

## BAB II KAJIAN LITERATUR

### 2.1 Landasan Teori

#### 2.1.1 Chili Indonesia

Chili Indonesia atau chili.id merupakan startup yang khusus memberikan solusi atas segala permasalahan cabai di Indonesia. Chili Indonesia merancang sistem bisnis pertanian terintegrasi dari hulu hingga hilir khusus komoditas cabai. Dimulai dari *sisi on farm, off farm, food processing* hingga Teknologi Informasi (IT) [1].

Pada sisi hulu, chili.id menerapkan Standar Operasional Prosedure (SOP) dan *Good Agricultural Practices (GAP)*, baik pada sistem pertanian konvensional dan teknologi *greenhouse*. Selain itu, untuk memenuhi kebutuhan pasar maka diterapkan kalender tanam serta pola tanam sehingga lonjakan kebutuhan cabai bisa terjaga [1].

Pada sisi hilir, chili.id memproduksi produk olahan dari cabai. Produk olahan untuk kebutuhan konsumsi hingga obat dan kosmetik. Produk yang akan difokuskan pada tahap awal adalah produksi Sambal beserta produk turunan cabai untuk konsumsi dengan merk Rumah Sambal yang diproduksi sesuai prinsip *Good Manufacturing Practice (GMP)* [1].

Sedangkan pada sisi IT, platform ini bisa digunakan bagi petani, umkm, pembeli, serta *investor*. Platform dapat digunakan sebagai *marketplace*, Kemudian, petani atau penggiat UMKM juga dapat mendapatkan modal usaha dari investor dengan akad salam serta mendapatkan materi-materi pengetahuan dan praktis yang berkaitan dengan cabai pada platform ini [1].

### 2.1.2 E-Commerce

*E-Commerce* atau kependekan dari perdagangan elektronik, merupakan transaksi bisnis yang terjadi dalam jaringan elektronik, seperti internet. Siapapun yang dapat mengakses komputer, memiliki sambungan ke internet, dan memiliki sarana untuk melakukan pembayaran terhadap barang-barang atau jasa yang mereka beli, dapat berpartisipasi dalam *e-commerce* [15].

Sedangkan Kalakota R, dan Whinston, A.B, [16] mendefinisikan *e-commerce* dari beberapa perspektif berikut:

- Perspektif komunikasi (*communications*), Menurut perspektif ini, *e-commerce* merupakan pengiriman informasi, produk/jasa, dan pembayaran melalui lini telepon, jaringan komputer atau sarana elektronik lainnya.
- Perspektif proses bisnis (*business*), Menurut perspektif ini, *e-commerce* merupakan aplikasi teknologi menuju otomatisasi transaksi dan aliran kerja perusahaan (*workflow*).
- Perspektif layanan (*service*), Menurut perspektif ini *e-commerce* merupakan satu alat yang memenuhi keinginan perusahaan, konsumen, dan manajemen dalam memangkas *service cost* ketika meningkatkan mutu barang dan ketepatan pelayanan.
- Perspektif *online* (*online*), Menurut perspektif ini *e-commerce* berkaitan dengan kapasitas jual beli produk dan informasi di internet dan jasa *online* lainnya.

Selain itu, menurut Karmawan [6] *e-commerce* adalah suatu jenis dari mekanisme bisnis secara elektronik yang memfokuskan diri pada transaksi bisnis berbasis individu dengan menggunakan internet (teknologi berbasis jaringan digital) sebagai medium pertukaran barang atau jasa baik antara dua buah institusi (*business to business*) dan konsumen langsung (*business to consumer*), melewati kendala ruang dan waktu yang selama ini merupakan hal-hal yang dominan.

Dengan aplikasi *e-commerce*, hubungan antar perusahaan dengan entitas eksternal lainnya (pemasok, distributor, rekanan, konsumen) dapat dilakukan secara lebih cepat, lebih intensif, dan lebih murah daripada aplikasi prinsip manajemen secara konvensional (*door-to-door, one-to-one relationship*).

#### a) Teknologi yang diperlukan untuk *E-Commerce*

Menurut Turban, dkk [12] pengembangan teknologi *e-commerce* memerlukan satu set keterpaduan aplikasi yang meliputi :

##### 1. Katalog *Online*

Katalog *online* digunakan untuk memberikan informasi kepada pelanggan atau calon pelanggan potensial untuk mendapatkan informasi yang lengkap dari suatu produk.

##### 2. Transaksi *Online*

Transaksi *online* adalah sebuah fasilitas yang disediakan oleh suatu situs *e-commerce* kepada pelanggan untuk dapat melakukan pembelian atau *Order* barang secara *online* lewat media internet, transaksi *online* ini dibagi menjadi dua bagian yaitu:

###### a) *Shopping Cart*

*Shopping cart* software merupakan sistem yang digunakan agar calon pembeli dapat membeli barang-barang yang ditawarkan melalui katalog *online*, mengawasi account kita setiap saat dan menggabungkan semua aspek *e-commerce* pada suatu situs.

###### b) *Online Order*

*Online order* menyediakan layanan pembayaran secara *online* dari transaksi yang sudah terjadi.

### 2.1.3 React

React merupakan sebuah pustaka/library *JavaScript* yang digunakan untuk membangun *user interface* atau UI. Berikut merupakan keunggulan react, diantaranya [7]:

- Deklaratif

*React* membuat proses pembuatan antarmuka pengguna interaktif menjadi lebih mudah. Buat tampilan sederhana untuk setiap *state* di aplikasi, dan React akan secara efisien memperbarui dan *me-render* hanya komponen yang diperlukan ketika data berubah.

Tampilan yang deklaratif membuat jalannya kode menjadi lebih terprediksi dan lebih mudah untuk di-*debug*.

- Berbasis Komponen

Komponen-komponen terenkapsulasi akan mengatur *State*-nya sendiri, kemudian pengembang dapat menggabungkan komponen-komponen tersebut untuk membentuk antarmuka pengguna yang lebih kompleks.

Selain itu, karena logika pada komponen React ditulis langsung menggunakan JavaScript (bukan menggunakan template), maka dapat dengan mudah mengoper data melalui aplikasi dan menempatkan *State* agar tetap berada di luar DOM.

- React juga dapat digunakan untuk *me-render* di server menggunakan Node.
- React memungkinkan untuk dapat berinteraksi dengan *library* dan *framework* lain.

#### 2.1.4 Node.js

Node.js adalah lingkungan atau *platform* untuk mengeksekusi kode-kode yang ditulis dalam JavaScript, yang dikenal dengan sebutan JavaScript *runtime environment*. Dalam melaksanakan tugasnya, Node.js menggunakan V8, yaitu mesin JavaScript yang diproduksi oleh Google. V8 itu sendiri bertugas untuk mengubah kode JavaScript ke dalam bentuk *bytecode* [8].

Aplikasi Node.js dijalankan dalam satu proses tunggal, tanpa membuat *thread* yang baru untuk setiap *request*. Node.js menyediakan satu set perangkat I / O asinkron yang primitif dalam pustaka standarnya yang akan mencegah kode JavaScript dari pemblokiran yang sering terjadi pada bahasa pemrograman sisi server pada umumnya, pustaka di Node.js ditulis dengan menggunakan paradigma *non-blocking* yang menjadikan perilaku pemblokiran sebagai pengecualian [8].

Melalui Node.js, kita dapat menggunakan JavaScript sebagai bahasa pemrograman untuk membuat semua tipe aplikasi, baik yang berupa aplikasi *console* (*command-line interface – CLI*), aplikasi berbasis GUI, aplikasi web, aplikasi *mobile* (Android atau iOS), atau aplikasi untuk sistem kontrol dan akses perangkat keras. Node.js diyakini dapat memproses permintaan-permintaan yang dikirimkan oleh klien (*web browser*) dalam jumlah besar dengan performansi yang baik [8].

Node.js masih menggunakan bahasa JavaScript standar 2015, yang dikenal dengan ECMAScript 6, atau yang sering disingkat ES6. Dalam ECMAScript 6, JavaScript telah mengalami perubahan yang berupa penambahan fitur-fitur baru seperti dukungan terhadap parameter opsional di dalam fungsi, interpolasi variabel di dalam *string*, pembuatan *string* yang terdiri dari beberapa baris, pendefinisian kelas, pewarisan kelas dan lain-lain [8].

### 2.1.5 *Unified Modelling Language (UML)*

Menurut Sri Dharwiyanti [16] *Unified Modelling Language (UML)* adalah sebuah "bahasa" yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem.

Seperti bahasa-bahasa lainnya, UML mendefinisikan notasi dan *syntax*/semantik. Notasi UML merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai diagram piranti lunak. Setiap bentuk memiliki makna tertentu, dan UML *syntax* mendefinisikan bagaimana bentuk-bentuk tersebut dapat dikombinasikan. UML mendefinisikan diagram-diagram sebagai berikut:

#### a) *Usecase Diagram*

*Use case* diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah "apa" yang diperbuat sistem, dan bukan "bagaimana". Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu.

*Use case* diagram dapat sangat membantu bila kita sedang menyusun *requirement* sebuah sistem, mengkomunikasikan rancangan dengan klien, dan merancang test case untuk semua feature yang ada pada sistem.

#### b) *Class Diagram*

*Class* diagram adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi). *Class* diagram menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti containment, pewarisan, asosiasi, dan lain-lain.

**c) Activity Diagram**

*Activity diagram* menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, decision yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity Diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. *Activity Diagram* merupakan *state* diagram khusus, di mana sebagian besar *State* adalah *action* dan sebagian besar transisi di-*trigger* oleh selesainya *State* sebelumnya (*internal processing*). Oleh karena itu *Activity Diagram* tidak menggambarkan behaviour internal sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum. Sebuah aktivitas dapat direalisasikan oleh satu *use case* atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara *use case* menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas.

**d) Sequence Diagram**

*Sequence diagram* menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence* diagram terdiri antar dimensi vertikal (waktu) dan dimensi *horizontal* (objek-objek yang terkait). *Sequence* diagram biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah *event* untuk menghasilkan output tertentu. Diawali dari apa yang men-*trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan *output* apa yang dihasilkan.

### 2.1.6 Model Pengembangan Scrum

**a) Scrum**

Scrum merupakan metode pengembangan perangkat lunak dengan menggunakan prinsip-prinsip pendekatan Agile yang tanggap dalam menangani setiap perubahan dan menekankan pada proses iterasi yang menghasilkan produk berupa *incremental product* [9].

## 1) **Komponen Scrum**

Scrum memiliki tiga peranan penting yaitu *Product Owner*, *Scrum master*, dan *Development Team*, yaitu [11]:

- a) **Product Owner**, seseorang yang memiliki visi, otoritas, dan ketersediaan waktu. Pemilik produk bertanggung jawab untuk terus mengkomunikasikan visi dan prioritas untuk tim pengembangan.
- b) **Scrum Master**, seseorang yang bertindak sebagai fasilitator untuk pemilik produk dan tim pengembangan yang terdiri dari *developer* dan *tester* (*Quality Assurance*). *Scrum master* tidak bertanggung jawab dengan pengelolaan tim. Fungsi dari *scrum master* untuk menghilangkan hambatan yang menghalangi tim dari mencapai tujuannya. Hal ini membantu tim tetap kreatif dan produktif sambil memastikan keberhasilan yang terlihat ke pemilik produk. Scrum Master juga bekerja untuk memberikan nasihat kepada pemilik produk tentang bagaimana memaksimalkan Return On Investment (ROI) untuk tim.
- c) **Scrum Development Team**, menurut pendiri Scrum “*the team is utterly*”. Tim pengembangan bertanggung jawab untuk mengatur diri untuk menyelesaikan pekerjaannya. Sebuah tim pengembangan Scrum memiliki sekitar tujuh anggota yang penuh dedikasi (resminya tiga sampai sembilan orang), idealnya dalam satu ruangan tim dilindungi dari gangguan dari luar. Untuk proyek pengembangan perangkat lunak, tim umum meliputi insinyur perangkat lunak (*software engineers*), arsitek (*architects*), *programmer*, analis (*analyst*), ahli QA (*Quality Assurance*), penguji (*testers*), dan UI designers. Setiap *sprint*, tim bertanggung jawab untuk menentukan bagaimana ia akan menyelesaikan pekerjaan yang harus diselesaikan. Tim memiliki otonomi dan tanggung jawab untuk memenuhi tujuan dari *sprint*.



Scrum juga mengidentifikasi empat objek artefak yang dioperasikan oleh tim Scrum selama siklus pengembangan, yaitu:

a) Product Backlog

1. Menentukan peringkat dari fungsi yang diinginkan.
2. Bisa dilihat dan ditambahkan oleh semua stakeholder (termasuk tim).
3. Pemilik produk menentukan prioritas kerja setiap saat.
4. Item yang berada di paling atas lebih rinci dari yang bawah.
5. Dipertahankan selama Backlog Refinement Meeting.

b) Sprint Backlog

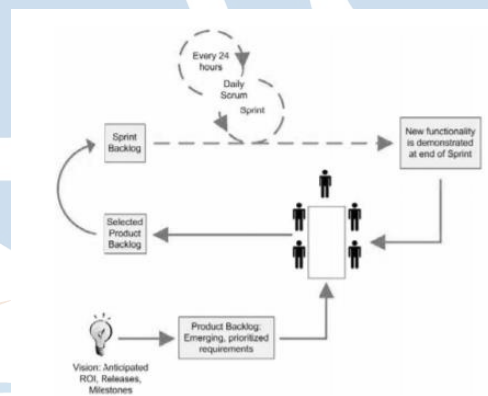
1. Terdiri dari item yang sudah dinegosiasikan antara tim dan pemilik produk.
2. Lingkup komitmen tidak berubah selama Sprint Execution.
3. Semua tugas awal sudah diidentifikasi selama Sprint Planning Meeting.
4. Tim akan menemukan tugas tambahan yang diperlukan untuk memenuhi komitmen yang ada.
5. Semua anggota tim bisa melihat tugas – tugas apa saja yang ada di dalam Sprint Execution.

c) Sprint Task

1. Menentukan bagaimana cara untuk memenuhi / mencapai Product Backlog Item (PBI).
2. Membutuhkan satu hari atau kurang dari satu hari kerja.
3. Sisa dari upaya (waktu) pengerjaan akan diestimasi ulang setiap harinya, biasanya dalam bentuk jam.
4. Selama Sprint Execution, anggota tim bisa menjadi relawan untuk jadi penanggung jawab atas sebuah tugas.
5. Sprint Task menjadi tanggung jawab seluruh anggota tim, jadi semua anggota tim harus bisa berkolaborasi.

Interaksi antara peran-peran menggunakan objek artefak di atas menggunakan beberapa jenis pertemuan pada Scrum [13], yaitu:

- 1) ***Sprint***, kerangka waktu iterasi dengan durasi maksimal satu bulan untuk mengembangkan produk.
- 2) ***Sprint Planning Meeting***, rapat perencanaan *Sprint* yang dilakukan di awal untuk memilih fitur-fitur apa saja yang akan dikerjakan.
- 3) ***Sprint Review***, pertemuan evaluasi pelaksanaan *Sprint* yang dilakukan di akhir *Sprint*. Pada pertemuan ini produk perangkat lunak akan didemonstrasikan kepada *Product Owner*.
- 4) ***Daily Scrum***, pertemuan harian bagi tim pengembang.
- 5) ***Sprint Retrospective***, pertemuan yang dilakukan setelah *Sprint Review* dan sebelum *Sprint Planning* berikutnya untuk kilas balik *Sprint* yang bertujuan mencari hal-hal yang dapat ditingkatkan pada *Sprint* berikutnya. Siklus iterasi Scrum ditampilkan pada gambar



Gambar 1. Scrum Framework [13]

Scrum memperkenalkan konsep *sprint* yang merepresentasikan sebuah iterasi dari siklus pengembangan berbasis waktu dengan durasi selama dua minggu sampai satu bulan. Inti dari Scrum terdiri dari satu set *sprint-sprint* yang menghasilkan perangkat lunak jadi pada setiap akhir *sprint*. *Sprint* adalah kerangka waktu iterasi dengan durasi maksimal satu bulan untuk menghasilkan produk dengan definisi “*Done*” atau “Selesai”, dapat digunakan, dan berpotensi untuk dirilis .

### 2.1.7 *User Story*

*User Story* merupakan salah satu hal utama yang ada pada artifacts scrum. *User story* adalah kebutuhan yang paling tinggi dan berisi informasi mengenai informasi-informasi yang dapat membantu *developer* dalam menjalankan dan mengestimasi waktu pada produk atau proyek yang sedang berlangsung [20].

## 2.2 Pengujian

### 2.2.1 *Black Box Testing*

*Black Box Testing* berfokus pada spesifikasi fungsional dari perangkat lunak. *Tester* dapat mendefinisikan kumpulan kondisi *input* dan melakukan pengetesan pada spesifikasi fungsional program [17]. *Black Box Testing* cenderung untuk menemukan hal-hal berikut:

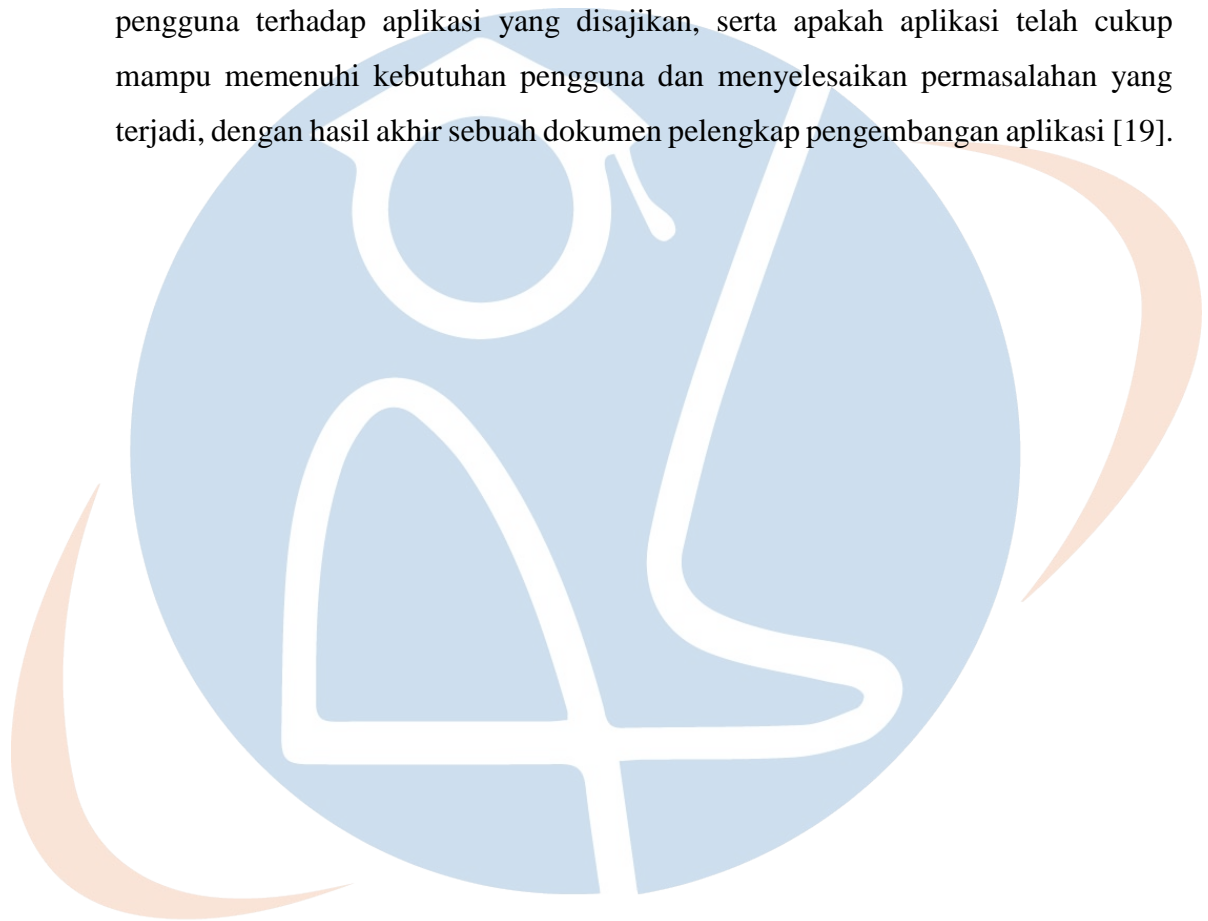
1. Fungsi yang tidak benar atau tidak ada.
2. Kesalahan antarmuka (*interface errors*).
3. Kesalahan pada struktur data dan akses basis data.
4. Kesalahan performansi (*performance errors*).
5. Kesalahan inisialisasi dan terminasi.

### 2.2.2 Skala Likert

Menurut Bilson Simamora [18], Skala Likert disebut juga *summated rating scale*. Skala ini banyak digunakan karena memberi peluang kepada responden untuk mengekspresikan perasaan mereka dalam bentuk persetujuan terhadap suatu pernyataan. Pertanyaan yang diberikan berjenjang, mulai dari tingkat terendah sampai tertinggi. Jumlah pilihan jawabannya bisa tiga, lima, tujuh, sembilan, yang jelas harus ganjil.

### **2.2.3 *User Acceptance Testing***

*User Acceptance Testing* (UAT) merupakan sekumpulan urutan langkah pengujian sebuah aplikasi di sisi pengguna, menggunakan format yang telah disepakati bersama, dengan tujuan untuk mengetahui sejauh mana pemahaman pengguna terhadap aplikasi yang disajikan, serta apakah aplikasi telah cukup mampu memenuhi kebutuhan pengguna dan menyelesaikan permasalahan yang terjadi, dengan hasil akhir sebuah dokumen pelengkap pengembangan aplikasi [19].



STT - NF

### 2.3 Penelitian Terkait

Dalam penyusunan proposal tugas akhir ini, dilakukan studi literatur terhadap penelitian terkait untuk membantu penelitian yang dilakukan. Berikut beberapa penelitian yang berkaitan dengan penelitian yang akan dilakukan, diantaranya:

**Tabel 1.** Penelitian Terkait

No	Nama dan Tahun	Judul	Topik	Tools	Metodologi	Deskripsi dan Hasil
1.	Rahmad Ilham Pratama Institut Pertanian Bogor 2018	Pengembangan Back End Bagian Provider pada <i>Marketplace</i> Travinesia.com dengan Rest Api	Pengembangan Back End Bagian <i>Provider</i> pada <i>Marketplace</i>	Node.Js Database : MongoDB	<i>Scrum</i>	Travinesia.com dikerjakan oleh delapan orang pengembang yaitu pengembangan UI/UX, back end, dan sisi <i>front end</i> website. Pengembangan menggunakan bahasa pemrograman Node.Js dengan framework Express.Js dan basis data MongoDB. Penelitian ini berfokus pada pengembangan <i>back end</i> bagian <i>provider</i> dengan REST API menggunakan metode <i>Scrum</i> . Sistem ini berhasil dikembangkan dengan iterasi sprint sebanyak 4 kali

2.	<p>Utin Kasma, Windy Agasia</p> <p>STMIK Pontianak 2017</p>	<p>Perancangan Sistem E-Grocery Pada Minimarket XYZ Pontianak Menggunakan Scrum Methodology</p>	<p>Merancang Sistem E- Grocery</p>	<p>PHP, Database : MySQL</p>	<p><i>Scrum</i></p>	<p>Penelitian ini menggunakan metode Agile yaitu dengan pendekatan Scrum Methodology. Pengembangan e-grocery diperlukan tiga tim pengembang yaitu, programmer, designer dan tester. Bahasa pemrograman yang digunakan adalah PHP dan database MySQL. Penelitian ini menghasilkan rancangan website e-grocery yang dapat memberikan kemudahan kepada konsumen dalam berbelanja.</p>
3.	<p>Fachry Maulana Prabowo</p> <p>Universitas Lampung Bandar Lampung 2019</p>	<p>Pengembangan Aplikasi Marketplace Pakan Ternak dan Produk Peternak (ePakan.id) Berbasis Android dengan Metode Scrum</p>	<p>Pengembang an Aplikasi Marketplace Pakan Ternak dan Produk Peternak</p>	<p>Android, Laravel API, JSON</p>	<p><i>Scrum</i></p>	<p>Epakan.id dikembangkan menggunakan metode scrum yang ditujukan bagi pengembangan aplikasi dengan skala menengah dan dikerjakan secara tim. Pada jurnal ini peneliti mengerjakan bagian android untuk ePakan. Sistem ini berhasil dikembangkan dengan iterasi sprint sebanyak 5 kali.</p>

4.	Ricky Handoyo, Leo Willyanto Santoso, Alexander Setiawan  Universitas Kristen Petra 2019	Real-Time BPMN Website Menggunakan Teknologi MERN Stack	Pembuatan Real-Time BPMN Website	MongoDB, Express, React, Node (MERN Stack)	-	Pembuatan website real-time bpmn dirancang dengan memanfaatkan MERN stack <i>technology</i> sebagai <i>single page application</i> dan <i>library javascript</i> yang dimanfaatkan dalam pembuatan website terutama <i>library bpmn</i> untuk pembuatan <i>diagram</i> dan socket untuk komunikasi dua arah secara real-time antara user dengan sistem. Sehingga user dapat berbagi project dan membuat diagram secara bersama-sama
5.	Annisa Tahira  Sekolah Tinggi Teknologi Terpadu Nurul Fikri, 2019.	Pengembangan Aplikasi Web <i>E-Commerce</i> Sebagai Media Transaksi Jual-Beli Online (Studi Kasus: chili.id)	Web <i>E-Commerce</i> pada chilli.id	React.Js dan Node.Js  Database server : <i>MongoDB</i>	<i>Scrum</i>	-

## 2.4 Aplikasi Terkait

Dalam penyusunan proposal tugas akhir ini, dilakukan penelitian terhadap *e-commerce* yang sejenis dimana menjual produk cabai pada websitenya. Berikut beberapa *e-commerce* yang ditemukan:

**Tabel 2.** Aplikasi Terkait

No	Nama Website	Kelebihan Web	Kekurangan Web
1	<a href="https://www.agromaret.com/">https://www.agromaret.com/</a>	<ul style="list-style-type: none"><li>• Menjual berbagai macam sayuran</li><li>• Terdapat banyak penjual sayuran yang berbeda-beda</li></ul>	<ul style="list-style-type: none"><li>• Tidak adanya kualitas kontrol dari <i>E-Commercenya</i> terhadap produk yang dijual</li></ul>
2	<a href="https://www.bukalapak.com">https://www.bukalapak.com</a>	<ul style="list-style-type: none"><li>• Menjual berbagai macam barang</li><li>• Tersedia barang-barang yang berbeda yang dapat dibeli pembeli</li></ul>	<ul style="list-style-type: none"><li>• Tidak adanya kualitas kontrol terhadap produk yang dijual</li></ul>
3	<a href="http://www.wildfirechilli.com.au/">http://www.wildfirechilli.com.au/</a>	<ul style="list-style-type: none"><li>• Website khusus untuk jual-beli produk cabai</li></ul>	<ul style="list-style-type: none"><li>• <i>Base market</i> di luar negeri</li></ul>
4	<a href="https://pepperworldhotshop.com/">https://pepperworldhotshop.com/</a>	<ul style="list-style-type: none"><li>• Website khusus untuk jual-beli produk cabai</li></ul>	<ul style="list-style-type: none"><li>• <i>Base market</i> di luar negeri</li></ul>
5	chili.id	<ul style="list-style-type: none"><li>• Website khusus untuk jual-beli produk cabai</li></ul>	<ul style="list-style-type: none"><li>• Sedang tahap pengembangan</li></ul>



## 2.5 Posisi Penelitian

Dalam penyusunan proposal tugas akhir ini, dilakukan penelitian terhadap *e-commerce* yang sejenis dimana menjual produk cabai pada websitenya. Berikut beberapa *e-commerce* yang ditemukan:

**Tabel 3.** Penelitian Terkait

No	SCRUM	E-Commerce	Web	MongoDB	NodeJS	ReactJS
1	<u>Muh Mudrikul Falaq (2014)</u> Pengembangan Back End Bagian Provider pada Marketplace Travinesia.com dengan Rest API					
2	<u>Utin Kasma, Windy Agasia (2017)</u> Perancangan Sistem <i>E-Grocery</i> Pada Minimarket XYZ Pontianak Menggunakan Scrum Methodology					
3	<u>Fachry Maulana Prabowo (2019)</u> Pengembangan Aplikasi <i>Marketplace</i> Pakan Ternak dan Produk Peternak (ePakan.id) Berbasis Android Dengan Metode Scrum					
4	<u>Leo Willyanto Santoso, Alexander Setiawan (2019)</u> <i>Real-Time</i> BPMN Website Menggunakan Teknologi MERN Stack					
5	<u>Annisa Tahira (2019)</u> Pengembangan Aplikasi Web <i>E-Commerce</i> Sebagai Media Transaksi Jual-Beli Online (Studi Kasus: chili.id)					