

BAB II

TINJAUAN PUSTAKA

2.1. Tinjauan Pustaka

Perancangan dan Pembangunan *Customer Relationship Management (CRM)* berbasis *website* digunakan untuk meningkatkan produktivitas dan perusahaan dan *customer*. Studi kasus ini sudah pernah diteliti oleh beberapa peneliti sebelumnya dan dampak yang dihasilkan dapat memberikan manfaat di perusahaan.

Pada penelitian berjudul **“Perancangan *Customer Relationship Management* Berbasis *Web* Dengan *Library JQuery Responsive Datatables*”** (2019), oleh Somya, Ramos dan Utama, Setiawan Chandra, permasalahan pada penelitian ini adalah pelayanan keluhan pelanggan dari aplikasi yang sudah ada yang belum berjalan secara optimal dan aktor yang terlibat hanya *user sales* tanpa departemen yang bertugas atas pelayanan sehingga menyebabkan penanganan keluhan sering tertunda. Dari permasalahan tersebut, penelitian ini bertujuan untuk membuat *Customer Relationship Management* berbasis *web* menggunakan *library JQuery Responsive datatables* agar meningkatkan kinerja *sales* dan pemantauan proses perbaikan produk. Selama proses pengembangan, metode yang digunakan adalah *prototype*, bahasa pemrograman *PHP (framework codeigniter)*, *database mysql*, *Jquery*, *datatables* dan pengujian *black box*. Hasil penelitian ini adalah sistem informasi *CRM* yang memiliki manfaat dalam peningkatan pemantauan proses penanganan keluhan serta proses pelaporan *sales* [7].

Pada penelitian berjudul **“Aplikasi *E-CRM (Customer Relationship Management)* Berbasis *Web* Pada Belle Studio Cirebon”** (2020), oleh Magdelan, Lena dan Hatta, Muhammad, permasalahan pada penelitian ini adalah pengelolaan *booking* studio fotografi masih manual menggunakan *instagram* dan *whatsapp*, pencatatan informasi internal masih menggunakan buku, dan tidak ada penghargaan pelanggan tetap seperti *promo* sehingga kepuasan pelanggan masih belum

tercukupi. Dari permasalahan tersebut, penelitian ini bertujuan untuk membuat *Customer Relationship Management* berbasis *web*. Selama proses pengembangan, metode yang digunakan adalah *prototype* dan bahasa pemrograman *PHP*. Hasil dari penelitian ini adalah sistem *CRM* yang berhasil mempermudah pengelolaan *customer*, pelaporan, dan promo pelanggan sebagai member sehingga meningkatkan kepuasan pelanggan lama [8].

Pada penelitian berjudul **“Desain Dan Implementasi *Customer Relationship Management* Berbasis *Web* (Studi Kasus : Toko Baju Ladya)”** (2020), oleh Al Mudzakir, Tohirin dan Bakar, Abu, permasalahan pada penelitian ini adalah pemasaran produk yang masih menggunakan metode konvensional hal ini menyebabkan penyebaran informasi produk belum efektif sehingga peluang untuk mendapatkan calon pelanggan tidak maksimal. Dari permasalahan tersebut, penelitian ini bertujuan untuk membuat *Customer Relationship Management* berbasis *web* untuk memperluas peluang mendapatkan pelanggan baru. Selama proses pengembangan, metode yang digunakan adalah *prototype*, bahasa pemrograman *PHP (framework codeigniter)*, *database mysql*, dan serangkaian pengujian *black box* dan *white box*. Hasil penelitian ini adalah aplikasi *CRM* dan *SWOT* yang berhasil meningkatkan penjualan dengan analisa kelebihan dan kekurangan sistem penjualan sebelumnya [9].

Pada penelitian berjudul **“Rancang Bangun Berbasis *Web* CRM (*Customer Relationship Management*) Studi Kasus PT Budi Berlian Motor Hajimena Bandar Lampung”** (2021), oleh Saputra, Ade Kiki dan Fahrizal, Mico. Permasalahan pada penelitian ini adalah tidak ada sistem penerimaan keluhan pelanggan dan pelaporan *follow up* keluhan. Dari permasalahan tersebut, penelitian ini bertujuan untuk membuat *Customer Relationship Management* berbasis *web* untuk memfasilitasi layanan keluhan pelanggan. Selama proses pengembangan, metode yang digunakan adalah *waterfall*, bahasa pemrograman *macromedia dreamweaver*, dan *database mysql*. Hasil penelitian ini adalah aplikasi *CRM* yang sudah diimplementasikan pada divisi *service* dan berhasil mempermudah laporan *follow up service* dan penampungan keluhan *customer* [10].

Pada penelitian berjudul **“Sistem Informasi Pemasaran Dengan Penerapan CRM (*Customer Relationship Management*) Berbasis Website Menggunakan Metode *Waterfall* Dan *Agile*”** (2021), oleh Irsandi, Johan Saputra, Fitri, Iskandar dan Nathasia, Novi Dian. Permasalahan pada penelitian ini adalah media promosi masih menggunakan brosur, poster, dan pamflet, tidak ada layanan feedback dan pesan dari pelanggan. Dari permasalahan tersebut, penelitian ini bertujuan untuk membuat *Customer Relationship Management* berbasis *web* untuk mempermudah pihak toko mengambil keputusan pemasaran dan pelayanan kepuasan pelanggan. Selama proses pengembangan, metode yang digunakan adalah *waterfall* (pengembangan tahap pertama) dan *agile* (pengembangan tahap berikutnya), bahasa pemrograman *PHP*, *database mysql*, dan pengujian *black box*. Hasil penelitian ini adalah sistem *CRM* yang berhasil membuat pelayanan semakin efisien, mempermudah pemasaran, dan meningkatkan kepuasan pelanggan melalui layanan kritik dan saran [11].

Pada penelitian berjudul **“Implementasi Metode *Scrum* Dan Analisis SWOT Sebagai Strategi Framework *Customer Relationship Management* (CRM) Pada Perusahaan Rental Mobil”** (2021), oleh Eriana, Emi Sita, dan Zein, Afrizal. Permasalahan pada penelitian ini tidak adanya sistem yang bisa memantau *customer*, kurangnya media informasi bagi *customer*, dan pelayanan yang belum optimal. Dari permasalahan tersebut, penelitian ini bertujuan untuk meningkatkan loyalitas *customer* melalui *Customer Relationship Management* berbasis *web* dengan pendekatan *scrum* dan analisis SWOT. Hasil dari penelitian ini adalah *Website* e-CRM berbasis *web* yang memberikan fungsionalitas pemesanan dan media informasi rental [12].

Pada penelitian berjudul “**Model Agile Scrum untuk Pengembangan e-Customer Relationship Management Pendukung Layanan Sewa Gudang**” (2023), oleh Haqqizar, Nizam, Widyarningsih, Tri Wahyu, dan Dewi, Meta Amalya. Permasalahan pada penelitian ini adalah tidak ada rekam jejak klien, aktifitas tim *sales*, dan terbatasnya sarana dan prasarana unit sehingga menghambat kinerja tim *sales*. Dari permasalahan tersebut, penelitian ini bertujuan untuk membuat *Customer Relationship Management* berbasis *web* dengan pendekatan *scrum* untuk memberikan layanan sewa gudang yang lebih mudah. Selama proses pengembangan, bahasa pemrograman yang digunakan adalah *PHP*, *framework Yii*, *Javascript*, *HTML*, *CSS*, *Bootstrap*, *database mysql*, dan pengujian *black box*. Hasil dari penelitian ini adalah Aplikasi e-CRM berbasis *web* yang memberikan fungsionalitas pengelolaan prospek, pembuatan proposal, dan penjadwalan pertemuan [13].

Tabel 2. 1 Ringkasan penelitian terdahulu

No.	Judul	Metode	Hasil	Perbedaan
1	Perancangan <i>Customer Relationship Management</i> Berbasis Web Dengan <i>Library JQuery Responsive Databases</i> . Somya R, Candra Utama 2019 [7]	<i>Prototype</i>	Sistem informasi <i>CRM</i> berbasis <i>website</i> yang memiliki manfaat dalam meningkatkan pemantauan proses penanganan keluhan serta proses pelaporan <i>sales</i>	Metode yang digunakan berbeda, penulis penelitian menggunakan metode <i>Scrum. CRM</i> yang dikembangkan difokuskan untuk pelayanan dan pelaporan <i>sales</i> .
2	Aplikasi E-CRM (<i>Customer Relationship Management</i>) Berbasis Web Pada Belle Studio Cirebon. Magdalena 2020 [8]	<i>Prototype</i>	Sistem <i>CRM</i> berbasis <i>website</i> yang berhasil mempermudah pengelolaan <i>customer</i> , pelaporan, dan promo pelanggan sebagai member sehingga meningkatkan kepuasan pelanggan lama	Metode yang digunakan berbeda, penulis penelitian menggunakan metode <i>Scrum. CRM</i> yang dikembangkan difokuskan untuk meningkatkan kepuasan <i>customer</i> melalui program loyalitas promo.
3	Desain Dan Implementasi <i>Customer Relationship Management</i> Berbasis Web (Studi Kasus : Toko Baju Ladya). Mudzakir	<i>Prototype</i>	Aplikasi <i>CRM</i> berbasis <i>website</i> dengan analisis SWOT yang berhasil meningkatkan penjualan dengan analisa kelebihan dan kekurangan sistem penjualan sebelumnya .	Metode yang digunakan berbeda, penulis penelitian menggunakan metode <i>Scrum. CRM</i> yang dikembangkan difokuskan untuk

No.	Judul	Metode	Hasil	Perbedaan
	2020 [9]			pemasaran dan penjualan produk dengan pendekatan analisis <i>SWOT</i> .
4	Rancang Bangun Berbasis Web CRM (<i>Customer Relationship Management</i>) Studi Kasus PT Budi Berlian Motor Hajimena Bandar Lampung. Saputra 2021 [10]	<i>Waterfall</i>	Aplikasi CRM berbasis <i>website</i> yang sudah diimplementasikan pada divisi service dan berhasil mempermudah laporan <i>follow up</i> service dan penampungan keluhan <i>customer</i>	Metode yang digunakan berbeda, penulis penelitian menggunakan metode <i>Scrum</i> .
5	Sistem Informasi Pemasaran Dengan Penerapan CRM (<i>Customer Relationship Management</i>) Berbasis <i>Website</i> Menggunakan Metode <i>Waterfall</i> Dan <i>Agile</i> . Saputra Irsandi 2021 [11]	<i>Waterfall</i> (pengembangan tahap pertama), <i>Agile</i> (pengembangan tahap selanjutnya)	Sistem CRM berbasis <i>website</i> yang berhasil membuat pelayanan semakin efisien, mempermudah pemasaran, dan meningkatkan kepuasan pelanggan melalui layanan kritik dan saran	Metode yang digunakan berbeda, penulis penelitian menggunakan metode <i>Scrum</i> . CRM yang dikembangkan difokuskan untuk media promosi dan penampungan kritik saran <i>customer</i> .
6	Implementasi Metode <i>Scrum</i> Dan Analisis <i>SWOT</i> Sebagai Strategi Framework <i>Customer Relationship Management</i>	<i>Scrum</i>	<i>Website e-CRM</i> berbasis <i>web</i> yang memberikan fungsionalitas pemesanan dan media informasi rental.	CRM yang dikembangkan difokuskan untuk pemesanan produk dan media informasi,

No.	Judul	Metode	Hasil	Perbedaan
	(CRM) Pada Perusahaan Rental Mobil. Sita Eriana 2021 [12]			pemasaran dan penjualan dengan pendekatan analisis <i>SWOT</i> .
7	Model <i>Agile Scrum</i> untuk Pengembangan <i>e-Customer Relationship Management</i> Pendukung Layanan Sewa Gudang. Haqqizar 2023 [13]	<i>Scrum</i>	Aplikasi e-CRM berbasis <i>web</i> yang memberikan fungsionalitas pengelolaan prospek, pembuatan proposal, dan penjadwalan pertemuan.	Penggunaan <i>framework Yii</i> . <i>CRM</i> yang dikembangkan difokuskan untuk pengelolaan prospek dan sistem penjadwalan.

2.2. Landasan Teori

2.2.1. Rancang Bangun

Rancang bangun adalah proses analisis kebutuhan produk berupa perencanaan dan desain yang terdiri dari beberapa bagian terpisah yang nantinya akan menjadi suatu kesatuan yang memiliki fungsi. Istilah rancang bangun sering digunakan pada karya ilmiah untuk membuat produk [14].

2.2.2. Sistem

Sistem adalah komponen-komponen yang memiliki fungsi untuk mencapai tujuan tertentu dan saling terikat satu sama lain. Tujuan dari pembuatan sistem adalah untuk mengorganisir proses agar terstruktur [15].

2.2.3. *Customer Relationship Management (CRM)*

Customer Relationship Management (CRM) merupakan salah satu strategi bisnis yang bertujuan untuk meningkatkan kinerja internal dan meningkatkan kepuasan pelanggan menggunakan teknologi informasi. *CRM* pada dasarnya memiliki kemampuan untuk pengelolaan informasi internal. *CRM* memiliki beberapa keuntungan bagi pengelola bisnis, antara lain:

1. Meningkatkan pelayanan dan kepuasan *customer*

Bentuk pelayanan pada *CRM* terhadap *customer* bisa dilakukan dengan berbagai cara, antara lain pengelolaan pesanan, perbaikan, konsultasi, kritik dan saran.

2. Meningkatkan kualitas pemasaran

CRM memiliki kemampuan untuk mengelola semua data *customer*, oleh karena itu, sebagai pengelola bisnis dapat memanfaatkan fungsionalitas tersebut untuk meningkatkan strategi pemasaran, salah satu caranya dengan mengirimkan promo kepada *customer* melalui data yang tersedia.

3. Meningkatkan kinerja tim *sales*

Sistem *CRM* membantu *sales* untuk melihat data *customer* seperti calon *customer*, detail *customer*, riwayat pembayaran, dan bukti berita acara sehingga tim *sales* bisa memanfaatkan data tersebut untuk meningkatkan penjualan secara *up-sell* dan *cross-sell*.

4. Meningkatkan kualitas pengelolaan data internal

Sistem *CRM* membantu pengelolaan data internal menjadi informasi yang digunakan untuk pengambilan keputusan di kemudian hari [16].

2.2.4. *Operational Customer Relationship Management*

Operational CRM salah satu tipe *CRM* untuk menyelesaikan permasalahan-permasalahan operasional dibagian *front office* (bagian yang bertanggung jawab menangani *customer*) seperti masalah pelayanan yang akan diberikan kepada *customer*. Ciri khas *Operational CRM* adalah kemampuan *automation* (proses pengoptimalan menggunakan teknologi), diantaranya adalah:

1. *Service Automation*

Bentuk pengoptimalan proses penanganan masalah layanan terhadap *customer* yang meliputi *Service Level Management (SLM)* dan *Interactive Voice Response (IVR)* untuk melayani *customer* secara tatap muka secara daring.

2. *Marketing Automation*

Bentuk pengoptimalan proses pemasaran seperti memanfaatkan semua pertanyaan, keluhan dan saran dari *customer* yang bisa dijadikan evaluasi untuk strategi bisnis selanjutnya, dan strategi promosi yang dapat dilakukan secara tepat sasaran dengan cara memfilter *customer* yang berpotensi yang bisa dilihat dari lama langganan dan jumlah transaksi.

3. *Sales Force Automation (SFA)*

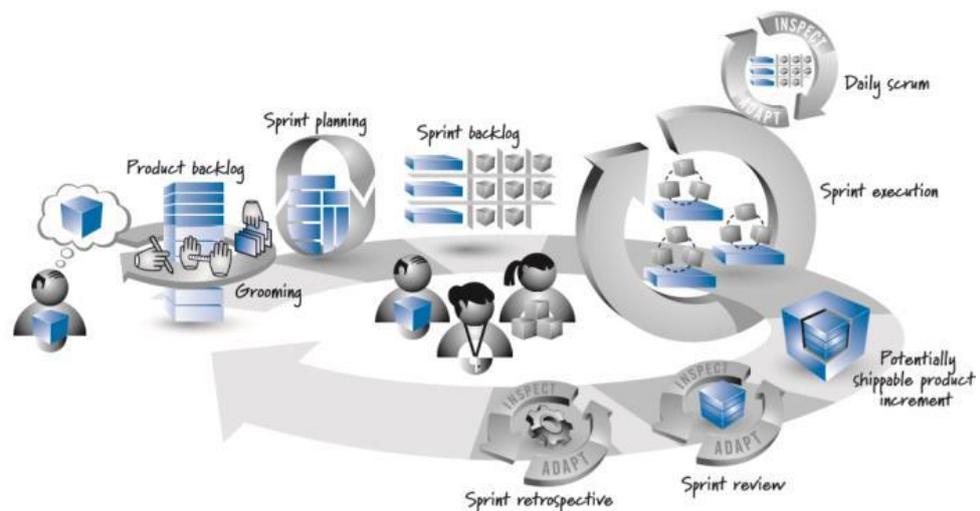
Bentuk pengoptimalan proses di bagian *sales* seperti manajemen akun *customer*, manajemen proposal, dan konfigurasi produk dimana

customer diberikan hak untuk berkontribusi memberikan pertanyaan, saran, kritik, dan keluhan dari produk yang dimiliki perusahaan [17].

2.2.5. Customer

Pelanggan atau *customer* adalah orang yang terbiasa membeli produk atau jasa yang ditawarkan oleh perusahaan atau orang yang memiliki banyak riwayat transaksi pembelian atau langganan. *Customer* berbeda dengan konsumen, perbedaan terletak pada intensitas transaksinya (*customer* memiliki transaksi yang kontinu), konsumen membeli produk untuk diri sendiri sedangkan *customer* biasanya membeli produk untuk dirinya sendiri dan orang lain [18].

2.2.6. Scrum



Gambar 2. 1 Metode *Scrum* [19]

Scrum adalah metode pengembangan perangkat lunak cukup populer, *scrum* merupakan implementasi metode *agile* yang dikemas sebagai kerangka kerja, pada penggunaan *scrum*, selama proses pengembangan perangkat lunak diharapkan selalu merespon perubahan secara cepat. Penggunaan *scrum* mempunyai siklus atau tahapan dilakukan secara berulang yang disebut dengan *sprint*. Setiap *sprint* akan digunakan untuk mengerjakan tugas-tugas yang ada dan setiap *sprint* memiliki rentang waktu

umumnya 1 sampai 4 minggu. *Scrum* memiliki tim yang terdiri dari beberapa aktor yang antara lain:

1. *Product owner*

Product owner adalah orang yang bertugas untuk menentukan fitur-fitur, tanggal rilis dan prioritas produk, *product owner* memiliki hak untuk merevisi hasil dari produk yang dikembangkan.

2. *Scrum master*

Scrum master adalah orang yang bertugas mengelola jalannya *sprint*, memantau *progress*, dan mentoring tim.

3. *Developer*

Developer adalah orang yang fokus untuk pengembangan produk, contohnya *software engineer*, *UI/UX*, *Quality Assurance*, dan sebagainya.

Pada penggunaan *scrum* terdapat beberapa tahapan yang harus dilakukan, antara lain:

1. *Product Backlog*

Product backlog adalah daftar fitur atau *requirement* yang harus diselesaikan selama pengembangan perangkat lunak. *Product backlog* bersifat fleksibel dan dapat ditentukan oleh *product owner*. *Product backlog* didapatkan dari *user stories* (narasi dalam bahasa sehari-hari dari klien terkait kebutuhan sistem). Tahapan untuk menentukan *user story* adalah sebagai berikut:

- a) Menyiapkan daftar pertanyaan dan melakukan wawancara kepada klien atau *user*. Bentuk pertanyaan yang diajukan terkait permasalahan, *business flow*, kebutuhan fungsional dari permasalahan setiap aktor yang terlibat, dan sebagainya.

Tabel 2. 2 daftar pertanyaan dan jawaban wawancara [20]

No.	Pertanyaan	Jawaban
-----	------------	---------

1.	Apakah terdapat permasalahan di MF Binary Digital Solution ?	Sebelumnya saya memasarkan produk saya melalui <i>marketplace</i> , namun saya merasa belum dapat memaksimalkan <i>branding</i> dari produk yang saya tawarkan karena ketika pelanggan mencari produk saya, ada banyak produk lain juga yang ditawarkan kepada pelanggan saya yang terkadang membuat pelanggan saya berpaling ke produk lain tersebut.
2.	Bagaimana alur pemasaran yang anda lakukan selama menjalankan bisnis?	Pertama, saya menambahkan produk saya di toko <i>marketplace</i> , kemudian ketika ada yang pesan, saya otomatis akan mendapat notifikasi pesanan masuk. Selanjutnya saya akan <i>packing</i> pesanan dan segera kirim produk ke pelanggan melalui jasa pengiriman.
3.	Dari permasalahan pemasaran apakah anda sudah pernah memikirkan solusi untuk menyelesaikan masalah tersebut?	Saya pernah berpikir untuk melakukan pemasaran di media yang lainnya seperti <i>website</i> ataupun <i>social media</i> .
4.	Jika anda ingin memiliki <i>website</i> penjualan, fitur seperti	<ul style="list-style-type: none"> • Saya ingin punya <i>website</i> yang informatif dengan logo dan <i>branding</i> yang bagus

	apa yang anda inginkan?	<p>agar pelanggan bisa mengenal produk saya.</p> <ul style="list-style-type: none"> • Saya ingin pelanggan bisa melihat katalog produk saya, mencari produk, dan bisa <i>filter</i> kategori dan harga. • Saya ingin ada fitur pembayaran digital dan otomatis seperti <i>QRIS</i>, <i>GOPAY</i>, dan sebagainya. • Saya ingin memiliki fitur pengelolaan produk dan melihat pesanan pelanggan.
--	-------------------------	--

Pada tabel 2.2, adalah contoh format pertanyaan dan jawaban wawancara, hasil tersebut akan menjadi *input* untuk dikelola menjadi *user story*.

- b) Mengolah hasil wawancara menjadi *user story* dengan membuat tabel *user story*.

Tabel 2. 3 Contoh *user story* [20]

Aktor	Story
Pelanggan	Sebagai seorang pelanggan, saya ingin mencari produk yang tersedia pada <i>e-commerce</i> , sehingga saya membutuhkan halaman pencarian yang memudahkan saya untuk mencari produk berdasarkan <i>brand</i> , kategori, dan harga.
	Sebagai seorang pelanggan saya ingin membeli banyak produk dalam satu kali transaksi sehingga saya membutuhkan keranjang virtual yang dapat

Aktor	Story
	menampung produk-produk yang saya beli sebelum dilakukan pembayaran.
	Sebagai seorang pelanggan saya ingin memiliki akun terdaftar pada <i>e-commerce</i> sehingga setiap kali pemesanan dapat dilakukan dengan mudah tanpa harus memasukkan info pribadi berulang kali.
Admin	Sebagai seorang admin, saya ingin melihat ringkasan pesan yang telah dilakukan <i>customer</i> seperti pesanan yang sudah dibayar, pesanan yang belum dibayar, dan income yang diperoleh.
	Sebagai seorang Admin, saya ingin mengelola data produk seperti menambah produk baru, memperbarui produk, dan menghapus produk.
	Sebagai seorang Admin, saya ingin memiliki akun khusus yang digunakan untuk <i>login</i> ke dalam sistem untuk melakukan pengelolaan produk, pesanan, dan <i>customer</i> .

Pada tabel 2.3 merupakan contoh hasil dari *user story* yang berisi aktor dan kebutuhan fitur setiap aktor. Setelah membuat *user story* tahap selanjutnya adalah mengolah *user story* menjadi tabel *product backlog*. Cara mengolah *user story* menjadi *product backlog* adalah dengan *creating and refinement*, *estimation*, dan *prioritizing*. Ketiga aktifitas tersebut disebut *product backlog grooming*, selanjutnya detail dari tiap aktifitas adalah sebagai berikut:

- a. *Creating and refinement*, proses membuat atau menentukan *product backlog* atau menyesuaikan *product backlog* jika terdapat perubahan pada saat proses *scrum* berjalan.

- b. *Estimation*, proses memperkirakan waktu dan tingkat kompleksitas fitur dengan *story poin*. *Story point* digunakan untuk menentukan seberapa besar atau kompleks fitur yang akan dibuat, *stories point* bisa didapatkan dari teknik *planning poker* (deret *fibonacci*) [21]. Contohnya fitur *login* akan diberikan 1 poin sedangkan fitur pengelolaan *user* akan diberikan 3 poin, artinya fitur *login* relatif lebih mudah dikerjakan daripada fitur pengelolaan *user*. Semakin tinggi poin maka semakin lama pengerjaan fiturnya.
- c. *Prioritizing*, proses menentukan prioritas pengerjaan fitur (*low, medium, high*). Cara menentukan prioritas fitur adalah dengan berdiskusi dengan klien.

Tabel 2. 4 Contoh hasil *product backlog* [20]

Fitur	Priority	Point	Durasi
<i>List Product</i>	<i>High</i>	13	2 hari
<i>Detail Product</i>	<i>Medium</i>	10	1 hari
<i>Shoping Cart</i>	<i>Medium</i>	13	1 hari
<i>Checkout</i>	<i>High</i>	13	3 hari
<i>Register User</i>	<i>High</i>	5	2 hari
<i>Login</i>	<i>High</i>	5	3 hari

Pada tabel 2.4 merupakan hasil dari *product backlog* yang diperoleh dari mengolah *user story* pada tabel 2.3, tabel tersebut terdiri dari nama *product backlog*, prioritas (*high, medium, dan low*), *story point*, dan estimasi waktu.

2. *Sprint Planning*

Sprint planning adalah proses menentukan rencana *sprint* yang akan dikerjakan, mulai dari penentuan durasi pengerjaan tiap *sprint*, batas

story poin tiap *sprint*, *product backlog* pada setiap *sprint* sesuai dengan prioritas dan *sprint goal* (target pengerjaan fitur pada satu *sprint*). *Input* dari tahap ini adalah *product backlog*. Berikut adalah tahap-tahap untuk melakukan *sprint planning*.

- a) Menentukan berapa lama durasi tiap *sprint* (1 – 4 minggu) dan durasi satu *sprint* idealnya adalah sama untuk semua *sprint*. Contohnya jika menggunakan 3 minggu untuk satu *sprint* maka setiap *sprint* akan berdurasi 3 minggu dan total estimasi waktu untuk pengerjaan *product backlog* tidak boleh melebihi batas 3 minggu tersebut.
- b) Menentukan estimasi jumlah *story poin* yang dapat diselesaikan dalam satu *sprint*. Contohnya jika perkiraan poin dalam satu *sprint* adalah 20-25 *story poin*, artinya ambil *product backlog* dengan prioritas tertinggi dan *product backlog* yang diambil tidak boleh melebihi batas jumlah *story poin* pada *sprint* tersebut.
- c) Menentukan *sprint goal* (target pengerjaan fitur) dan *product backlog* dengan cara mengambil *product backlog* yang akan dikerjakan dengan memperhatikan prioritas, batas *story poin*, dan batas estimasi waktu yang sudah ditentukan pada tahap sebelumnya.

Tabel 2. 5 Contoh hasil *sprint goal* [20]

<i>Sprint</i>	Deskripsi
<i>Sprint 1</i>	Menyediakan halaman untuk menampilkan daftar produk.
	Membuat fitur pencarian produk berdasarkan kategori dan harga.

<i>Sprint</i>	Deskripsi
	Membuat halaman detail produk.
	Membuat halaman keranjang belanja.

Pada tabel 2.5 adalah hasil *sprint goal* yang berisi target fitur yang akan diselesaikan pada *sprint* yang akan dieksekusi.

3. *Sprint Backlog*

Sprint Backlog adalah proses memecah *product backlog* menjadi bagian-bagian yang lebih kecil atau lebih detail yang nantinya akan menjadi sebuah tugas pada sebuah *sprint*. *Input* dari tahap ini adalah hasil dari *sprint planning*. Selanjutnya untuk menentukan *sprint backlog* dapat dilakukan dengan cara memecah setiap *product backlog item* yang didapat dari tahap *sprint planning* menjadi lebih kecil, dan pemberian estimasi waktu dari setiap *task*.

Tabel 2. 6 Contoh hasil *sprint backlog* [20]

ID-Task	Fitur	Deskripsi	Poin	Estimasi
T-001	<i>List Product</i>	<i>Design tampilan list produk</i>	13	2 hari
		<i>Backend fitur list produk</i>		
T-002	<i>Searching product.</i>	<i>Design tampilan fitur searching product.</i>	8	2 hari
	<i>Slicing dashboard admin windmill</i>	<i>Backend fitur searching produk.</i>		1 hari

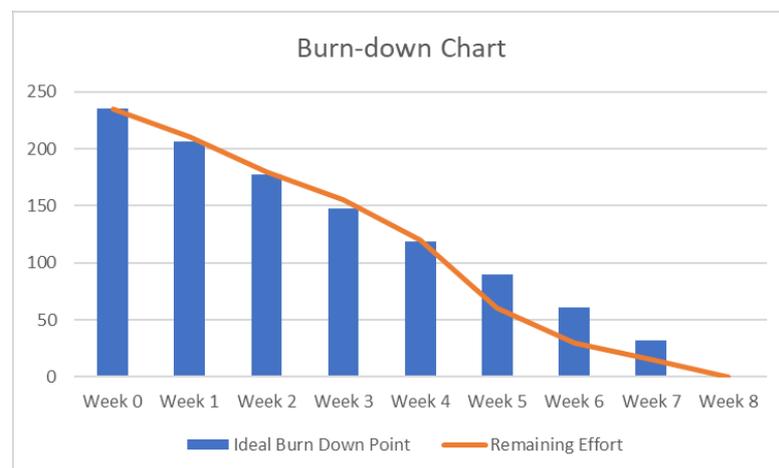
Pada tabel 2.6 adalah contoh hasil dari *sprint backlog*, hasil tersebut berisi pendetailan pemecahan *product backlog item* beserta penjadwalan dari hasil sebelumnya yaitu *sprint planning*.

4. *Sprint Execution*

Sprint execution adalah siklus dalam *scrum* untuk menyelesaikan tugas-tugas yang sudah dirancang pada *sprint backlog* dengan rentang waktu yang diatur. Tahap ini akan diulang sesuai dengan kebutuhan dan perencanaan *sprint*. Tahap ini memiliki beberapa aktifitas, antara lain:

a. *Daily scrum*

Daily scrum adalah *meeting* singkat (10 - 15 menit) setiap hari bersama tim untuk membahas *progress*, tugas yang belum dikerjakan, kendala, dan apa yang akan dikerjakan pada hari itu. Cara untuk menggambarkan *progress sprint* adalah melalui *burndown chart* dengan cara menentukan target poin dan poin yang didapat hari ini.



Gambar 2. 2 Contoh *burndown chart* untuk pemantauan progres *sprint* pada *daily scrum*. [20]

Pada gambar 2.2, merupakan contoh pemantauan *progress sprint* yang digambarkan menggunakan *burndown chart*. *Burndown chart* menunjukkan target (garis biru) dan perolehan

poin (garis oren) setiap hari sehingga dapat dipantau apakah *sprint* yang berjalan terdapat kendala atau percepatan.

b. Pengerjaan tugas

Mulai mengerjakan tugas-tugas yang sudah didefinisikan dan dibagi pada hasil *daily scrum*.

5. *Potentially Shippable product increment*

Potentially shippable product increment adalah produk dan rancangan yang dihasilkan selama satu *sprint* yang artinya bukan produk secara utuh dan nantinya akan direview oleh tim pada tahap *sprint review*. Cara menentukannya adalah mendefinisikan fitur apa saja yang sudah diselesaikan dalam satu *sprint* ke dalam bentuk tabel.

Tabel 2. 7 Contoh *potentially shippable product increment* [22]

No.	Fitur	Status
1.	<i>Login dan register</i>	<i>Done</i>
2.	<i>List product</i>	<i>Done</i>
3.	<i>Searching product</i>	<i>Done</i>

Pada tabel 2.7 merupakan contoh *potentially shippable product increment* atau fitur dihasilkan pada satu *sprint* dan siap untuk *direview* bersama klien.

6. *Sprint Review dan Sprint Retrospective*

Sprint review adalah proses *meeting* bersama klien untuk memeriksa hasil pengerjaan selama satu *sprint* apakah sudah sesuai dengan kebutuhan klien, hasil *meeting* dapat berupa penambahan *requirement* baru yang akan dijadikan *product backlog item* ke *sprint* selanjutnya. Sedangkan *sprint retrospective* proses untuk memeriksa apakah selama pengerjaan dalam satu *sprint* ada kendala dengan manajemen bisa dijelaskan dalam kalimat deskriptif [22].

2.2.7. Website

Menurut Yuhfizar, *website* adalah media informasi berbentuk tulisan, gambar maupun suara yang terdapat pada suatu halaman yang saling terhubung [23]. Kelebihan *website* salah satunya adalah kemudahan akses karena hanya menggunakan internet untuk mengaksesnya, mudah digunakan karena memiliki antarmuka yang dirancang sebaik mungkin, dan *website* sudah memiliki standarisasi sehingga sudah banyak digunakan.

2.2.8. UML

UML (Unified Modeling Language) merupakan penggambaran atau visualisasi dari sebuah sistem yang didokumentasikan dengan baik menggunakan diagram dan teks. Bentuk visualisasi ini akan menjadi sebuah referensi bagi pengembang karena dapat membantu untuk analisis awal pada perancangan dan pembangunan [24]. *UML* banyak digunakan sebagai pemodelan untuk metodologi berorientasi objek. *UML* memiliki 4 kategori diagram, diagram yang umum digunakan untuk pengembangan antara lain:

1. Use case Diagram

Use case Diagram adalah model yang digunakan untuk menggambarkan interaksi antara satu atau lebih aktor terhadap fitur di sebuah sistem.

Tabel 2. 8 Simbol *Use case Diagram* [25]

Simbol	Keterangan
	<i>Use case</i> menggambarkan aktifitas atau fitur yang dilakukan oleh aktor pada sebuah sistem.
	<i>Association</i> menggambarkan komunikasi antara aktor dengan <i>Use case</i> .
	Aktor menggambarkan orang atau entitas yang terlibat atau berinteraksi dengan sistem.

<<extend>> ----->	Extend menggambarkan relasi terhadap <i>use case</i> untuk memperluas fungsionalitas <i>use case</i> turunannya.
<<include>> ----->	Include menggambarkan bahwa <i>use case</i> tambahan membutuhkan <i>use case</i> utama sebagai syarat <i>use case</i> tambahan dijalankan.

Pada tabel 2.8 merupakan simbol dan keterangan pada *use case* diagram yang terdiri dari aktifitas, asosiasi, *extend* dan *include*. Adapun langkah-langkah untuk membuat sebuah *use case* adalah sebagai berikut:

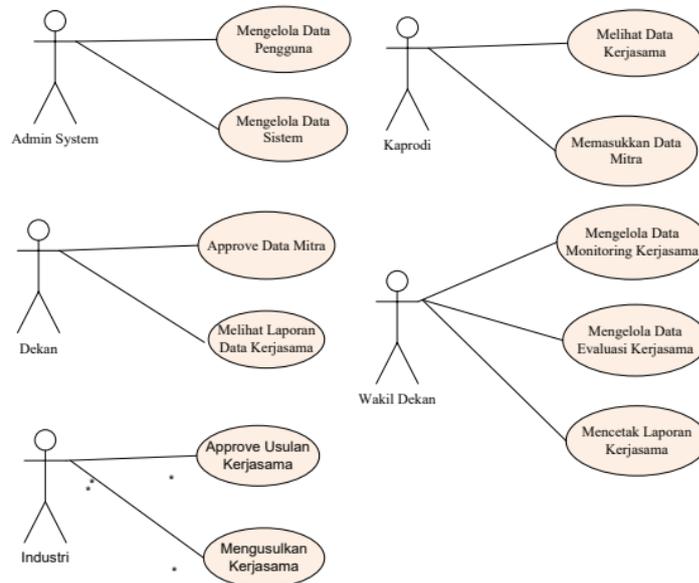
- a) Siapkan daftar pertanyaan wawancara dan lakukan wawancara.
- b) Buat tabel analisis kebutuhan sistem berdasarkan hasil wawancara.

Tabel 2. 9 Analisis kebutuhan sistem [26]

No.	Pengguna	Kebutuhan
1.	Administrator sistem	Mengelola data pengguna.
		Mengelola data sistem
2.	Dekan	<i>Approve</i> data mitra
		Melihat data laporan kerja sama.
3.	Wakil dekan	Mengelola data monitoring kerja sama
4.	Kaprodi	Cetak laporan kerja sama
		<i>Input</i> data mitra
5.	Industri	Setujui permintaan kerja sama.

Pada tabel 2.9 merupakan hasil pengolahan dari wawancara kebutuhan sistem.

- c) Buat *use case* diagram dari tabel kebutuhan sistem dengan ketentuan pengguna akan menggunakan simbol aktor dan kebutuhan sistem menggunakan simbol *use case*.



Gambar 2. 3 Contoh *Use case diagram* [26]

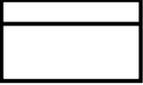
Pada gambar 2.3 merupakan *use case diagram* yang dihasilkan dari kebutuhan sistem yang digambarkan dengan simbol *use case*.

2. Activity Diagram

Activity Diagram adalah model yang menggambarkan aliran proses atau tugas dari sebuah sistem. Tujuan dari model ini adalah untuk menjabarkan detail dari proses atau tugas yang dilakukan oleh sistem.

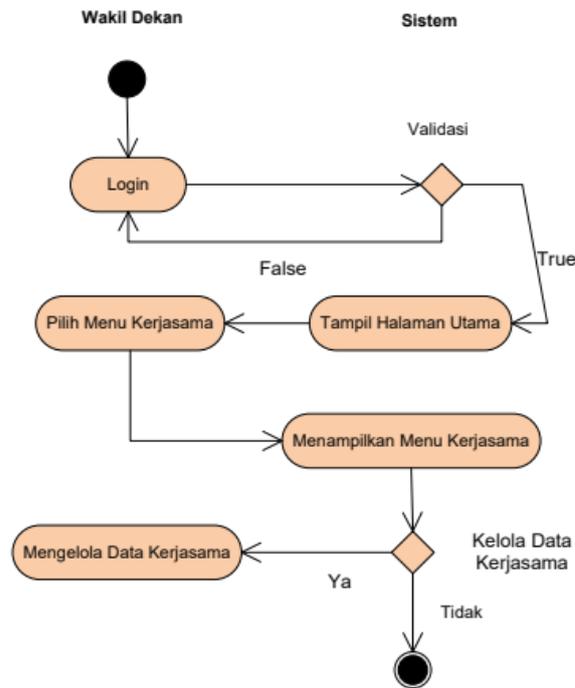
Tabel 2. 10 Simbol *Activity Diagram* [25]

Simbol	Keterangan
	Status awal menggambarkan awal mula proses atau aktifitas pada sebuah sistem yang dijalankan.
	Status akhir menggambarkan akhir dari proses atau aktifitas pada sebuah sistem yang dijalankan.
	Aktifitas menggambarkan proses maupun fungsi yang dijalankan pada sebuah sistem.

Simbol	Keterangan
	Decision menggambarkan percabangan dari aktifitas lebih dari satu.
	Join menggambarkan gabungan dari beberapa aktifitas menjadi satu aktifitas.
	Swimlanes menggambarkan pembagian dan pembeda organisasi bisnis yang bertanggung jawab atas berbagai aktifitas.

Pada tabel 2.10 merupakan simbol dan keterangan pada *activity diagram* yang terdiri dari status awal, status akhir, aktifitas, asosiasi, *extend* dan *include*. Adapun langkah-langkah untuk membuat sebuah *activity* adalah sebagai berikut:

- a) Tentukan atau pilih aktifitas dari *use case diagram*. Contohnya aktifitas *login*.
- b) Uraikan langkah-langkah dari aktifitas tersebut, contoh wakil dekan diharuskan *login* terlebih dahulu kemudian masuk halaman utama dan memilih menu kerja sama untuk mengelola data kerja sama.
- c) Gambarkan uraian tersebut kedalam simbol *activity diagram* sesuai dengan ketentuan.



Gambar 2. 4 Contoh *activity diagram* [26]

Pada gambar 2.4 merupakan *activity diagram* dari proses *login* yang dihasilkan dari uraian proses *login*.

3. *Sequence Diagram*

Sequence diagram merupakan model yang menggambarkan perilaku, interaksi, metode, dan urutan proses dari sebuah objek *use case* terhadap objek yang lain.

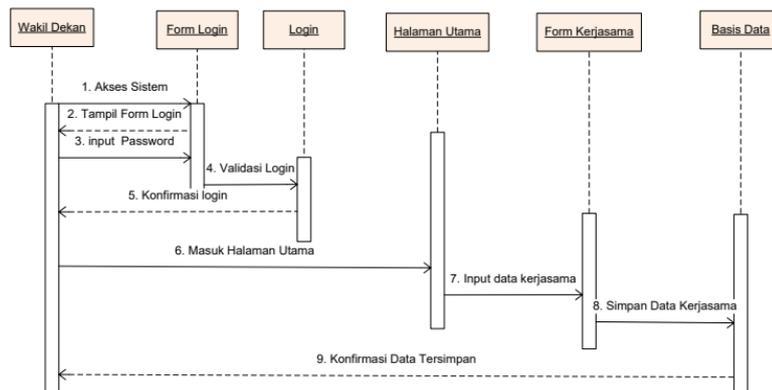
Tabel 2. 11 Simbol *Sequence Diagram* [25]

Simbol	Keterangan
	Aktor menggambarkan orang atau entitas yang terlibat atau berinteraksi dengan sistem
	Objek , menggambarkan objek yang bisa berinteraksi menggunakan pesan.
	Garis hidup menggambarkan kehidupan sebuah objek.

Simbol	Keterangan
	<p>Waktu aktif menggambarkan objek berstatus aktif dan sedang berinteraksi. Jika simbol ini memiliki beberapa hubungan, maka hubungan tersebut merupakan tahapan yang sedang dijalankan.</p>
<p style="text-align: center;">pesan</p> 	<p>Pesan menggambarkan pengiriman informasi atau masukan dari sebuah objek ke objek lain</p>
<p style="text-align: center;">keluaran</p> 	<p>Keluaran menggambarkan bahwa objek telah menjalankan aksi dan menghasilkan kembalian ke objek penerima.</p>

Pada tabel 2.11 merupakan simbol dan keterangan pada *activity diagram* yang terdiri dari aktor, objek, garis hidup, waktu aktif, pesan, dan keluaran. Adapun langkah-langkah untuk membuat sebuah *sequence* adalah sebagai berikut:

- a) Tentukan atau pilih aktifitas dari *use case diagram*. Contohnya aktifitas *login*.
- b) Uraikan langkah-langkah dari aktifitas tersebut, contoh dimulai dengan pengguna mengakses *form login*, kemudian memasukkan data *password* dan *username*. Data *login* kemudian divalidasi oleh sistem agar pengguna dapat masuk ke dalam halaman utama dan memilih menu mengelola data kerja sama.
- c) Gambarkan uraian tersebut kedalam simbol *sequence diagram* sesuai dengan ketentuan.

Gambar 2. 5 Contoh *sequence diagram* [26]

Pada gambar 2.5 merupakan *sequence diagram* dari proses *login* yang dihasilkan dari pendetailan waktu dari proses *login*.

4. *Class diagram*

Class diagram adalah model yang menggambarkan sistem berdasarkan kelas-kelas yang memiliki *attribute* dan fungsi [27].

Tabel 2. 12 Simbol *Class Diagram* [25]

Simbol	Keterangan
	Kelas menggambarkan sebuah rancangan bagi sebuah objek yang terdiri dari <i>attribute</i> dan fungsi.
	Asosiasi menggambarkan hubungan terhadap kelas lain dengan makna sama atau setara.
	Komposisi menggambarkan relasi antar kelas dengan salah satu kelas adalah bagian dari kelas lain, kelas tersebut tidak dapat berdiri sendiri.
	Agregasi menggambarkan relasi antar kelas dengan salah satu kelas adalah bagian dari kelas lain, namun semua kelas dapat berdiri sendiri.

Pada tabel 2.12 merupakan simbol dan keterangan pada *class* diagram yang terdiri dari kelas, asosiasi, komposisi, dan agregasi. Adapun langkah-langkah untuk membuat sebuah *class* adalah sebagai berikut:

- a) Tentukan atau pilih aktifitas dari *use case diagram*. Contohnya aktifitas *login*.
- b) Definisikan sebuah tabel yang merepresentasikan setiap struktur tabel di *database*.

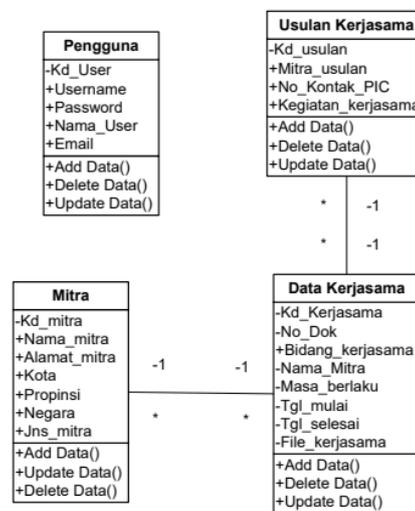
Tabel 2. 13 Contoh definisi setiap struktur tabel di *database*

No.	Class	Keterangan	Operasional
1.	Pengguna	Mendefinisikan data pengguna yang memiliki hak mengakses sistem.	Mampu menambah, menghapus dan menambah data pengguna.
2.	Mitra	Mendefinisikan data mitra kerja sama yang memiliki hubungan kerja sama dengan fakultas.	Mampu menambah, menghapus dan menambah data mitra.
3.	Data kerja sama	Mendefinisikan data dokumen kerja sama yang dimiliki oleh fakultas serta telah memiliki kegiatan.	Mampu menambah, menghapus dan menambah data kerja sama.
4.	Usulan kerja sama	Mendefinisikan data usulan kerja sama dari program studi (kaprodi) kepada fakultas untuk	Mampu menambah, menghapus dan menambah data usulan kerja sama

		diperiksa dan disetujui Dekan	
--	--	----------------------------------	--

Pada tabel 2.13 merupakan hasil representasi definisi dan fungsi dari setiap tabel di *database*.

- c) Gambarkan uraian tersebut kedalam simbol *class diagram* sesuai dengan ketentuan.



Gambar 2. 6 Contoh *class diagram*

Pada gambar 2.6 merupakan *class diagram* dari proses kerja sama yang dihasilkan dari tabel pendefinisian struktur setiap tabel beserta *attribute* di *database*.

2.2.9. *Laravel*

Laravel merupakan sebuah kerangka kerja yang diadopsi dari bahasa pemrograman, *laravel* salah satu teknologi yang populer di Indonesia. *Laravel* salah satu kerangka kerja *backend* yang memiliki arsitektur *MVC* sehingga mempunyai struktur folder yang baik. *Laravel* sangat cocok digunakan untuk proyek dengan rentang waktu yang singkat karena *laravel* memiliki kenyamanan saat proses pengembangan dan

kemudahan dalam penggunaannya karena sudah bisa terintegrasi dengan banyak *library* [28].

2.2.10. React Js

React js merupakan *library* untuk pengembangan antarmuka pada *website*, *react js* dikembangkan menggunakan bahasa pemrograman *javascript* oleh *facebook*, *react js* salah satu *library* yang sangat populer dan memiliki komunitas yang besar. Kelebihan dari *react js* salah satunya adalah membuat antarmuka *website* menjadi interaktif dan dinamis. *React js* merupakan *library open source* sehingga memiliki banyak kontributor serta didukung banyak *library* lain [29].

2.2.11. Inertia Js

Inertia Js adalah *library* yang dibangun dari bahasa pemrograman *javascript* untuk membuat *Single Page Application (SPA)* tanpa membuat *API*, *inertia js* berperan sebagai jembatan atau penghubung antara *laravel* dan *react* agar bisa berkomunikasi secara langsung dalam satu proyek atau *folder* [30].

2.2.12. Single Page Application (SPA)

Single Page Application (SPA) adalah salah satu konsep pengembangan *website* yang digunakan untuk membuat *website* yang awalnya terdiri dari banyak halaman menjadi satu halaman. Penggunaan satu halaman ini merupakan bentuk optimasi pemrosesan halaman *website* agar menjadi interaktif dan dapat membuat konten dinamis tanpa melakukan pemuatan ulang halaman karena satu halaman bisa berganti-ganti komponen [31].

2.2.13. Tailwind CSS

Tailwind CSS adalah *library CSS* yang digunakan pengembangan antarmuka *website*. *Tailwind CSS* dilengkapi dengan *class-class CSS* yang

siap digunakan dan mendukung fitur *responsive* sehingga dapat mempersingkat pembuatan antarmuka pada *website* [32].

2.2.14. *Mysql*

Mysql adalah *database server* yang bertipe *relasional*, artinya data yang disimpan pada *database* ini dapat berelasi dengan tabel atau data lainnya. *Mysql* menggunakan bahasa *SQL* untuk menjalankan berbagai jenis operasi. *Mysql* salah satu *database* yang paling populer dan ramah untuk pemula karena mudah digunakan [33].

2.2.15. *Blackbox Testing*

Black box testing adalah salah satu pengujian fitur-fitur pada perangkat lunak. Target dari pengujian ini berfokus terhadap fungsionalitas tanpa menguji *design* dan kode, oleh karena itu setiap pengujian melibatkan satu fitur dengan cara mensimulasikan cara kerja fitur tersebut apakah sudah sesuai dengan kebutuhan *user*. Hasil dari pengujian ini adalah perbandingan dari harapan dan hasil percobaan dari fitur apakah sudah sesuai seperti yang diharapkan [34].

2.2.16. Pengukuran Efisiensi

Pengukuran efisiensi proses pada penerapan sistem dapat dilakukan dengan pengujian untuk membandingkan proses sebelumnya dengan proses setelah penerapan teknologi. Perbandingan proses bisa dibentuk dalam format tabel perbandingan.

Tabel 2. 14 Contoh perbandingan proses sebelum dan sesudah penerapan teknologi [35]

No.	Proses	Sebelum penerapan	Setelah penerapan
1.	Pencarian data <i>customer</i> .	Manual dikarenakan sebelum penerapan sistem masih berbentuk <i>hardfile</i> .	Mencari melalui <i>inputan search</i> berdasarkan kode dokumen

			dikarenakan dokumen sudah berbentuk <i>sofffile</i> .
2.	<i>Log</i> aktifitas data.	Belum ada <i>log</i> .	Sudah otomatis <i>log</i> jika ada aktifitas perubahan data.
3.	Pembuatan dokumen serah terima berita acara.	Memakan waktu 1 menit 30 detik.	Memakan waktu 1 menit.

Pada tabel 2.12 merupakan contoh perbandingan proses dari sebelum penerapan teknologi dan setelah penerapan teknologi. Untuk mengukur efisiensi waktu maka bisa menggunakan rumus:

$$\text{Peningkatan kecepatan waktu (\%)} = \frac{\text{waktu lama} - \text{waktu baru}}{\text{waktu lama}} \times 100\% \quad (2.1)$$

$$\text{Peningkatan kecepatan pembuatan dokumen serah terima berita (\%)} = \frac{90 - 60}{90} \times 100\% = 33\%$$

Dari hasil pengukuran kecepatan menggunakan rumus di atas maka diperoleh peningkatan sebesar 33% untuk proses pembuatan dokumen *invoice* [35].