. Mudah ditemukan dimanapun seperti pada *website* server Apache, IIS Microsoft Personal Web Server, Xitami dan sebagainya.
4. Dapat dijalankan pada berbagai mesin operasi (Linux, Unix, Macintosh, Windows) serta dapat dijalankan secara runtime dan serta menjalankan perintah-perintah sistem.

2.10 MySQL

MySQL adalah sebuah server untuk database SQL yang populer dan paling banyak digunakan di dunia. Implementasi dari program server database ini adalah program dari 'mysqld' dan beberapa program lainnya. MySql merupakan RDBMS (atau server database) yang dapat mengelola database dengan sangat cepat, dapat menampung jumlah user yang banyak, dan dapat melakukan melakukan suatu proses secara bersamaan(multi-threaded). Saat ini, MySQL banyak digunakan di berbagai kalangan untuk melakukan penyimpanan dan pengolahan data, mulai dari kalangan akademis sampai ke industri kecil, menengah, maupun besar [34].

MySQL merupakan *database* atau DBMS (*Database Management System*) *open-source* yang dapat menyimpan berbagai macam data dalam jumlah besar berbentuk tabel yang saling berhubungan satu sama lain serta dapat digunakan sebagai *client* maupun *server*. Pada perkembangannya, MYSQL disebut juga SQL yang merupakan singkatan dari Structured Query Language. MySql juga merupakan sistem manajemen yang bersifat rasional, data yang dikelola oleh database akan diletakkan pada tabel yang terpisah sehingga untuk memanipulasi data akan lebih cepat [35].

2.11 Black Box Testing

Black Box Testing merupakan bagian integral dari pengujian kebenaran tetapi ide-idenya tidak terbatas pada pengujian kebenaran saja. Pengujian kotak hitam melengkapi teknik pengujian kotak putih dan cenderung mengungkap kelas kesalahan yang berbeda dari metode kotak putih. Penguji, dalam pengujian kotak hitam hanya mengetahui input (proses oleh sistem) dan keluaran yang dibutuhkan, atau dengan kata lain penguji tidak perlu mengetahui kerja internal sistem. Pengujian kotak hitam terjadi sepanjang siklus hidup pengembangan perangkat lunak dan siklus hidup pengujian perangkat lunak yaitu dalam pengujian regresi, pengujian penerimaan, pengujian unit, pengujian integrasi dan tahap pengujian sistem. Jenis pengujian di bawah teknik ini benar-benar terfokus pada pengujian fungsionalitas aplikasi perangkat lunak [36].

Black Box Testing adalah jenis pengujian yang mengabaikan mekanisme internal suatu sistem atau komponen dan hanya berfokus pada output yang dihasilkan sebagai respons terhadap input dan kondisi eksekusi yang dipilih. Ada banyak cara di mana tugas pengujian Black Box dapat dilakukan. Beberapa diantaranya adalah *Boundary Value Analysis*, *Robustness*, *Worst Case*, *Equivalence*, *Cause-Effect* dan *Decision Table Based Testing* [37]. Merupakan pengujian fungsional yang berpusat terhadap *input* dan *output* suatu aplikasi perangkat lunak telah berjalan sesuai dengan apa yang dibutuhkan [38]. *Black box testing* sendiri memiliki kelebihan dan kekurangan. Kelebihannya yaitu:

1. Tidak memerlukan penguasaan yang lebih dalam memahami bahasa pemrograman
2. Sudut pandang dari pengguna digunakan sebagai subjek pengujian
3. Tidak perlu melakukan pemeriksaan kode

Sedangkan kekurangannya yaitu:

1. Kemungkinan terdapat kesalahan tidak dapat terdeteksi karena kurangnya pemahaman secara teknis
2. Tidak melakukan pengujian pada bagian *back-end*
3. Pembuat program memiliki kemungkinan untuk turut melakukan pengujian

2.12 Penelitian Terdahulu

Sebagai bahan referensi dari penelitian ini mengambil dari beberapa penelitian terdahulu berikut penelitian terdahulu yang digunakan sebagai referensi yang digunakan pada penelitian ini.

Tabel 2. 4 Penelitian Terdahulu

Nama Peneliti	Nama Jurnal	Judul Penelitian	Hasil Penelitian
Beben Sutara	Jurnal Informatika: Jurnal Pengembangan IT, 2017	Optimalisasi Aplikasi Transaksi Penjualan Berbasis Web Menggunakan Framework Codeigniter (Studi Kasus: Rumah Makan Tirta Sanding Sumedang)	Hasil penelitian ini adalah pengoptimalan pembukuan menjadi komputerisasi dengan menggunakan framework Codeigniter
Jehan Septia Kurnia, Fitria Risyda	JSI (Jurnal Sistem Informasi), 2021	Rancang Bangun Penerapan Model Prototype Dalam Perancangan Sistem Informasi Pencatatan Persediaan Barang Berbasis Web	Penelitian ini menghasilkan sebuah sistem informasi yang dibangun dengan metode prototype yang menghasilkan aplikasi pengolahan data barang masuk dan keluar
Prof. Doutor Virgílio António Cruz Machado, Prof. Doutor Nuno Manuel Robalo Correia, Prof. Doutora Isabel Maria Nascimento Lopes Nunes	Jurnal Fakultas Sains dan Teknologi Universitas Baru Lisbon, 2013	User-Centered Design of the Interface Prototype of a Business Intelligence Mobile Application	Penelitian ini menghasilkan bagaimana membangun sebuah aplikasi dengan menggunakan prototype sesuai
Wahyu Nugraha, Muhamad Syarif	JUSIM, Vol 3, No 2, Desember 2018	Penerapan Metode Prototype Dalam Perancangan	Penelitian ini menghasilkan sebuah sistem untuk menghitung volume

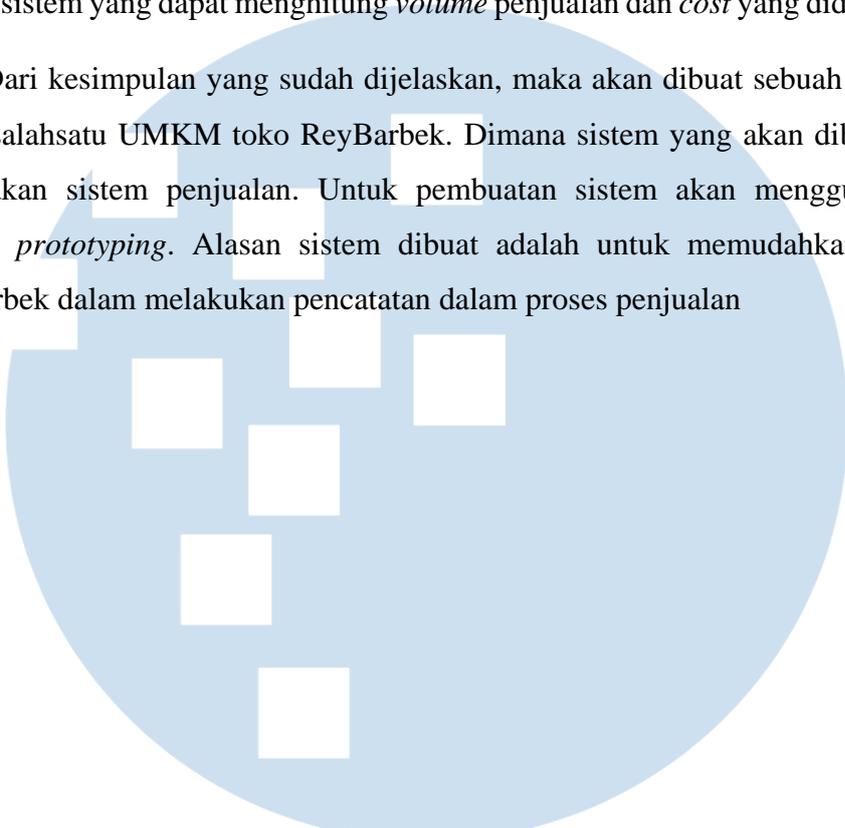
Nama Peneliti	Nama Jurnal	Judul Penelitian	Hasil Penelitian
		Sistem Informasi Perhitungan Volume dan Cost Penjualan Minuman Berbasis Website	penjualan dan cost minuman pada hotel mercure pontianak masih menggunakan data berdasarkan catatan pembukuan.
Agus Heryanto, Hilmi Fuad, Dani Dananggi	Jurnal Sisfotek Global, 2014	Rancang Bangun Sistem Informasi Inventory Barang Berbasis Web Studi Kasus di PT. Infinetworks Global Jakarta	Penelitian ini menghasilkan sistem mencatat dan mengolah data seluruh barang dapat dilakukan dengan semakin mudah.
Ani Oktarini Sari, Elan Nuari	Jurnal PILAR Nusa Mandiri Vol. 13, No. 2 September 2017	Rancang Bangun Sistem Informasi Persediaan Barang Berbasis WEB dengan metode FAST (Framework for The Applications)	Dengan dibuatnya sistem informasi persediaan barang dapat mempermudah pengelolaan barang
Putu Gede Surya Cipta Nugraha, Ni wayan Wardani, I wayan Sukarmayasa	JST (Jurnal Sains dan Teknologi), 2021	Rancang Bangun Sistem Informasi Software Point of Sale dengan Metode Waterfall Berbasis Web	sistem point of sale dapat membantu administrator dan kasir (wirausahawan) dalam mengelola arus bisnis usahanya

2.13 Kesimpulan Terdahulu

Berdasarkan beberapa artikel jurnal yang telah diambil pada penelitian terdahulu diatas, dijadikan referensi tentang bagaimana membangun sistem yang dapat membantu untuk melakukan mencatat barang secara terkomputerisasi [39]. Dijadikan referensi untuk membangun sebuah sistem informasi menggunakan metode *prototype* untuk mengolah barang yang masuk dan barang yang terjual [40], dijadikan referensi membangun sebuah sistem bisnis intelegen dengan

menggunakan metode *prototype* [41], dijadikan referensi agar menghasilkan sebuah sistem yang dapat menghitung *volume* penjualan dan *cost* yang didapat [42].

Dari kesimpulan yang sudah dijelaskan, maka akan dibuat sebuah sistem untuk salahsatu UMKM toko ReyBarbek. Dimana sistem yang akan dibangun merupakan sistem penjualan. Untuk pembuatan sistem akan menggunakan metode *prototyping*. Alasan sistem dibuat adalah untuk memudahkan toko ReyBarbek dalam melakukan pencatatan dalam proses penjualan



UMN

UNIVERSITAS
MULTIMEDIA
NUSANTARA