

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Penelitian Sebelumnya**

Penelitian sebelumnya yang digunakan adalah penelitian yang terkait langsung dengan penelitian-penelitian yang sebelumnya telah dilakukan. Penelitian yang dilakukan akan menggunakan beberapa referensi penelitian sebelumnya yang relevan sebagai acuan yang sesuai. Berikut merupakan beberapa penelitian dengan studi kasus yang relevan dengan penelitian yang dilakukan di antaranya adalah:

Tabel 2. 1 Perbandingan Referensi Penelitian Sebelumnya

No	Judul Penelitian	Masalah	Metode	Hasil
1.	Perancangan Aplikasi Manajemen Persediaan Gudang Berbasis Website Pada Umkm Batik Sinuwun Dengan Agile Scrum Development Method [5]	Semakin banyaknya permintaan dan jenis yang beragam membuat UMKM Batik Sinuwun dalam mengatur persediaan bahan baku produksinya, dikarenakan tidak adanya pencatatan serta manajemen persediaan.	Menerapkan Metode <i>Agile Scrum Development</i>	Dengan adanya Perancangan Aplikasi Manajemen Persediaan Gudang Berbasis <i>Website</i> mempermudah dalam pengelolaan bahan baku masuk dan keluar gudang sehingga dapat diketahui stok bahan baku terkini, selain itu juga dapat meminimalisir kesalahan pencatatan akibat perhitungan manual.
2.	Perancangan dan Implementasi Sistem Website E-Commerce dalam Bisnis Bakery Upaya Meningkatkan Penjualan dan Pemasaran Menggunakan Metode Waterfall pada Pabrik Prima Sari Bakery [6]	Prima Sari Bakery dilakukan secara manual, Selama ini sistem penjualannya masih mengandalkan media interaksi dengan konsumen dengan menggunakan sistem pembukuan faktur, struk bon.	Menerapkan Metode <i>Waterfall</i>	Sebuah sistem <i>website</i> yang dapat diakses oleh <i>User</i> Umum untuk melihat produk dan informasi, <i>User</i> Member atau <i>Customer</i> dapat mengakses proses pemesanan produk roti Prima Sari Bakery, <i>User</i> Admin yang dapat mengolah sistem <i>website</i> terkait datapenjualan, data <i>Customer</i> dan User Pimpinan yang dapat mengakses data laporan penjualan.

No	Judul Penelitian	Masalah	Metode	Hasil
3.	Integrasi Serum dengan Design Thinking dalam Pengembangan Perangkat Lunak pada Startup Awal (Studi Kasus KIOOS)[7]	kegagalan dari startup yaitu produk yang dihasilkan tidak memenuhi kebutuhan pasar, sehingga diperlukan suatu produk yang tepat. Namun dalam proses pengembangan perangkat lunak, untuk menciptakan produk yang tepat memerlukan waktu serta biaya yang besar. Hal tersebut mengakibatkan startup teknologi baru (early-stage startup) yang memiliki keterbatasan dalam resource dan biaya, banyak mengalami kegagalan pada bagian produk.	Menerapkan Metode <i>Scrum</i> .	Hasil dari penelitian ini yaitu sebuah Minimum Viable Product (MVP) yang telah diujicobakan oleh Startup KIOOS. Hasil Pengujian prototype menghasilkan nilai SUS Scorerata-rata sebesar 70.625 yang menandakan prototype dapat diterima oleh market.
4.	Aplikasi Persediaan Barang PT. CAD Solusindo Menggunakan Metode Waterfall[8]	PT. CAD Solusindo masih menggunakan program Microsoft Excel, mulai dari penerimaan barang, pengeluaran barang dan pembuatan laporan yang ditujukan kepada pihak manajemen.	Menerapkan Metode <i>Waterfall</i>	Dengan pemakaian sistem yang telah terkomputerisasi dan berbasis web dapat mengurangi kesalahan-kesalahan yang terjadi dalam penginputan barang. Sistem Informasi persediaan barang yang berbasis web membantu admin/logistik dalam mengontrol stok barang dan persediaan barang. Sistem Informasi inventory yang berbasis web dapat mempermudah dalam membantu aktivitas kegiatan transaksi barang di PT. CAD Solusindo.

No	Judul Penelitian	Masalah	Metode	Hasil
5.	Aplikasi Sistem Informasi Pemantauan Inventory Stock Opname Berbasis Web Pada Pt Makmur Berkat Solusi Logistic	Memiliki beberapa kendala dalam menggunakan sistem yang sudah ada yaitu informasi tidak update dan keterlambatan laporan sehingga pemilik barang tidak dapat melakukan pengecekan laporan stock.	Menerapkan Metode <i>Unified Modelling Language (UML)</i>	Dengan adanya perancangan sistem persediaan laporan stock yang diharapkan dapat menampilkan laporan yang aktual dan mudah dipahami.
6.	Perancangan Sistem Informasi Stok Kontainer Dengan Metode Scrum Pada Pt Putra Batam Jasa Mandiri Utama [10]	Aplikasi yang di gunakan saat ini memiliki masalah inkonsistensi data yang besar sehingga informasi yang di hasilkan tidak efektif.	Menerapkan metode <i>Scrum</i>	Aplikasi yang peneliti kembangkan dengan menggunakan Laravel Framework ini merupakan upgrade dari yang sekarang digunakan oleh PT Putra Batam Jasa Mandiri Utama,

No	Judul Penelitian	Masalah	Metode	Hasil
7.	Development Point of Sales Using SCRUM Framework[11].	<p>MPC Computers adalah retail yang menjual aksesoris komputer. Namun dalam pengoperasiannya MPC Computer masih menerapkan sistem manual, sehingga pelayanan yang diberikan tidak maksimal. Pelaporan yang membutuhkan waktu yang lama membuat pemilik toko kesulitan dalam menentukan persediaan untuk kedepannya karena informasi yang tidak detail sehingga mengakibatkan penurunan kinerja.</p>	Menerapkan Metode <i>Scrum</i>	<p>Aplikasi yang akan disediakan berbagai modul seperti pembelian, penjualan, inventaris dan pelaporan. Tujuannya adalah untuk mengontrol jumlah aset persediaan toko. Sehingga toko dapat meningkatkan pelayanan, mendapatkan informasi yang detail, menghemat waktu, dan menekan biaya.</p>

No	Judul Penelitian	Masalah	Metode	Hasil
8.	Perancangan Aplikasi Android Penjualan Barang Dan Jasa Berbasis E-Commerce Pada Pixels Computer Cilegon Banten [12]	Sistem yang berjalan masih dinilai belum efektif dalam hal pemasaran, konsumen selalu menginginkan kemudahan ataupun kecepatan informasi yang relevan untuk memudahkan segala aktivitasnya, salah satunya pemesanan ataupun pembelian barang dan jasa yang di tawarkan pada Pixels Computer.	Menerapkan Metode <i>Waterfall</i>	Sistem penjualan yang di bangun pada aplikasi pixels computer ini memuat stok produk, promosi, perhitungan, sampai dengan pembayaran. Yang akhirnya dapat mempermudah konsumen dalam melakukan aktifitas pembeliannya. Hal ini menjadikan sistem penjualan yang ada pada pixels computer lebih efisien karena dapat membuat transaksi lebih cepat, pemberian informasi kepada komsumen yang lebih lengkap dan akurat tanpa harus membuat konsumen bertanya kembali tentang stok yang tersedia pada pixels.

No	Judul Penelitian	Masalah	Metode	Hasil
9.	Membangun Aplikasi E-Commerce Diet Sehat Menggunakan Metode Scrum Build E-Commerce Application Healthy Diet Using Scrum Methodology[13]	Kesadaran masyarakat Indonesia akan pentingnya menjaga pola makan sehat masih sangat rendah dan kurangnya informasi mengenai makanan sehat dan penyedia jasa makanan sehat sebagai penunjang diet. Ketersediaan penyedia jasa makanan sehat saat ini belum banyak diketahui oleh masyarakat Indonesia.	Menerapkan Metode Scrum	Dengan menggunakan aplikasi e-Commerce diet sehat berbasis website dapat membantu penyedia dalam berbagi informasi mengenai produk makanan, paket makanan, dan artikel mengenai kesehatan sebagai penunjang diet. Penyedia dapat melakukan pelaporan transaksi serta dapat memperluas pemasaran dengan menggunakan website dan internet sebagai penyedia.
10.	Pengujian Black Box pada Aplikasi Penjualan Berbasis Web Menggunakan Metode Equivalents Partitions (Studi Kasus: PT Arap Store)[14].	Proses validasi data yang belum maksimal dapat mengakibatkan data yang disimpan pada database tidak sesuai, di antara proses loginyang menjadi error. Hal ini dapat merugikan pemilik dan pengguna dalam menggunakan aplikasi ini. Oleh karena itu, diperlukan perbaikan kualitas validasi yang lebih akurat, sehingga dapat membantu pemilik dalam keakuratan data.	Menerapkan Metode Equivalents Partitions	Perangkat lunak ini dapat digunakan dengan baik, setelah dilakukan perbaikan pada kesalahan yang ditemukan. Namun pengujian tersebut dapat dikatakan belum sempurna, karena hanya dilakukan dengan menggunakan beberapa form sampel. Dari hasil pengujian ini diharapkan aplikasi dapat digunakan sesuai kebutuhan pengguna.

## 2.2 Konklusi

- a. Pada penelitian pertama penulis menggunakan satu aktor dalam melakukan prosesnya yaitu *Administrator* dan untuk pengujian sistem menggunakan metode Blackbox Testing serta ISO:9126.
- b. Pada penelitian kedua penulis menggunakan empat aktor dalam melakukan prosesnya, User Umum, User Member atau Costumer, User Admin, User Pimpinan. Untuk pengujian penulis menggunakan black box tasting dan user acceptance test (UAT).
- c. Pada penelitian ketiga penulis menggunakan dua pengujian yaitu Usability Testing dan Framework Pirate Funnel
- d. Pada penelitian ke-empat menggunakan tiga aktor yaitu admin, *user*, karyawan.
- e. Pada penelitian kelima menghasilkan output laporan stok barang yang aktual dan mudah dipahami.
- f. Pada penelitian ke-enam penulis mempermudah dalam mengelola dan mengontrol data persediaan stok kontainer
- g. Pada penelitian ketujuh penulis menggunakan empat aktor yaitu Owner, Cashier, Finance, dan Inventory.
- h. Pada penelitian kedelapan penulis menggunakan dua aktor yaitu *User* dan *Admin* dan menghasilkan aplikasi android personal.
- i. Pada penelitian kesembilan selain memberikan informasi mengenai makanan untuk diet, penulis juga menghasilkan output berupa laporan transaksi produk.
- j. Pada penelitian kesepuluh pengujian dilakukan dengan metode Black Box berbasis Equivalence Partitioning.



## **2.3 Dasar Teori**

### **2.3.1 Sistem**

Asal kata Sistem berasal dari bahasa Latin *systema* dan bahasa Yunani *sustema*. Pengertian sistem adalah suatu kesatuan yang terdiri komponen atau elemen yang di hubungkan bersama untuk memudahkan aliran informasi, materi atau energi untuk mencapai suatu tujuan[15]. Atau dapat di katakan juga pengertian Sistem adalah suatu jaringan kerja yang terdiri dari prosedur prosedur yang saling berhubungan, kemudian berkumpul bersama-sama untuk melakukan atau menyelesaikan kegiatan dan mencapai suatu sasaran tertentu[16].

### **2.3.2 Informasi**

Informasi adalah sebuah data yang telah di lakukan pengolahan menjadi suatu bentuk yang lebih berarti serta berguna bagi penggunaanya dalam pengambilan keputusan baik untuk masa kini atau yang akan datang[17]. Data akan di tangkap kembali sebagai input, di proses kembali melalui suatu model dan seterusnya yang merupakan suatu siklus[18].

### **2.3.3 Sistem Informasi**

Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi bersifat manajerial dan kegiatan strategi-strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan[19]. Penggunaan teknologi informasi pada aktivitas perusahaan merupakan seperti *value chain* dapat menghasilkan beberapa keuntungan, seperti penghematan

biaya, percepatan waktu operasi, peningkatan produktivitas, percepatan waktu pengiriman barang dan jasa kepada pelanggan, serta peningkatan nilai barang dan jasa yang tinggi pada pelanggan[20].

#### **2.3.4 Website**

Pengertian *website* dapat di artikan suatu kumpulan-kumpulan halaman yang menampilkan berbagai macam informasi teks, data, gambar diam maupun bergerak, data animasi, suara, *video* maupun gabungan dari semuanya, baik itu yang bersifat statis maupun yang dinamis, yang dimana membentuk satu rangkaian bangunan yang saling berkaitan dimana masing-masing di hubungkan dengan jaringan halaman atau *hyperlink*[21].

*Website* mengizinkan pemberian *highlight* (penyorotan atau penggaris bawah) pada kata-kata atau gambar dalam sebuah dokumen untuk menghubungkan atau menunjuk ke media lain seperti dokumen, frase, *movie clip*, atau file suara. *Website* dapat menghubungkan dari sembarang tempat dalam sebuah dokumen atau gambar ke sembarang tempat didokumen lain. Dengan sebuah browser yang memiliki *Graphical User Interface* (GUI), link dapat di hubungkan ke tujuannya dengan menunjuk *link* tersebut dengan *mouse* (alat bantu untuk menjalankan kursor pada laptop atau komputer) dan menekannya [14].

### 2.3.5 HTML

*Hypertext Markup Language* adalah bahasa yang di gunakan untuk membuat suatu situs web atau *homepage*. Setiap dokumen dalam web di tulis dalam format HTML. Semua format dokumen, *hyperlink* yang dapat di klik, gambar, dokumen multimedia *form* yang dapat di isi dan sebagainya di dasarkan atas HTML. HTML lebih menekankan pada penggambaran komponen-komponen struktur dan *formatting* di dalam halaman web dari pada menentukan penampilannya, HTML tidak di desain untuk *desktop publishing*, tetapi di desain sebagai bahasa pengkodean untuk *World Wide Web* contoh dari HTML[22]. HTML itu bahasa yang fleksibel karena tidak tergantung pada suatu *platform* (sistem operasi) tertentu. HTML terdiri dari *tag* yang mendefinisikan elemen tertentu pada sebuah halaman *web*[23].

### 2.3.6 JavaScript

*JavaScript* adalah sebuah bahasa pemrograman berbentuk skrip yang dapat di pakai untuk mengakses elemen-elemen di dokumen HTML. Oleh karena itu, di mungkinkan untuk mengendalikan perilaku elemen-elemen HTML[24].

*JavaScript* digunakan pada banyak browser seperti *Internet Explorer, Firefox, Chrome, Opera, Safari* dan lain sebagainya. Hampir seluruh *browser* mendukung *JavaScript* sehingga tidak perlu khawatir kode *JavaScript* yang di gunakan pada *website* tidak berfungsi[11].

### 2.3.7 PHP

*Hypertext Preprocessor* (PHP) adalah bahasa program yang di gunakan untuk membuat aplikasi berbasis web (*website*, blog atau aplikasi web)[24]. PHP termasuk bahasa program yang hanya bisa berjalan di sisi server, atau sering disebut *Side Server Language*. PHP adalah *script* pemrograman yang terletak dan dieksekusi di server. Salah satunya adalah untuk menerima, mengolah, dan menampilkan data dari dan ke sebuah situs. Data akan di olah ke sebuah *database* server untuk kemudian hasilnya di tampilkan di *browser* sebuah situs[23]. Tanpa adanya server web yang terus berjalan dia tidak akan bisa di jalankan[11].

### 2.3.8 XAMPP

XAMPP adalah perangkat lunak bebas, yang mendukung banyak sistem operasi, merupakan kumpulan dari beberapa program. Fungsinya adalah sebagai server yang berdiri sendiri (*localhost*), yang terdiri atas program Apache HTTP Server, MySQL *database*, dan penerjemah bahasa yang di tulis dengan bahasa pemrograman PHP dan Perl[13].

XAMPP berperan sebagai web pada komputer lokal. XAMPP juga dapat disebut sebuah *Cpanel server virtual*, yang dapat membantu melakukan preview sehingga dapat di modifikasi *website* tanpa harus *online* atau terakses dengan internet[21].

### 2.3.9 PhpMyAdmin

PhpMyAdmin adalah sebuah aplikasi/perangkat lunak bebas (*open source*) yang di tulis dalam bahasa pemrograman PHP yang di gunakan untuk menangani Administrasi *database* MySQL melalui jaringan lokal maupun internet[25]. PhpMyAdmin mendukung berbagai operasi MySQL, di antaranya (mengelola basis data, tabel-tabel, bidang (*fields*), relasi (*relations*), indeks, pengguna (*users*), perijinan (*permissions*), dan lain- lain)[17].

### 2.3.10 Framework

#### 1. Bootstrap

*Bootstrap* adalah *framework* CSS yang dapat di gunakan oleh semua *web developer* karena memiliki komponen dasar antarmuka web yang di rancang agar mudah di gunakan. Selain itu, *Bootstrap* juga merupakan *front end* yang bagus karena memperhatikan detail kode sehingga dapat di gunakan di layar yang berbeda. Hal ini memungkinkan pengembang untuk mempercepat pengembangan situs *web* dengan tidak harus membuat ulang tampilan dan nuansa situs *web* mereka saat di lihat di platform dan monitor yang berbeda[26].

#### 2. Codeigniter

*CodeIgniter* merupakan *framework* aplikasi yang akan di bangun di *web* dengan menggunakan konsep *MVC* (*Model, View, Controller*). *Framework PHP* ini dapat menjadi alat bagi pengembang web untuk membuat pengembangan situs web menjadi lebih mudah karena menyediakan sumber daya yang komprehensif[20].

### 2.3.11 Usability Metrics

Usability Metrics adalah sistem atau standar pengukuran yang direpresentasikan dalam satuan yang dapat digunakan untuk menggambarkan lebih dari satu atribut. Usability Metrics sangat berguna untuk mengukur kegunaan selama evaluasi kegunaan perangkat lunak, situs web, dan aplikasi[26].

Sebagian besar Usability Metrics dihitung berdasarkan data yang dikumpulkan selama pengujian kegunaan. Pengguna diminta untuk menyelesaikan tugas sementara peneliti mengamati perilaku pengguna dan membuat catatan. Tugas dapat berupa sesuatu seperti "Bagaimana cara transaksi", atau "Daftar di situs web"[27]. Ada pun untuk penghitungan *Usability Metrics* antara lain:

#### a. *Completion Rate Effectiveness*

*Effectiveness* dapat dihitung dengan menghitung tingkat penyelesaian dengan angka “1” dan jika pengguna tidak berhasil maka nilai yang ditetapkan “0”. Selain itu, dapat dikumpulkan selama setiap tahap perkembangan. Efektivitas dengan demikian dapat direpresentasikan sebagai persentase dengan menggunakan rumus sederhana ini[26]:

$$Effectiveness = \frac{\text{Jumlah tugas yang berhasil diselesaikan}}{\text{Jumlah total tugas yang dilakukan}} \quad (2.1)$$

#### b. *Time-Based Efficiency*

*Time-Based* perhitungan seberapa efisien tugas yang dilakukan oleh pengguna dalam waktu yang dikerjakan. Dengan rumus dibawah ini digunakan untuk menghitung efisiensi waktu.

$$\bar{P}_t = \frac{\sum_{j=1}^R \sum_{i=1}^R \frac{n_{ij}}{t_{ij}}}{NR} \quad (2.2)$$

Keterangan:

$\bar{P}_t$  : *time-based efficiency*

R : Jumlah responden

N : Jumlah total tugas

$n_{ij}$  : Hasil tugas i oleh pengguna j; jika pengguna berhasil menyelesaikan tugas, maka  $n_{ij} = 1$ , jika tidak, maka  $n_{ij} = 0$

$t_{ij}$  : Waktu yang dihabiskan oleh pengguna j untuk menyelesaikan tugas i. Jika tugas tidak berhasil diselesaikan, maka waktu diukur hingga saat pengguna keluar dari tugas.

### 2.3.12 *System Usability Scale*

*System Usability Scale* (SUS) merupakan metode evaluasi kegunaan yang memberikan hasil yang memadai berdasarkan pertimbangan jumlah sampel yang kecil, waktu, dan biaya. Hasil dari perhitungan dengan menggunakan metode SUS akan dikonversi ke dalam sebuah nilai, yang dapat dijadikan pertimbangan untuk menentukan apakah sebuah aplikasi layak atau tidak layak untuk diterapkan[27]. Berdasarkan skor SUS dapat di ketahui tingkat usability dan penerimaan tampilan dan kinerja *website*. Untuk kasus ini nilai SUS yang di dapat dari rata-rata nilai yang di dapat dari responden. Perhitungan nilai rata menggunakan persamaan berikut:

$$\text{nilai rata - rata} = \sum_{i=1}^n \frac{x_i}{n} \quad (2.3)$$

$x_i$  = nilai score responden

N = Jumlah responden

Penilaian berdasarkan 3 (tiga) kategori:

*Not Acceptable*

= skor 0-50,9 (2.4)

*Marginal*

= skor 51-70,9 (2.5)

*Acceptable*

= skor 71-100 (2.6)

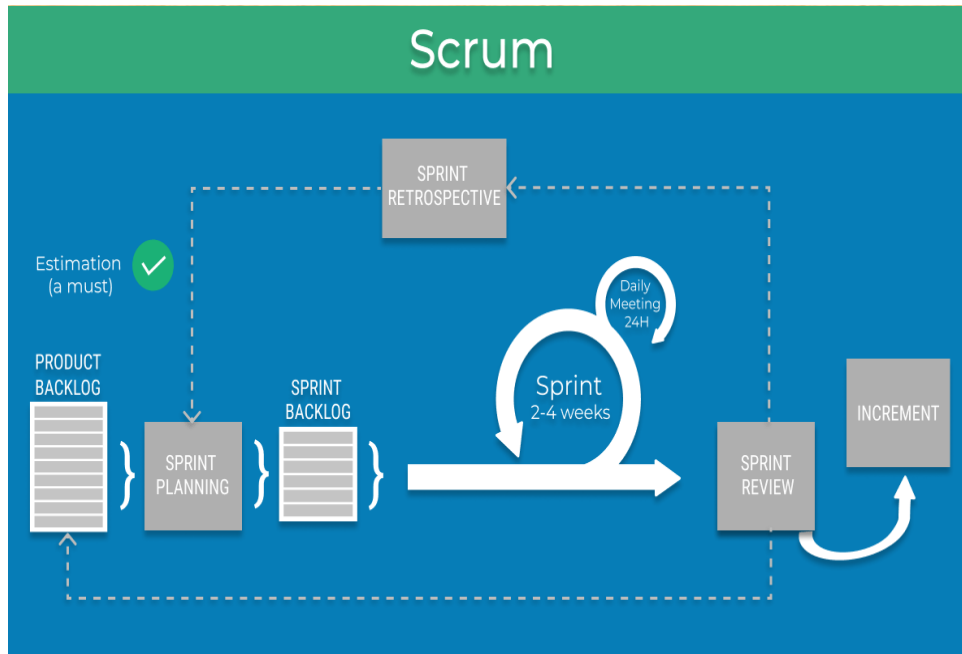
Dalam menentukan hasil perhitungan penilaian terhadap *system usability scale* (SUS) terdapat tiga sudut pandang yaitu *Acceptable*, *Marginal*, dan *Not Acceptable* [28]. Selain dari ketiga cara tersebut, *system usability scale* (SUS) memiliki cara lain dalam melakukan penentuan hasil penelitian yaitu dengan cara *SUS score percentile rank*. Penentuan hasil penilaian berdasarkan *SUS score percentile rank* dilakukan secara umum berdasarkan hasil perhitungan penilaian pengguna, berikut adalah ketentuan penentuan penilaian pada *SUS score percentile rank* [29].

- a) *Grade A* : dengan skor  $\geq 80,3$ .
- b) *Grade B* : dengan skor  $\geq 74$  dan  $< 80,3$ .
- c) *Grade C* : dengan skor  $\geq 68$  dan  $< 74$ .
- d) *Grade D* : dengan skor  $\geq 51$  dan  $< 68$ .
- e) *Grade F* : dengan skor lebih  $< 51$ .



### 2.3.13 SCRUM

*Scrum* adalah kerangka proses yang telah di gunakan untuk mengelola pengembangan produk yang kompleks sejak awal 1990-an. *Scrum* bukanlah sebuah proses, teknik, atau metodologi. Namun, *Scrum* adalah kerangka kerja di mana Anda dapat menggunakan berbagai proses dan teknik di dalamnya[11]. Pada *scrum* terdapat *Scrum Team* kerangka kerja dan tugasnya *scrum* terdapat *Product Owner* adalah orang yang bertanggung jawab untuk memaksimalkan nilai bisnis dari produk yang dihasilkan oleh *Development Team*. *Product Owner* adalah orang yang sangat bertanggung jawab pada pengolahan *Product Backlog*, *Development Team* terdiri dari para ahli profesional yang bekerja untuk memberikan peningkatan Selesai yang berpotensi di rilis pada akhir setiap *Sprint*, *Scrum Master* bertanggung jawab untuk mengenalkan, membantu orang-orang untuk memaksimalkan interaksi guna menghasilkan nilai bisnis yang baik terkait pemahaman teori, praktik-praktik, aturan dan tata nilai dalam *Scrum*[29].



**Gambar 2. 1** Tahapan *Scrum*[29]

Berikut terdapat tahapan-tahapan dari metodologi *scrum*:

### **2.3.10.1 ProductBacklog**

Langkah pertama dari *scrum* adalah penentuan dari fitur atau sistem berdasarkan prioritasnya oleh manager proyek.

### **2.3.10.2 Sprint Planning**

*Sprint Planning* merupakan aktivitas rapat yang di hadiri oleh *Scrum Team* untuk menegosiasikan atau memilah *Product Backlog* apa saja yang akan di usahakan untuk dikerjakan selama *sprint*. *Sprint Planning* memiliki batasan waktu maksimal delapan jam untuk *Sprint* yang berdurasi satu bulan. Untuk *Sprint* yang lebih singkat, acara ini biasanya lebih singkat. *Scrum Master* memastikan acara ini di

selenggarakan dan peserta memahami tujuannya. *Scrum Master* mengedukasi *Scrum Team* untuk menjaganya di dalam batasan waktu[29].

a) Menentukan tujuan *Sprint*

Tujuan dari *Sprint* merupakan bagian penting dalam perencanaan *Sprint*. Tujuan *Sprint* dapat mencegah anggota tim pengembang kebingungan apa yang harus di lakukan[29].

b) Menentukan Panjang *Sprint*

panjang *sprint* atau penetapan tanggal demo adalah salah satu hasil dari *Sprint Planning*. *Sprint* pendek memungkinkan perusahaan menjadi lebih gesit, sering melakukan *delivery*, dan lebih banyak memberikan umpan balik. Namun, sebuah *sprint* yang panjang juga memiliki kelebihan yaitu tim memiliki lebih banyak waktu untuk membangun momentum serta memiliki ruang yang cukup untuk menyelesaikan masalah dan mencapai tujuan *sprint*. Dari dua batasan waktu di atas, menentukan waktu yang tepat untuk *sprint* merupakan hal yang penting untuk di lakukan[29].

c) Melakukan Perkiraan Waktu Pengerjaan Dan Memecah Item *Backlog* Ke Ukuran Yang Lebih Kecil Jika Diperlukan Perkiraan waktu pengerjaan sangat dibutuhkan untuk menyamakan nilai estimasi yang telah dipikirkan oleh pemilik produk pada *product backlog* yang tidak sesuai dengan tim pengembang. Perkiraan

besarnya nilai estimasi akan lebih mudah dan akurat bila di pecah ke dalam bentuk yang lebih kecil[29].

- d) Memutuskan Item *Backlog* yang akan di ikutkan Dalam *Sprint* Salah satu aktivitas utama dalam perencanaan *sprint* adalah menentukan item *Backlog* yang akan di ikutkan dalam *Sprint*[29]

### **2.3.10.3 Sprint Backlog**

Aktivitas ini akan mengubah *product backlog* menjadi *sprint backlog*. Untuk memutuskan item *backlog* yang akan di ikutkan tim dapat menggunakan teknik perhitungan kecepatan yang terdiri dari 2 (dua) tahap yaitu[29]:

1. Memutuskan perkiraan kecepatan

Untuk memutuskan perkiraan kecepatan, pertama yang harus di lakukan adalah menghitung *man-days* yang dimiliki oleh tim pengembang. Cara perhitungannya adalah dengan mengalikan hari kerja dalam *sprint* dengan jumlah tim pengembang Contoh: panjang *Sprint* yang di tetapkan 3 (tiga) minggu atau 18 (delapan belas) hari kerja dengan anggota tim pengembang 3 (tiga) orang, maka di dapatkan perhitungan  $man-days = 18 \times 3 = 54$ . Jumlah *man-days* yang di dapatkan merupakan jumlah kecepatan tim ideal ketika kondisi pengerjaan tidak mendapat gangguan.

Namun hal itu sulit terjadi di karenakan adanya kejadian-kejadian tidak terduga yang di alami tim pengembang, salah satu contohnya adalah saat kita tahu waktu terbagi dengan aktivitas pengembangan lain. Oleh karena itu di butuhkan sebuah *focus factor* di butuhkan untuk menangani masalah ini.

*Focus factor* adalah estimasi tingkat fokus tim. Tingkat *focus factor* yang kecil menunjukkan tim mendapatkan banyak gangguan atau memperkirakan waktu pengerjaan dengan terlalu optimis. Cara terbaik menentukan *focus factor* adalah melihat performa dari sprint sebelumnya.

*Last Sprint's Focus Factor :*

$$Focus\ Factor = \frac{Actual\ Velocity}{available\ Man-Day} \quad (2.7)$$

perkiraan kecepatan dapat di tentukan dengan rumus berikut:

*This Sprint's Estimated Velocity :*

$$Available\ Man-Days \times Focus\ Factor \quad (2.8) \\ = Estimated\ Velocity$$

2. Menghitung banyak item *Backlog* yang akan di tambahkan tanpa melebihi perkiraan kecepatan yang telah di tentukan.

Setelah mendapatkan perkiraan kecepatan tim pada *sprint*, tim akan memilih item *backlog* yang akan di ikutkan

ke dalam sprint selanjutnya dengan pertimbangan pemilik produk. Pertimbangan yang akan dilakukan oleh pemilik produk bersifat tidak wajib diikuti, karena yang bertanggung jawab dalam penentuan item *backlog* dalam *sprint* adalah tim pengembang.

#### **2.3.10.4 Sprint**

Jantung dari *Scrum* adalah *Sprint*, yaitu sebuah batasan waktu dengan durasi satu bulan atau kurang, dimana terdapat proses pembuatan pencapaian yang selesai, dapat digunakan dan berpotensi untuk dirilis. Sebuah inkremen yang selesai di dalam *Sprint* harus berfungsi, berpotensi untuk dirilis dan dikembangkan. *Sprint* biasanya memiliki durasi yang konsisten sepanjang proses pengembangan produk dan dibatasi selama satu bulan menurut kalender. Bila jangka waktu *sprint* terlalu panjang, maka definisi mengenai apa yang akan dibangun akan berubah, kompleksitas dapat meningkat, dan risiko dapat bertambah. *sprint* meningkatkan prediktabilitas karena adanya peninjauan dan pengadaptasian terhadap perkembangan, setidaknya setiap satu bulan sekali. Setiap *sprint* yang dijalankan akan memuat *Scrum Events* yang terdiri dari *Sprint Planning*, *Daily Scrum*, *Sprint Review*, dan *Sprint Retrospective*[29].

#### **2.3.10.5 *Sprint Review***

*Sprint review* di selenggarakan di akhir *Sprint* untuk menginspeksi *Increment* dan mengadaptasi *Product Backlog* biladi perlukan. Pada saat *Sprint Review*, *Scrum Team* dan pemegangkepentingan berkolaborasi untuk meninjau apa yang sudah di selesaikan di *Sprint*. Berdasarkan hasil tinjauan tersebutdan perubahan terhadap *Product Backlog* di dalam *Sprint*, hadirin berkolaborasi untuk menentukan pekerjaan selanjutnya yang dapat dilakukan untuk mengoptimalkan nilai bisnis. Ini adalah pertemuan informal, bukan pertemuan laporan status, dan presentasi *Increment* di lakukan guna mendapatkan umpan balik dan mengembangkan kemampuan kolaborasi[29].

#### **2.3.10.6 *Sprint Retrspective***

*Sprint Retrospective* adalah sebuah kesempatan untuk menginspeksi diri dan membuat perencanaan mengenai peningkatan yang akan dilakukan di *Sprint* selanjutnya. Tujuan dari dilakukannya *Sprint Retrospective* adalah menginspeksi bagaimana jalan *Sprint* terakhir yang terkait dan mengidentifikasi hal utama yang berjalan dengan baik dan peningkatan yang berpotensi untuk dilakukan[29].

#### **2.3.10.7 MVP (*Minimum Valuable Product*)**

*Minimum valuable product* (MVP) merupakan versi dari produk baru yang mampu memberikan hasil maksimum tentang pengetahuan konsumen secara mudah[30]. Sebuah perancangan atau pengembangan produk memiliki perbedaan dengan prototipe, dimana pada MVP, proses yang dilakukan bertujuan untuk mendapatkan pembelajaran secepat mungkin dengan fitur produk yang minimal dan dilakukan pengembangan berulang untuk mendapatkan keseluruhan fungsi pada suatu produk atau sistem[31].

#### **2.3.14 UML**

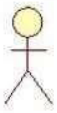



*Unified Modeling Language* (UML) adalah "bahasa" yang telah menjadi standar industri untuk memvisualisasikan, merancang, dan mendokumentasikan sistem perangkat lunak. UML memberikan standar untuk merancang model sistem. Dengan menggunakan UML, Anda dapat membuat model untuk semua jenis aplikasi perangkat lunak, di mana aplikasi tersebut dapat berjalan di perangkat keras, sistem operasi, dan jaringan apa pun, dan dapat ditulis dalam bahasa pemrograman apa pun[32].

Terdapat beberapa diagram UML yang sering digunakan dalam pengembangan maupun pembuatan sebuah sistem, yaitu[32]:



a. Usecase



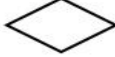



*Use case* diagram menjelaskan manfaat dari aplikasi jika di lihatdari sudut pandang orang yang berada di luar sistem (*aktor*). Diagram ini menunjukkan fungsionalitas suatu sistem atau kelas dan bagaimana sistem berinteraksi dengan dunia luar, berikut merupakan contoh simbolpada *use case*:

Simbol	Nama	Keterangan
	<i>Actor</i>	Menggambarkan manusia atau suatu hal yang menggunakan atau berinteraksi dengan sistem.
	<i>Use Case</i>	Menjelaskan bagian utama dari kegunaan sistem.
	<i>Association Relationship</i>	Sebagai penghubung antara <i>actor</i> dengan <i>use case</i> yang saling berinteraksi.
	<i>Directed Association Relationship</i>	Hubungan asosiasi yang diarahkan hanya kepada satu arah.

**Gambar 2.2** Simbol *Use case* Diagram

b. *Activity Diagram*







Merupakan gambaran alir dari aktivitas-aktivitas di dalam sistem yang berjalan, berikut merupakan contoh simbol dari *activity diagram*:

Simbol	Nama	Keterangan
	Status awal	Sebuah diagram aktivitas memiliki sebuah status awal.
	Aktivitas	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
	Percabangan / Decision	Percabangan dimana ada pilihan aktivitas yang lebih dari satu.
	Penggabungan / Join	Penggabungan dimana yang mana lebih dari satu aktivitas lalu digabungkan jadi satu.
	Status Akhir	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
	Swimlane	Swimlane memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

**Gambar 2.3** Simbol *Activity Diagram*

c. *Sequence Diagram*








*Sequence* diagram menjelaskan interaksi antarobjek di dalam dan di sekitar sistem berupa pesan (*message*) yang di susun dalam suatu urutan waktu yaitu urutan kejadian yang di lakukan oleh seorang aktor dalam menjalankan sistem berikut merupakan contoh simbol dari *sequence* diagram:

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Actor</i>	Menggambarkan orang yang sedang berinteraksi dengan sistem.
2		<i>Entity Class</i>	Menggambarkan hubungan yang akan dilakukan
3		<i>Boundary Class</i>	Menggambarkan sebuah gambaran dari foem
4		<i>Control Class</i>	Menggambarkan penghubung antara boundary dengan tabel
5		<i>A focus of Control &amp; A Life Line</i>	Menggambarkan tempat mulai dan berakhirnya message
6		<i>A message</i>	Menggambarkan Pengiriman Pesan

**Gambar 2.4** Simbol *Sequence Diagram*

d. Class Diagram

Class diagram memberikan gambaran hubungan antara tabel- tabel yang ada dalam *database*, berikut adalah contoh simbol dari *class* diagram:

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Generalization</i>	Hubungan dimana objek anak ( <i>descendent</i> ) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk ( <i>ancestor</i> ).
2		<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3		<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4		<i>Collaboration</i>	<u>Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor</u>
5		<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
6		<i>Dependency</i>	<u>Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan memengaruhi elemen yang bergantung padanya elemen yang tidak mandiri</u>
7		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya

**Gambar 2.5** Simbol *Class* Diagram