

## IMPLEMENTASI METODE SCRUM PADA MANAJEMEN PROYEK PENGEMBANGAN *THE DOWNSTREAM SUPPLY CHAIN* (STUDI KASUS : CV. MARVELINDO UTAMA)

Arhans Setiawan Supandji, Toni Anwar

Sistern Informasi, Institut Teknologi Telkom Purwokerto, Indonesia

Jl. DI Panjaitan No.128, Karangreja, Purwokerto Kidul, Indonesia

20103009@ittelkom-pwt.ac.id

### ABSTRAK

CV. Marvelindo Utama, sebagai distributor resmi Smartfren di area Banyumas, Purbalingga, Cilacap, dan Kebumen (Barlingmascakeb), menghadapi masalah dalam sistem proses bisnisnya yang kurang efektif akibat belum adanya sistem manajemen rantai pasokan yang memadai. CV. Marvelindo Utama memerlukan sistem informasi manajemen rantai pasok yang terintegrasi untuk menyelesaikan masalah ini. Manajemen rantai pasok mengatur proses dari bahan baku hingga produk akhir, dan karena CV. Marvelindo Utama tidak memproduksi produknya sendiri, maka termasuk dalam manajemen rantai pasok downstream. Pengembangan sistem informasi manajemen rantai pasok downstream dilakukan dengan metode agile development system menggunakan framework SCRUM, yang berfokus pada manajemen proyek dalam pengembangan perangkat lunak untuk memastikan ketepatan waktu, biaya, serta kepastian pengerjaan proyek. Selain manajemen proyek, perhitungan estimasi biaya pengembangan sistem menggunakan metode use case point (UCP) yang menghitung estimasi usaha dan biaya berdasarkan jumlah dan kompleksitas use case dalam sistem. Hasil perhitungan UCP menunjukkan bahwa diperlukan setidaknya 158 hari kerja untuk mengembangkan sistem informasi manajemen rantai pasok downstream ini. Manajemen proyek yang baik sangat penting, dan penggunaan metode SCRUM efektif dalam melakukan dokumentasi manajemen proyek sistem informasi ini.

**Kata kunci :** *Agile Scrum, Manajemen Proyek, Manajemen Rantai Pasok*

### 1. PENDAHULUAN

Penggunaan kuota internet mobile di Indonesia meningkat signifikan, terbukti dari survei APJII yang menunjukkan kenaikan 8.7% pengguna internet seluler prabayar pada triwulan pertama 2023[1]. CV. Marvelindo Utama, distributor resmi Smartfren di Banyumas, Purbalingga, Cilacap, dan Kebumen (Barlingmascakeb), merasakan dampak peningkatan ini namun menghadapi ketidakefisienan dalam proses bisnis mereka.

Saat ini, penjualan masih menggunakan WhatsApp untuk pemesanan, yang dinilai lambat oleh pimpinan CV. Marvelindo Utama, Bapak Pinky Widya Halim. Proses monitoring dan input data di cluster dan depo masih manual menggunakan Microsoft Excel, menyebabkan masalah ketidaksesuaian stok dan kehilangan barang.

Untuk mengatasi masalah ini, CV. Marvelindo Utama membutuhkan sistem manajemen rantai pasok downstream yang terintegrasi. Sistem berbasis web dan aplikasi mobile e-commerce ini akan membantu memonitor pasokan dan penjualan barang, serta validasi outlet. Sistem informasi ini dikembangkan menggunakan metode Agile dengan framework Scrum untuk memastikan proyek berjalan sesuai jadwal, biaya, dan kebutuhan[2][3].

Scrum dipilih karena adaptif dalam manajemen proyek perangkat lunak[4], sementara perhitungan estimasi usaha dan biaya menggunakan pendekatan Use Case Point (UCP). UCP memberikan estimasi

berdasarkan kompleksitas use case dalam sistem. Penelitian ini memanfaatkan website dan aplikasi[5]

Trello untuk monitoring dan penjadwalan Scrum, serta Microsoft Excel untuk perhitungan UCP. Permasalahan yang menjadi fokus pada penelitian ini adalah belum adanya sistem informasi manajemen rantai pasok downstream yang memadai untuk mendukung pendistribusian produk Smartfren yang dijual oleh CV. Marvelindo Utama. Serta kebutuhan akan dokumentasi manajemen proyek sistem informasi untuk memastikan estimasi biaya, estimasi usaha, ruang lingkup, serta waktu kerja dalam proyek pengembangan sistem informasi tersebut.

Didasari dari rumusan masalah yang ada maka tujuan dari penelitian ini adalah:

- Membuat sistem informasi manajemen rantai pasok *downstream* yang menjawab kebutuhan dari CV. Marvelindo Utama.
- Melakukan dokumentasi dari manajemen proyek pengembangan sistem informasi manajemen rantai pasok *downstream* pada CV. Marvelindo Utama untuk menjawab pentingnya manajemen proyek pada pengembangan sebuah sistem informasi.

Berdasarkan pada rumusan masalah dan tujuan penelitian, maka dapat dijabarkan batasan masalah pada penelitian ini adalah sebagai berikut:

- Titik fokus utama dalam penelitian ini adalah dokumentasi tahapan pengembangan sistem informasi manajemen rantai pasok *downstream* untuk CV. Marvelindo Utama.

- b. Melakukan perhitungan estimasi usaha dan biaya yang dibutuhkan untuk pengembangan sistemnya.
- c. Penelitian ini tidak melakukan pengujian untuk sistem, pengujian dilakukan oleh tim *developer* yang ada pada lampiran penelitian.

## 2. TINJAUAN PUSTAKA

### 2.1. Penelitian terdahulu

Berdasarkan banyaknya penelitian yang ada, ada yang menghasilkan sebuah sistem, menghasilkan rekap dokumen dari perancangan sistem, serta keseluruhan rekap perhitungan estimasi biaya dan usaha dalam pengembangan proyek dari sebuah sistem. Jurnal-jurnal yang dipilih mengangkat tentang penelitian menggunakan metode *agile scrum* dan perhitungan *effort estimation* serta *cost estimation* menggunakan *use case point*.

Jurnal yang diacu diantaranya, “*Effort and Cost Estimation Using Decision Tree Techniques and Story Points in Agile Software Development*” yang ditulis oleh Rodríguez Sánchez et. Al.[6] yang membahas tentang perhitungan *cost* dan *effort estimation* menggunakan metode *decision tree* dalam metode *agile software development*. “*The Challenges of Implementing Agile Scrum in Information System's Project*” yang dituliskan oleh Yusnorizam Ma[4] membahas tentang tantangan penggunaan metode *agile scrum* dalam pengembangan proyek sistem informasi. “*Penggunaan Metode Estimasi Use Case Points (UCP) Dalam Proyek Software Domain Bisnis*” yang ditulis oleh Edhi Prayitno[5] membahas tentang penggunaan metode *use case point* untuk mengestimasi perangkat lunak yang berbasis objek.

### 2.2. Manajemen Rantai Pasok

Manajemen rantai pasok atau Supply Chain Management adalah sebuah perencanaan desain dan pengendalian alir barang serta informasi sepanjang rantai pasokan. Manajemen rantai pasok secara efisien dan berkelanjutan untuk memenuhi kebutuhan pelanggan. Tujuan dari manajemen rantai pasok adalah maksimalnya nilai (value) serta keuntungan (profit) yang didapat dari setiap komponen didalamnya[7]. Manajemen rantai pasok terbagi menjadi tiga jenis, yaitu:

#### a. Upstream Supply Chain

*Upstream Supply Chain* adalah bagian manajemen rantai pasok yang mengatur proses dari bahan mentah hingga ke proses manufaktur atau produksi[8].

#### b. Internal Supply Chain

Proses internal pada manajemen rantai pasok mencakup seluruh tahapan yang ada pada proses produksi. Mengubah masukan dari proses upstream menjadi produk jadi yang nantinya diteruskan ke proses ketiga, yaitu *downstream*[8].

#### c. Downstream Supply Chain

Adalah bagian manajemen rantai pasok yang mengatur proses setelah manufaktur, mengatur tentang distribusi produk hingga ke tangan

pelanggan[8]. Penelitian yang dilakukan akan membuat dokumentasi dari pengembangan sistem informasi manajemen rantai pasok *downstream* untuk CV. Marvelindo Utama.

### 2.3. Manajemen Proyek

Proses pengolahan proyek yang didalamnya terdapat proses perencanaan, pengorganisasian, dan pengaturan tugas-tugas sumber daya, merupakan pengertian dari manajemen proyek. Berfungsi untuk mewujudkan tujuan-tujuan yang ingin dicapai dari sebuah proyek[9].

Manajemen proyek ini dapat diterapkan secara luas didalam proyek yang memiliki skala besar dan kompleks. Fokus utama dari manajemen proyek ini adalah bagaimana bisa mencapai tujuan proyek dengan keterbatasan yang ada, baik dari segi waktu maupun dana yang tersedia[9].

Salah satu metode yang sering digunakan dalam manajemen proyek adalah metode *agile* dengan kerangka kerja *scrum*.

### 2.4. Metode Agile Scrum

*Scrum* merupakan salah satu *framework* dalam metode *Agile* yang dapat menyelesaikan masalah yang kompleks secara adaptif[10].

*Scrum* sendiri adalah sebuah *framework* yang dapat menggunakan berbagai proses dan teknik. Didalam *Scrum* terdapat *role* atau peran dari masing-masing anggota tim, seperti:

- a. *Scrum Master* seorang yang bertugas untuk memelihara lingkungan *Scrum*. Selain itu, *Scrum Master* memiliki tugas untuk menerapkan *scrum* sesuai dengan panduan yang ada. Bertanggung jawab atas efektifitas tim, kerja tim, dan juga kepada *product owner*.
- b. *Product Owner* seseorang yang mengarahkan urutan pekerjaan kedalam *product backlog*. *Product owner* bukanlah komite melainkan hanya satu, tetapi dapat mewakili banyak kebutuhan dari para pemangku kepentingan dalam *Product backlog* yang ada.
- c. *Developers* Adalah anggota tim *scrum* yang bertugas untuk membuat aspek dari increment yang digunakan pada setiap sesi *Sprint*. Membuat rencana *sprint* dan *sprint backlog*.
- d. *Scrum Team* adalah gabungan dari semua di atas. Unit kohesif yang fokus pada satu tujuan, yakni *Product Goal*. Tim ini bersifat lintas fungsi yang berarti setiap anggota didalamnya memiliki kemampuan untuk menciptakan nilai pada setiap *Sprint*. Tim ini cukup kecil, beranggotakan 10 orang atau kurang, hal ini untuk memastikan komunikasi dapat lebih efektif dan terjalin dengan baik.

Adapun istilah-istilah tahapan yang ada di dalam *framework Scrum* antara lain:

#### a. Product backlog

Adalah daftar dari apa saja yang perlu dilakukan saat pengembangan sistem, yang bertugas pada

sektor *backlog* ini adalah *Scrum Master*. *Product backlog* sebelumnya sudah disusun oleh *Product Owner* kemudian disampaikan kepada *Scrum Master* untuk selanjutnya ditugaskan pada *Scrum Team*.

b. *Product Goal*

*Product Goal* secara singkat adalah hasil dari *Product backlog*. Merupakan target yang direncanakan oleh tim *scrum*. Sebagai contoh, apabila dalam *Product backlog* nya adalah pembuatan website, maka *Product Goal* nya adalah sebuah website yang dapat berfungsi sesuai dengan apa yang dideskripsikan pada *Product backlog*.

c. *Sprint*

Adalah kegiatan inti dalam *Scrum* dan mencakup seluruh kegiatan yang ada pada proses *Scrum*. *Sprint* pada umumnya dilakukan selama satu bulan atau kurang, untuk memastikan bahwa target dapat terkejar dengan maksimal dan konsisten. Setelah satu sesi berakhir, maka akan disiapkan untuk memulai *Sprint* yang baru.

d. *Sprint Planning*

Fase awal dari *Sprint*, untuk mengatur pekerjaan yang akan dilakukan dalam satu sesi *Sprint*. Dalam fase ini, *Product Owner* membahas dan memilih *Backlog*, dan akan membahas beberapa hal, seperti seberapa penting *sprint* ini, apa saja yang dapat dilakukan, dan langkah-langkah penyelesaiannya

e. *Sprint Review*

Adalah fase peninjauan dari *sprint* yang sudah dilakukan sebelumnya. Tujuan dari *Sprint Review* adalah untuk mengevaluasi hasil *Sprint* dan menentukan adaptasi di masa depan. Tim *Scrum* membahas kemajuan menuju *Product Goal* dan menyatakan hasil pekerjaan mereka kepada pemangku kepentingan utama

f. *Sprint Retrospective*

Tujuan dari fase ini adalah untuk mengulas apa saja yang terjadi selama *Sprint*. Seluruh tim akan mengidentifikasi apabila terjadi perubahan yang dapat berpengaruh pada *product goal* dan memperbaikinya sesegera mungkin.

## 2.5. Use Case Point Estimation

*Use Case Point Estimation* adalah teknik perhitungan atau estimasi untuk meramalkan ukuran dan kerumitan dari sebuah sistem ataupun biaya dari sebuah sistem yang sedang dikembangkan[11]. Dalam *use case point estimation* variabel yang dihitung antara lain:

a. *Unadjusted Use Case Weight (UUCW)*

UUCW merupakan faktor penting dalam perhitungan *Use Case Point*. Berkontribusi dalam perkiraan besar kecilnya sistem yang sedang dikembangkan, untuk menemukan UUCW dalam sebuah sistem, maka *use case* akan dibagi kedalam tiga kategori yakni, sederhana, rata-rata, dan kompleks dengan memiliki bobot disetiap kategorinya. Berikut contoh perhitungannya:

Tabel 1. UUCW

| Use case | Deskripsi               | Bobot |
|----------|-------------------------|-------|
| Simple   | Kurang dari 3 transaksi | 5     |
| Average  | 4 hingga 7 transaksi    | 10    |
| Complex  | Lebih dari 8 transaksi  | 15    |

b. *Unadjusted Actor Weight (UUCW)*

Proses mengidentifikasi dan mengklasifikasikan jenis aktor dari diagram kasus pengguna disebut sebagai Penimbangan Aktor Tanpa Penyesuaian (UAW).

Tabel 2. UAW

| Aktor   | Bobot | Deskripsi                                  |
|---------|-------|--------------------------------------------|
| Simple  | 1     | Berinteraksi lewat API atau command prompt |
| Average | 2     | Interaksi melalui protocol                 |
| Complex | 3     | Berinteraksi melalui GUI                   |

c. *Unadjusted Use Case Point (UUCP)*

UUCP merupakan ukuran kompleksitas dari software yang hasil perhitungannya didapatkan dari penjumlahan UUCW dan UAW. UUCW merupakan perhitungan total aktivitas, sedangkan UAW merupakan total kompleksitas. Rumusnya adalah

$$UUCP = UUCW + UAW$$

d. *Technical Complexity Factor*

Faktor kompleksitas secara teknis. Menunjukkan faktor apa saja yang memiliki kompleksitas secara teknis didalam sistem. Ini merupakan ukuran untuk tingkat kesulitan dan kompleksitas dari sistem yang sedang dibuat. *Technical Factor* (TF) memiliki nilai antara 0-5 dengan 0 sebagai yang paling tidak relevan, dan 5 adalah sebaliknya. Sedangkan untuk nilai yang netral maka diberi 3.

Tabel 3. TCF

| Faktor | Keterangan                     | Bobot |
|--------|--------------------------------|-------|
| T1     | Sistem terdistribusi           | 2,0   |
| T2     | Waktu respons/tinjauan kinerja | 1,0   |
| T3     | Efisiensi end-user             | 1,0   |
| T4     | Kompleksitas proses internal   | 1,0   |
| T5     | Kemampuan penggunaan kembali   | 1,0   |
| T6     | Kemudahan pemasangan           | 0,5   |
| T7     | Kemudahan penggunaan           | 0,5   |
| T8     | Portabilitas ke sistem lain    | 2,0   |
| T9     | Perbaikan sistem               | 1,0   |
| T10    | Pemrosesan paralel             | 1,0   |
| T11    | Fitur keamanan                 | 1,0   |
| T12    | Akses pihak ketiga             | 1,0   |
| T13    | Pelatihan end-user             | 1,0   |

Rumus untuk TCF adalah:

$$TCF = 0,6 + (0,1 * Total TF)$$

e. *Environmental Complexity Factor*

Hampir mirip seperti TCF, hanya saja ECF digunakan untuk menghitung kompleksitas dari sebuah sistem berdasarkan lingkungannya, dalam hal ini adalah lingkungan pengembang (programmer).

Tabel 4. ECF

| Faktor | Keterangan                           | Bobot |
|--------|--------------------------------------|-------|
| E1     | Keakraban dengan metode pengembangan | 1,5   |
| E2     | Pengalaman pada aplikasi             | 0,5   |
| E3     | Pengalaman tim dalam orientasi objek | 1,0   |
| E4     | Kemampuan analisis utama             | 0,5   |
| E5     | Motivasi Tim                         | 1,0   |
| E6     | Stabilitas Persyaratan               | 2,0   |
| E7     | Staff Paruh Waktu                    | -1,0  |
| E8     | Tingkat kesulitan bahasa pemrograman | -1,0  |

Rumus untuk ECF adalah:

$$ECF = 1,4 + (-0.03 \times EF)$$

f. *Use case point*

Setelah semua faktor di atas sudah dihitung dan ditemukan, maka langkah selanjutnya adalah melakukan perhitungan *use case point* (UCP) dengan rumus sebagai berikut:

$$UCP = UUCP * TCF * ECF$$

g. *Estimasi Usaha*

Langkah selanjutnya dalam perhitungan ini adalah mengubah UCP menjadi nilai *effort*, digunakan dengan menghitung UCP dengan PHM (*Person Hour Multiplier*) yang berupa dua angka pengali yakni 20 dan 28. Rumusnya:

$$F_1 + F_2 <= 2$$

$$PHM = 20$$

$$F_1 + F_2 = 3 \text{ or } 4$$

$$PHM = 28$$

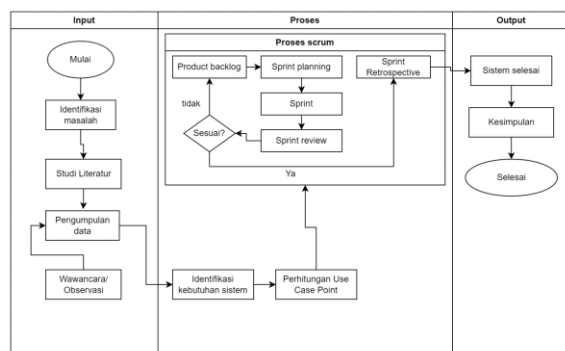
Rumus untuk konversi UCP menjadi effort:

$$Effort = UCP * PHM$$

h. *Perhitungan Biaya*

Apabila sudah diketahui berapa banyak *effort* yang dibutuhkan, maka bisa dilakukan perhitungan biaya dengan cara membagi jumlah *effort* kedalam beberapa bagian pekerjaan. Kemudian dibagi lagi dengan standar gaji yang ada untuk dijumlahkan, sehingga dapat diketahui berapa biaya yang dibutuhkan untuk mengembangkan suatu sistem.

### 3. METODE PENELITIAN



Gambar 1. Diagram Alur Penelitian

### 3.1. Identifikasi Masalah

Identifikasi masalah merupakan langkah pertama yang dilakukan pada penelitian kali ini, tujuan dari identifikasi masalah adalah untuk mengetahui masalah yang dialami. Kemudian, setelah masalah diidentifikasi dan dipahami maka dapat dipikirkan langkah untuk penyelesaiannya. Permasalahan yang ditemui dalam penelitian ini, sesuai dengan hasil wawancara dengan CV. Marvelindo Utama adalah belum adanya sistem informasi manajemen rantai pasok terintegrasi yang dapat memantau alur penjualan produknya.

### 3.2. Studi Literatur

Studi literatur dalam penelitian ini merupakan langkah setelah diketahui apa masalah yang ada. Dalam studi literatur ini dilakukan pembelajaran berdasarkan penelitian terdahulu yang terkait dengan topik dan masalah pada penelitian ini. Seperti studi tentang dasar-dasar teori tentang *Supply Chain Management*, metode *Agile Development*, *framework Scrum*, dan perhitungan estimasi usaha menggunakan *Use case point*. Hal-hal itu digunakan sebagai bukti penyokong dan referensi dalam penyelesaian penelitian ini untuk menjaga relevansinya.

### 3.3. Pengumpulan Data

Langkah pengumpulan data pada penelitian ini diambil dengan menggunakan dua cara, yaitu wawancara dan observasi.

### 3.4. Identifikasi Kebutuhan Sistem

Setelah data terkumpul langkah selanjutnya adalah untuk mengidentifikasi kebutuhan sistem yang ingin dikembangkan. Proses ini berguna untuk mengetahui kebutuhan apa saja yang harus dipenuhi didalam sisitem yang sedang dikembangkan. Hal ini mencakup kemampuan apa saja yang nantinya akan ditampilkan pada sistem, kebutuhan untuk pengembangan sistem, termasuk didalamnya ada biaya yang dibutuhkan untuk pengembangan sistemnya.

### 3.5. Perhitungan Usaha dengan Use Case Point

Selanjutnya, setelah mengetahui apa saja kebutuhan sistemnya maka akan dilakukan perhitungan untuk estimasi usaha. Fase ini menghitung perkiraan biaya, dan waktu pengerjaannya. Hasil perhitungan kemudian akan dicocokkan dengan data biaya yang diberikan oleh CV. Marvelindo Utama, apabila biaya sudah sesuai maka bisa dilakukan pengerjaan untuk pengembangan sistemnya.

### 3.6. Proses Framework Scrum

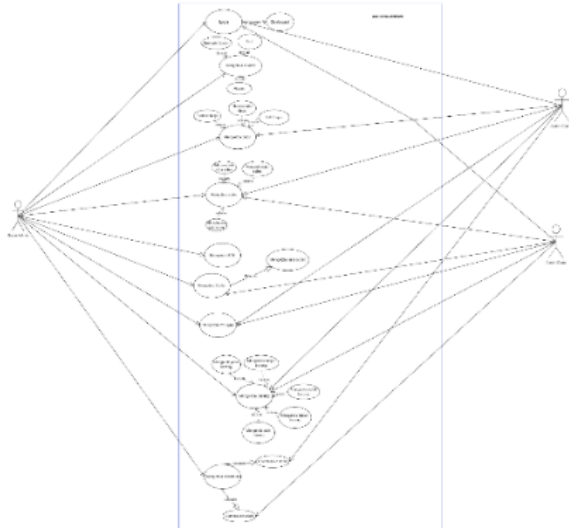
Apabila estimasi perhitungan biaya sudah selesai dilakukan dan sesuai dengan apa yang diinginkan, maka fase selanjutnya adalah implementasi *framework Scrum* untuk mulai mengembangkan sistemnya.

#### 4. HASIL DAN PEMBAHASAN

##### 4.1. Identifikasi Kebutuhan Sistem

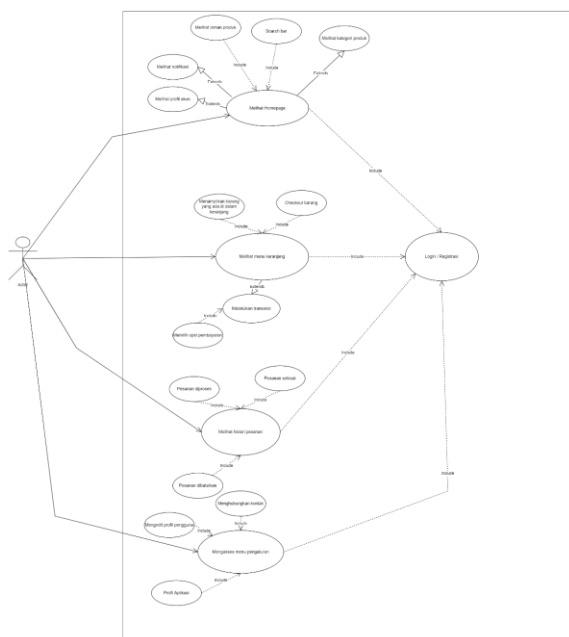
Berdasarkan hasil pengumpulan data yang sudah dilakukan, dapat disimpulkan bahwa yang dibutuhkan oleh CV. Marvelindo adalah sistem manajemen rantai pasok *downstream* berupa *website* dan aplikasi *mobile*. Sistem tersebut nantinya akan memuat detail alur perpindahan barang dari *cluster* serta depo CV. Marvelindo Utama kepada sales dan akhirnya kepada *outlet*.

##### 4.2. Perhitungan Usaha Dengan Use Case Point



Gambar 2. Use Case Website

Gambar 2 menampilkan *use case* untuk *website* yang dikembangkan untuk CV. Marvelindo Utama. Ada 3 aktor pada sistem *website*-nya masing-masing aktor memiliki hak akses yang berbeda tergantung pada peran masing-masing aktornya.



Gambar 3. Use Case Mobile

Gambar 3 menampilkan *use case* untuk aplikasi yang dikembangkan untuk CV. Marvelindo Utama. Ada 1 aktor yaitu *outlet* yang nantinya akan berperan sebagai *customer* yang memesan barang dari CV. Marvelindo Utama.

##### 4.3. UUCW

Tabel 5. UUCW Website

| NO    | Klasifikasi Use Case | Bobot | Jumlah | UUCW |
|-------|----------------------|-------|--------|------|
| 1.    | Sederhana            | 5     | 2      | 10   |
| 2.    | Rata-rata            | 10    | 6      | 60   |
| 3.    | Kompleks             | 15    | 0      | 0    |
| Total |                      |       |        | 70   |

Tabel 6. UUCW Mobile

| NO    | Klasifikasi Use Case | Bobot | Jumlah | UUCW |
|-------|----------------------|-------|--------|------|
| 1.    | Sederhana            | 5     | 3      | 15   |
| 2.    | Rata-rata            | 10    | 8      | 80   |
| 3.    | Kompleks             | 15    | 5      | 75   |
| Total |                      |       |        | 170  |

Hasil perhitungan UUCW untuk *website* adalah 170 dan 70 untuk UUCW aplikasi *mobile*. Perhitungan UUCW dilakukan dengan mengalikan bobot klasifikasi *use case* yang ada dengan jumlah *use case* nya. Hasil perhitungan dapat dilihat pada tabel 5 dan 6.

##### 4.4. UAW

Tabel 7. UAW Website

| Tipe Aktor | Bobot | Jumlah aktor | Total |
|------------|-------|--------------|-------|
| Sederhana  | 1     | 0            | 0     |
| Rata-rata  | 2     | 0            | 0     |
| Kompleks   | 3     | 2            | 6     |
| UAW        |       |              | 6     |

Tabel 8. UAW Mobile

| Tipe Aktor | Bobot | Jumlah aktor | Total |
|------------|-------|--------------|-------|
| Sederhana  | 1     | 0            | 0     |
| Rata-rata  | 2     | 0            | 0     |
| Kompleks   | 3     | 1            | 3     |
| UAW        |       |              | 3     |

Tabel 7 dan 8 menunjukkan perhitungan UAW dari *website* dan aplikasi *mobile* dari CV. Marvelindo Utama. Perhitungan dilakukan dengan mengalikan bobot dari klasifikasi tipe aktor dengan jumlah aktornya. Hasil perhitungannya untuk *website* UAW nya 6 untuk *mobile* UAW nya adalah 3.

##### 4.5. UUCP

Tabel 9. UUCP

| Nama Sistem | UUCW | UAW | UUCP |
|-------------|------|-----|------|
| Website     | 170  | 6   | 176  |
| Aplikasi    | 70   | 3   | 73   |

Tabel 9 menampilkan hasil perhitungan UUCP dari *website* dan aplikasi *mobile* CV. Marvelindo Utama. Perhitungan ini dilakukan dengan

menjumlahkan UUCW dan UAW yang sebelumnya sudah dihitung.

#### 4.6. TCF

TCF atau *Technical Complexity Factor* adalah indikator perhitungan kompleksitas sistem berdasarkan faktor teknis, perhitungan ini dilakukan oleh *project manager* yang berkoordinasi dengan *product owner* untuk mengisi nilainya, nilai akan dikalikan dengan bobot yang sudah ditentukan sebelumnya. Tabel 10 menampilkan detail dari TCF untuk *website* dan *mobile*. Rumus perhitungan dapat dilihat di bawah tabel.

Tabel 10. TCF

| Faktor                 | Pertanyaan                                                                              | Bobot | Nilai | Total |
|------------------------|-----------------------------------------------------------------------------------------|-------|-------|-------|
| T1                     | Apakah sistem bisa terdistribusi dengan baik?                                           | 2,0   | 4     | 8     |
| T2                     | Apakah sistem memiliki waktu respons yang baik?                                         | 1,0   | 4     | 4     |
| T3                     | Apakah sistem sudah memenuhi efisiensi end-user?                                        | 1,0   | 3     | 3     |
| T4                     | Apakah sistem memiliki kompleksitas pemrosesan internal?                                | 1,0   | 2     | 2     |
| T5                     | Apakah sistem dapat digunakan kembali dengan mudah?                                     | 1,0   | 5     | 5     |
| T6                     | Apakah sistem dapat dipasang dengan mudah?                                              | 0,5   | 5     | 2,5   |
| T7                     | Apakah sistem dapat digunakan dengan mudah?                                             | 0,5   | 4     | 2     |
| T8                     | Apakah sistem memiliki portabilitas dengan sistem lain?                                 | 2,0   | 4     | 8     |
| T9                     | Apakah akan ada perbaikan sistem kedepannya?                                            | 1,0   | 4     | 4     |
| T10                    | Bagaimana sistem mengelola pemrosesan parallel?                                         | 1,0   | 3     | 3     |
| T11                    | Apakah sistem memiliki fitur keamanan yang memadai?                                     | 1,0   | 4     | 4     |
| T12                    | Apakah sistem memungkinkan untuk adanya akses dari pihak ketiga?                        | 1,0   | 3     | 3     |
| T13                    | Apakah harus dilaksanakan pelatihan khusus bagi end-user sebelum menggunakan sistemnya? | 1,0   | 1     | 1     |
| Total Technical Factor |                                                                                         |       |       | 45,5  |

$$TCF = 0,6 + (0,1 * 45,5) = 1,055$$

TCF untuk kedua sistem baik *website* atau aplikasi *mobile* adalah 1,055.

#### 4.7. ECF

ECF atau *Environmental Complexity Factor* adalah indikator perhitungan kompleksitas sistem berdasarkan faktor lingkungan yang dilihat dari kompetensi anggota tim dalam mengeksekusi bahasa pemrograman serta motivasi dalam pengerjaan proyek. Tabel 11 menampilkan detail perhitungan dari ECF untuk *website* dan aplikasi *mobile* untuk CV. Marvelindo Utama, perhitungan dilakukan dengan mengalikan bobot dan nilai yang sudah diisi oleh *project manager* untuk mendapatkan total ECF. Rumus perhitungan dan hasil dapat dilihat di bawah tabel.

Tabel 11. ECF

| Faktor                     | Pertanyaan                                                                      | Bobot | Nilai | Total |
|----------------------------|---------------------------------------------------------------------------------|-------|-------|-------|
| E1                         | Apakah familiar dengan metode agile?                                            | 1,5   | 5     | 7,5   |
| E2                         | Apakah pengembang memiliki pengalaman pada aplikasi ini?                        | 0,5   | 3     | 1,5   |
| E3                         | Apakah pengembang berpengalaman dalam pengembangan object-oriented programming? | 1,0   | 3     | 3     |
| E4                         | Apakah pengembang memiliki kemampuan analisis yang baik?                        | 0,5   | 5     | 2,5   |
| E5                         | Adakah motivasi bekerja dalam tim?                                              | 1,0   | 5     | 5     |
| E6                         | Apakah pengembang memenuhi stabilitas persyaratan?                              | 2,0   | 5     | 10    |
| E7                         | Apakah staff pengembang adalah paruh waktu?                                     | -1,0  | 4     | -4    |
| E8                         | Apakah ada kesulitan dalam memahami bahasa pemrograman yang digunakan?          | -1,0  | 3     | -3    |
| Total Environmental Factor |                                                                                 |       |       | 22,5  |

$$ECF = 1,4 + (-0,03 \times 22,5) = 0,725$$

ECF untuk kedua sistem adalah 0,725.

#### 4.8. UCP

Tabel 12. UCP

| UUCP | TCF   | ECF   | UCP     |
|------|-------|-------|---------|
| 176  | 1,055 | 0,725 | 134,618 |
| 73   | 1,055 | 0,725 | 55,835  |

Tabel 12 menampilkan perhitungan *use case point* (UCP) dari *website* dan aplikasi *mobile* CV. Marvelindo Utama, perhitungan ini dilakukan dengan mengalikan UUCP dengan TCF dan ECF. UCP untuk *website* adalah 134,618 dan untuk *mobile* adalah 55,835.

#### 4.9. Perhitungan Estimasi Usaha

Tabel 13. *Effort Estimation*

| Sistem          | UCP     | PHM | Effort (Hour) |
|-----------------|---------|-----|---------------|
| Website         | 134,618 | 20  | 2.692,360     |
| Aplikasi Mobile | 55,835  | 20  | 1.116,700     |
| Combined Effort |         |     | 3.809,060     |

Tabel 13 menampilkan hasil perhitungan *effort estimation* dengan mengalikan UCP dengan PHM atau *person hour multiplier* penjelasan tentang PHM dapat dilihat di Bab 2. Diketahui *effort* untuk pengembangan *website* dan aplikasi *mobile* untuk CV. Marvelindo Utama adalah 3.809,060 jam atau setara dengan 158 Hari Kerja.

#### 4.10. Estimasi Biaya

Tabel 14. *Effort Estimation*

| Phases/<br>Activities                     | Effort<br>Distribution<br>Percentage | Effort<br>Distribution | Cost<br>Estimation<br>(Rupiah) |
|-------------------------------------------|--------------------------------------|------------------------|--------------------------------|
| <b>Software<br/>Phase<br/>Development</b> | <b>72,6</b>                          | 2765,44                |                                |
| Need Analysis<br>(Requirement)            | 1,6                                  | 60,9                   | 1.484.437                      |
| Specification                             | 7,5                                  | 285,6                  | 6.961.500                      |
| Design                                    | 6,0                                  | 228,5                  | 4.641.292                      |
| Implementation                            | 52,0                                 | 1980,6                 | 40.225.884                     |
| Acceptance<br>and<br>installation         | 5,5                                  | 209,4                  | 5.104.125                      |
| <b>Ongoing life-<br/>cycle activity</b>   | <b>17,5</b>                          | 666,5                  |                                |
| Project<br>management                     | 3,8                                  | 144,7                  | 3.527.062                      |
| Configuration<br>management               | 4,3                                  | 163,7                  | 3.325.074                      |
| Documentation                             | 8,4                                  | 319,9                  | 7.797.562                      |
| Training &<br>technical<br>support        | 1,0                                  | 38,09                  | 1.702.127                      |
| <b>Quality and<br/>testing phase</b>      | <b>9,9</b>                           | 377,09                 |                                |
| Integrated<br>testing                     | 7,0                                  | 266,63                 | 5.415.788                      |
| Evaluation<br>and Testing                 | 2,0                                  | 76,18                  | 1.547.368                      |
| Total of Effort                           | 100                                  | 3809                   | 80.030.092                     |

Total estimasi biaya didapatkan dengan membagi *effort* ke dalam beberapa tahapan, untuk kemudian dikalikan dengan *payrate per hour*. Tabel 14 menampilkan rincian dari *effort estimation*-nya. Total estimasi biaya yang diperlukan dalam pengembangan sistem ini adalah Rp. 80.030.092.

#### 4.11. Product Backlog

Berikut adalah *backlog* yang harus dikerjakan oleh *scrum team* dalam proyek ini.

Tabel 15. *Backlog Website*

| NO  | Backlog Item                     |
|-----|----------------------------------|
| 1.  | Membuat menu <i>login</i>        |
| 2.  | Membuat menu <i>dashboard</i> .  |
| 3.  | Membuat menu petugas             |
| 4.  | Membuat menu <i>cluster</i>      |
| 5.  | Membuat menu depo                |
| 6.  | Membuat menu barang              |
| 7.  | Membuat menu barang masuk        |
| 8.  | Membuat menu harga barang        |
| 9.  | Membuat menu histori barang      |
| 10. | Membuat menu jenis barang        |
| 11. | Membuat menu data <i>outlet</i>  |
| 12. | Membuat menu jenis <i>outlet</i> |
| 13. | Membuat menu <i>sales</i>        |
| 14. | Membuat menu data BTS            |
| 15. | Membuat menu detail barang       |
| 16. | Membuat menu depo                |
| 17. | Membuat menu <i>sales</i>        |
| 18. | Membuat menu transaksi barang    |

Tabel 15 menampilkan daftar *product backlog* yang harus dikerjakan oleh developer *website*. Terdapat 18 backlog yang nantinya akan dibagi kedalam 6 sesi *sprint*. *Product Backlog* berisikan hal apa saja yang harus dikembangkan.

Tabel 16. *Backlog Mobile*

| NO | Backlog Item                       |
|----|------------------------------------|
| 1. | Membuat menu <i>login register</i> |
| 2. | Membuat menu beranda               |
| 3. | Membuat menu etalase               |
| 4. | Membuat menu tambah barang         |
| 5. | Membuat menu keranjang             |
| 6. | Membuat menu <i>checkout</i>       |
| 7. | Membuat menu order                 |
| 8. | Membuat menu transaksi             |
| 9. | Membuat menu opsi pembayaran       |

Tabel 16 menampilkan daftar *product backlog* yang harus dikerjakan oleh developer *mobile*. Terdapat 9 backlog yang nantinya akan dibagi kedalam 6 sesi *sprint*. *Product Backlog* berisikan hal apa saja yang harus dikembangkan.

#### 4.12. Sprint Planning

*Sprint Planning* dilakukan setelah *backlog* dibuat dan disetujui oleh *product owner* dan *scrum master*. *Scrum master* selanjutnya akan memanggil *scrum team* untuk berdiskusi membahas tentang rencana *sprint* yang nantinya akan dilakukan. Berdasarkan hasil diskusi bersama yang dilakukan oleh *product owner*, *scrum master*, dan *scrum team* maka diputuskan bahwa dalam proyek pengembangan sistem informasi manajemen rantai pasok *downstream* CV. Marvelindo Utama akan memiliki 6 *sprint*, dimana masing-masing *sprint* akan berdurasi selama 1 bulan, dengan dilakukan *sprint review* pada setiap minggunya.

Tabel 17. *Sprint Planning*

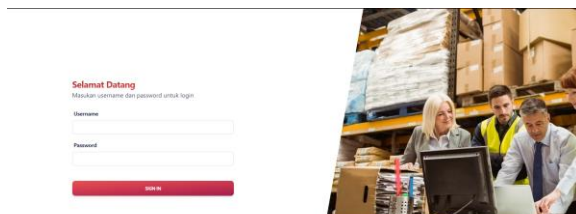
| ID  | Name            | Expected Time                  |
|-----|-----------------|--------------------------------|
| SP1 | <i>Sprint 1</i> | <i>1 Month (Des 23-Jan 24)</i> |
| SP2 | <i>Sprint 2</i> | <i>1 Month (Jan 24-Feb 24)</i> |
| SP3 | <i>Sprint 3</i> | <i>1 Month (Feb 24-Mar 24)</i> |
| SP4 | <i>Sprint 4</i> | <i>1 Month (Mar 24-Apr 24)</i> |
| SP5 | <i>Sprint 5</i> | <i>1 Month (Apr 24-May 24)</i> |
| SP6 | <i>Sprint 6</i> | <i>1 Month (May 24-Jun24)</i>  |

#### 4.13. *Sprint*

*Sprint* dilaksanakan setelah *sprint planning* dibuat. Proses pada *sprint* adalah mengerjakan *product backlog* untuk mewujudkan *product goals*.

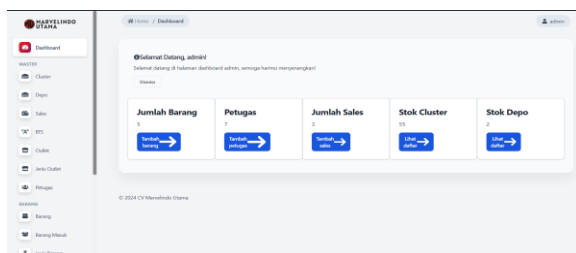
#### 4.14. Hasil

Berikut adalah tampilan hasil dari pengembangan sistem informasi manajemen rantai pasok *downstream* untuk CV. Marvelindo Utama.



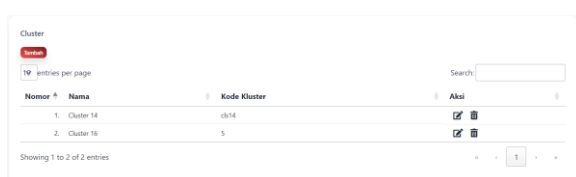
Gambar 4. Tampilan Login

Gambar 4 menampilkan halaman login, *user* akan diminta untuk mengisi data *login* berupa *username* dan *password*. Setelah berhasil *login user* akan dialihkan langsung ke menu *dashboard*. Apabila *user* tidak memasukkan data yang valid maka akan dikembalikan ke menu awal untuk mengisi data yang sesuai.

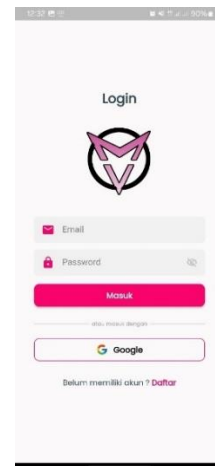


Gambar 5. Tampilan Dashboard

Gambar 5 menampilkan tampilan *dashboard* yang merupakan tampilan awal setelah *login*. *Dashboard* menampilkan fitur-fitur yang ada pada *website* yang dapat diakses melalui *side-menu* yang ada.

Gambar 6. Tampilan Menu *Cluster*

Gambar 6 menunjukkan salah satu menu yang ada, yaitu menu *cluster*. Fitur pada *cluster* mewakili menu lain yang ada. Pada menu *cluster* terdapat fitur CRUD yang ada pada menu lainnya.

Gambar 7. Menu Login *Mobile*

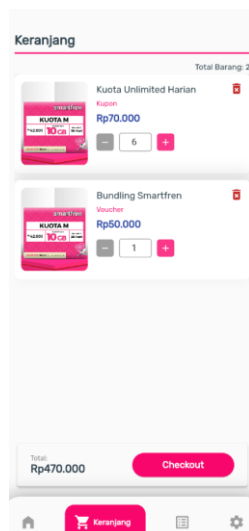
Gambar 7 menampilkan halaman login untuk aplikasi *mobile* CV. Marvelindo Utama. *User* akan diminta untuk memasukkan info *login* berupa *email* dan *password*, ada juga opsi untuk *login* melalui *Google Authentication* dengan menghubungkan akun *Google* milik *user*. Apabila *user* tidak atau belum memiliki akun maka diharapkan untuk melakukan *register*.



Gambar 8. Menu Beranda dan Etalase

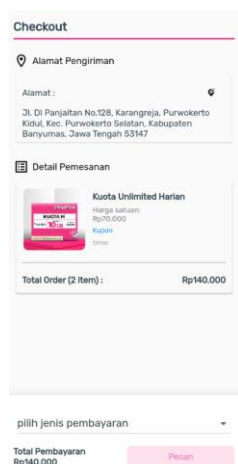
Gambar 8 menampilkan menu beranda dan etalase pada aplikasi *mobile* CV. Marvelindo Utama. Beranda berisikan daftar produk yang sebelumnya ditambahkan di etalase. Ada juga opsi untuk melakukan *searching* dan melihat produk berdasarkan kategorinya.





Gambar 9. Menu Keranjang, Tambah Barang, Checkout

Gambar 9 menampilkan menu keranjang, dimana barang yang sudah ditambahkan sebelumnya akan masuk kedalam keranjang. Ada juga opsi untuk melakukan CRUD dengan menambahkan atau mengurangi jumlah barang yang diinginkan. Setelah barang sudah sesuai bisa langsung dilakukan opsi *checkout* untuk memesan barang.



Gambar 10. Menu Checkout, Transaksi, Order

Gambar 10 menampilkan menu *checkout* dimana *user* dapat melakukan pemesanan. Produk yang sebelumnya dimasukkan ke *list checkout* akan ditampilkan disini. Terdapat menu yang menampilkan alamat, detail pesanan serta total harga yang harus dibayarkan. Ada juga opsi pembayaran, setelah memilih opsi dan melakukan pembayaran maka *user* dapat langsung order barangnya.

## 5. KESIMPULAN DAN SARAN

Kesimpulan dari penelitian ini menunjukkan bahwa metode Scrum cocok untuk manajemen proyek sistem informasi dengan pengembangan berkelanjutan, meskipun lebih sesuai untuk tim kecil. Metode Use Case Point efektif untuk menghitung

waktu dan biaya pengembangan sistem informasi manajemen rantai pasok downstream karena akurasi. Penelitian ini berhasil mengembangkan sistem yang memenuhi kebutuhan spesifik CV. Marvelindo Utama, meningkatkan efisiensi dan efektivitas operasional. Dokumentasi proyek yang komprehensif telah disusun, menegaskan pentingnya manajemen proyek yang terstruktur dalam pengembangan sistem informasi. Peneliti menyarankan untuk memperhitungkan waktu pengerjaan proyek dengan matang, memastikan kompatibilitas sistem, menjaga komunikasi tim dalam Scrum, dan menyusun dokumentasi yang lebih lengkap untuk penelitian serupa di masa mendatang.

## DAFTAR PUSTAKA

- [1] "Asosiasi Penyelenggara Jasa Internet Indonesia." <https://survei.apjii.or.id/> (accessed Jan. 21, 2024).
- [2] M. Fitriawati and R. H. Lestari, "Design of the Information System for Kindergarten Learning Plan used Scrum Methodology," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 407, no. 1, 2018, doi: 10.1088/1757-899X/407/1/012131.
- [3] M. Agarina, S. Karnila, and A. S. Karim, "The Application Of Scrum Agile Development Method In The Design Of Marketing Information Systems And Supply Of Goods In CV . Melodi Mega Kencana," no. December, pp. 158–167, 2020.
- [4] M. Yusnorizam Ma, "The Challenges of Implementing Agile Scrum in Information System's Project," *Jour Adv Res. Dyn. Control Syst.*, vol. 10, p. 9, 2018, [Online]. Available: <https://ssrn.com/abstract=3786550>
- [5] H. Rahman, M. A. Putra, W. Hayuhardhika, N. Putra, and D. Pramono, "Estimasi Biaya Perangkat Lunak menggunakan Metode Use Case Point (Studi Kasus: PT. Pln(Persero) Area Malang)," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 3, no. 6, pp. 5599–5607, 2019, [Online]. Available: <https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/5520>
- [6] E. Rodríguez Sánchez, E. F. Vázquez Santacruz, and H. Cervantes Maceda, "Effort and Cost Estimation Using Decision Tree Techniques and Story Points in Agile Software Development," *Mathematics*, vol. 11, no. 6, 2023, doi: 10.3390/math11061477.
- [7] R. D. Prihadianto, Y. Hariyati, I. Ibanah, K. Muhtadi, S. Suwasono, and D. Sumarno, "Model Ideal Sistem Manajemen Rantai Pasok Agribisnis Kakao: Pendekatan Soft System Methodology (SSM)," *JSEP (Journal Soc. Agric. Econ.*, vol. 15, no. 3, p. 333, 2022, doi: 10.19184/jsep.v15i3.35000.
- [8] E. Turban, E. McLean, and J. Wetherbe, "Information Technology for Management: Transforming Organizations in the Digital Economy," *Inf. Technol.*, vol. 2, no. 4, pp. 192–

- 221, 2000, doi: 10.1108/09593840010377644.
- [9] D. Darmawan and A. Ratnasari, "Rancang Bangun Sistem Informasi Manajemen Proyek Berbasis Web Pada Pt Seatech Infosys," *J. Sisfokom (Sistem Inf. dan Komputer)*, vol. 9, no. 3, pp. 365–372, 2020, doi: 10.32736/sisfokom.v9i3.931.
- [10] K. Schwaber and J. Sutherland, "Panduan Definitif untuk Scrum: Aturan Permainan," *Scrum.Org*, no. November, pp. 1–17, 2020.
- [11] M. T. Yahya, "Estimasi Usaha Dan Biaya Proyek Pengembangan Aplikasi Web Di Cv. Technobit Indonesia Menggunakan Metode Use Case Point," *J. Sist. Informasi, Teknol. Informasi, dan Edukasi Sist. Inf.*, vol. 4, no. 2, pp. 64–73, 2023, doi: 10.25126/justsi.v4i2.162.