

Pengembangan Sistem Informasi Penjualan Berbasis Web Dengan Metode Test Driven Development Pada CV IN-BOX Jepara

Nur Raditya Mahardika¹, Lalang Erawan^{*2}

Program Studi Sistem Informasi, Fakultas Ilmu Komputer, Universitas Dian Nuswantoro

Email: ¹nurradityamahardika@gmail.com, ²lalang.erawan@dsn.dinus.ac.id

*Penulis Korespondensi

Diterima: 8 September 2023; Direvisi: 19 Oktober 2023; Disetujui: 25 Mei 2024

Abstrak

Permasalahan penurunan akurasi dan kecepatan proses pengolahan data dalam suatu perusahaan atau organisasi seringkali disebabkan pemakaian teknologi informasi yang kurang optimal. Seperti yang dialami oleh CV INBOX Jepara dalam menangani proses penjualannya yang berjenis Make-to-Order atau penjualan berdasarkan pesanan. Penerapan aplikasi spreadsheet untuk mengolah data kegiatan penjualan semula berjalan lancar. Seiring waktu dengan bertambahnya jumlah data dan pesanan yang harus dilayani, karyawan perusahaan menjadi kewalahan. Penyelesaian pesanan berjalan lambat karena proses loading file spreadsheet yang semakin lama. Proses pengambilan informasi yang dibutuhkan, perubahan data, dan pembuatan laporan mengalami penundaan yang semestinya tidak perlu terjadi. Solusi yang logis terhadap permasalahan perusahaan ini adalah dengan mengembangkan sebuah sistem informasi untuk menangani proses penjualan perusahaan. Sistem dibuat berbasis web dengan pertimbangan kesiapan infrastruktur perusahaan yang sudah memiliki koneksi jaringan internet sendiri di unit-unit perusahaan yang tersebar di beberapa lokasi. Selain itu, pimpinan perusahaan memiliki mobilitas tinggi karena sering mengunjungi unit-unit perusahaan sementara pada saat yang sama memerlukan akses ke sistem. Metode pengembangan sistem yang dipilih adalah metode Test Driven Development yang sesuai untuk pengembangan sistem skala kecil dan sistem yang dihasilkan relatif lebih baik dalam hal jumlah kesalahan. Sistem diuji dengan menggunakan alat pengujian Black Box untuk memastikan sistem berjalan sesuai yang diharapkan. Sistem penjualan berbasis web ini berhasil dikembangkan dan diharapkan dapat diimplementasikan untuk mengatasi permasalahan perusahaan saat ini.

Kata kunci: test driven development, Perancangan Sistem, sistem penjualan, make-to-order

Abstract

The problem of declining accuracy and processing speed in data processing within a company or organization is often caused by suboptimal use of information technology. This is evident in the case of CV INBOX Jepara, dealing with its Make-to-Order sales processes or sales based on orders. The initial implementation of a spreadsheet application for sales data processing proceeded smoothly. However, as the volume of data and orders to be handled increased over time, company employees became overwhelmed. Order fulfillment slowed down due to the prolonged loading process of the spreadsheet file. The retrieval of necessary information, data manipulation, and report generation experienced unnecessary delays. A logical solution to this company's problem is to develop an information system to handle the sales processes. The system is web-based, considering the company's existing infrastructure with its own internet network connections in various locations. Additionally, company leaders have high mobility, frequently visiting different units while requiring access to the system simultaneously. The chosen system

development method is Test Driven Development, suitable for small-scale system development and known for producing relatively fewer errors. The system undergoes testing using Black Box testing tools to ensure it operates as expected. This web-based sales system has been successfully developed and is expected to be implemented to address the current challenges faced by the company

Keywords: *test driven development, system design, sales system, make-to-order*

1. PENDAHULUAN

Teknologi informasi saat ini membawa dampak perubahan yang signifikan terhadap cara perusahaan menjalankan bisnisnya. Informasi telah menjadi aset yang berharga, perusahaan yang mampu memahami dan memanfaatkan teknologi informasi dengan baik akan memperoleh keuntungan pertumbuhan yang signifikan. Teknologi informasi akan membuat perusahaan dapat menanggapi perubahan pasar dengan cepat, memperbaiki layanan terhadap pelanggan, dan bahkan dapat menciptakan peluang-peluang bisnis baru melalui inovasi. Teknologi informasi juga memungkinkan perusahaan dapat memiliki persediaan data dalam jumlah besar yang dapat digunakan perusahaan untuk mendukung proses pengambilan keputusan lebih efektif. Selain itu, dengan teknologi informasi perusahaan dapat merencanakan strategi yang lebih baik untuk bersaing dengan para kompetitornya ditengah persaingan pasar yang semakin ketat [1].

Di Jepara, terdapat sebuah perusahaan manufaktur yang bergerak di bidang pembuatan berbagai kemasan atau kotak untuk berbagai keperluan. Produk yang dihasilkan perusahaan bernama CV. IN-BOX JEPARA ini berupa *packaging box* untuk perusahaan mebel. Dalam menjalankan bisnisnya perusahaan menggunakan strategi *Make-to-Order* (MTO) [2]. Konsumen yang memesan produk perusahaan, dapat memesan kemasan sesuai dengan spesifikasi mereka. [2]Proses pemesanan dimulai dari pembeli memberikan dokumen *purchase order* (PO) ke bagian marketing CV. IN-BOX JEPARA, dokumen tersebut dapat dikirim melalui email atau whatsapp. Dalam prakteknya, perusahaan juga akan memberikan bantuan dalam pengisian PO jika terdapat pembeli baik non badan hukum atau perorangan yang tidak terbiasa menggunakan dokumen PO. Dokumen PO tersebut oleh bagian marketing akan diperiksa untuk menentukan apakah pesanan tersebut dapat disetujui atau tidak berdasarkan *delivery date* dan spesifikasi yang diminta. Jika PO diterima maka setiap item yang ada di PO beserta data-data lainnya akan dicatat dalam sebuah file spreadsheet serta diberi nomor *sales order*.

Aplikasi spreadsheet masih banyak digunakan dalam pengelolaan data bisnis berbagai perusahaan. Jenis aplikasi ini dijadikan pilihan alat pengolahan data di lingkungan bisnis karena tidak memerlukan pelatihan intensif dan dapat dipelajari lalu digunakan dalam waktu singkat[3]. CV. IN-BOX JEPARA telah cukup lama menggunakan aplikasi ini untuk mengolah data bisnisnya. Seiring pertumbuhan perusahaan, data yang harus diolah semakin banyak. Beberapa waktu terakhir, masalah yang berhubungan dengan penggunaan aplikasi ini mulai muncul. Pemrosesan data membutuhkan waktu yang semakin lama. Proses-proses seperti pemerolehan informasi PO atau *sales order*, proses pembukaan file, proses koreksi data, berjalan dengan sangat lambat jika dibandingkan dengan jumlah pesanan yang harus diproses. Ukuran file membengkak yang membuat proses pemanggilan file menjadi lama. Penambahan waktu proses data yang tidak seharusnya terjadi ini menyebabkan kegiatan bisnis perusahaan menjadi terganggu. Pimpinan menjadi sulit memperoleh laporan dalam waktu singkat. Masalah ini jika dibiarkan dan diabaikan akan membuat perusahaan mengalami kerugian karena pekerjaan yang seharusnya cepat menjadi lambat, sehingga kinerja perusahaan dalam memenuhi pesanan menurun. Kinerja menurun mengakibatkan ketidakpuasan pelanggan semakin besar. Tentunya kondisi ini tidak diinginkan oleh setiap perusahaan yang sangat berkepentingan dengan laju pendapatan dan kelangsungan hidup perusahaan dalam jangka panjang.

Penerapan sistem informasi sudah tidak asing lagi bagi organisasi dan perusahaan yang berpandangan maju. Sistem informasi memiliki kemampuan untuk mengintegrasikan dan

mengolah data secara cepat dan akurat. Selain itu, penggunaan waktu dan biaya dapat lebih efisien yang dalam jangka panjang dapat menguntungkan perusahaan secara ekonomis. Keakuratan sistem informasi dapat mengurangi risiko kesalahan manusia dalam pengolahan data. Kecepatan pengolahan data yang diberikannya dapat menyelesaikan permasalahan lamanya waktu pemrosesan data dan meminimalkan keterlambatan informasi. Bersama dengan teknologi informasi, sistem informasi dapat memberikan media komunikasi dan koordinasi antar bagian dalam organisasi secara lebih efektif dan efisien. Pertukaran data dan informasi dapat berjalan lebih cepat, aman, dan efektif [4].

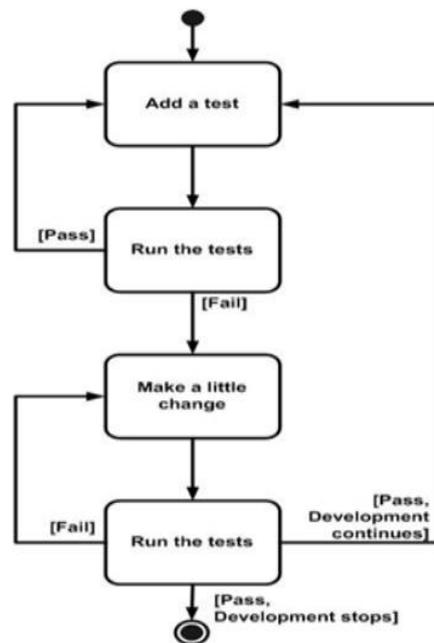
Berdasarkan karakteristik dan kemampuan sistem informasi, permasalahan yang sedang dihadapi oleh CV. IN-BOX JEPARA dapat diselesaikan dengan mengimplementasikan sebuah sistem informasi yang akan mendukung proses bisnis perusahaan. Sistem informasi yang akan diterapkan termasuk kategori sistem informasi penjualan yaitu sebuah sistem yang didalamnya terdapat berbagai prosedur untuk melakukan pembuatan dokumen, mencatat, dan mengkalkulasi, serta menghasilkan informasi penjualan untuk keperluan manajemen dan bagian lain yang berkepentingan. Sistem informasi penjualan merupakan salah satu sub sistem informasi bisnis. Sub sistem lainnya seperti pemasaran, sumber daya manusia, keuangan dan akuntansi serta manufaktur produksi [5].

Sistem informasi penjualan yang dikembangkan akan berbasis web. Hal ini didasarkan beberapa pertimbangan antara lain kebutuhan akses sejumlah karyawan CV.IN-BOX JEPARA yang dalam melakukan pekerjaannya harus berada di lokasi di luar perusahaan, dan direktur perusahaan yang memiliki mobilitas tinggi karena sering memantau secara *on-site* beberapa unit perusahaan lainnya yang tersebar di Kabupaten Jepara. Fakta lapangan lain yang menjadi bahan pertimbangan adalah dari sisi infrastruktur yang dimiliki perusahaan. Meskipun masing-masing unit perusahaan tidak memiliki infrastruktur interkoneksi antar lokasi unit perusahaan, tetapi mereka memiliki infrastruktur internet. Dengan demikian, alternatif sistem informasi penjualan berbasis web menjadi relevan dengan kondisi infrastruktur. Ditambah dengan kepemilikan server web *on-premise*, penerapan sistem berbasis web dalam perusahaan tidak akan membutuhkan biaya yang terlalu besar.

Seperti halnya pengembangan sebuah sistem informasi, pengembangan sistem informasi penjualan berbasis web ini memerlukan metode pengembangan sistem tertentu agar dapat melakukan pengembangan sistem secara terencana dan sistematis. Metode pengembangan sistem yang akan digunakan adalah metode *Test Driven Development* (TDD) [6]. Metode ini merupakan turunan dari metode pengembangan perangkat lunak *agile*. Langkah-langkah pengembangan sistem dalam metode ini meliputi penentuan dan pembuatan unit tes, penulisan kode program yang dapat lolos test tersebut, dan refactoring kode program sesuai kebutuhan. Metode ini cocok untuk proyek skala kecil serta memiliki keunggulan menghasilkan kualitas kode program yang baik karena kode program wajib lolos pemeriksaan kesalahan menggunakan unit test. Dalam prakteknya, pemrogram hanya akan membuat satu fitur yang telah direncanakan dalam satu waktu. Hal ini dimaksudkan agar pemrogram terfokus hanya pada sebuah fitur sehingga diharapkan fitur demi fitur ditambahkan secara bertahap dan terfokus dengan jumlah kesalahan yang lebih minimal. Hal lain yang menjadi perhatian dalam pengembangan fitur adalah menjaga kompatibilitas antar fitur yang akan mengurangi potensi kemunculan kesalahan.

2. METODE

Metode pengembangan sistem TDD memiliki strategi yang menempatkan pengujian sebagai bagian utama dalam proses pengembangan. Tujuan dari TDD adalah semua kode diverifikasi dengan benar dan memberikan kepastian kepada pemrogram dengan mengurangi kompleksitas kode. Hal ini didasarkan kepada hasil penelitian yang menyatakan bahwa kualitas kode akan meningkat jika produktivitas pemrogram menurun dan sebaliknya [6]. Penerapan metode ini diawali dengan proses pengumpulan data persyaratan sistem.



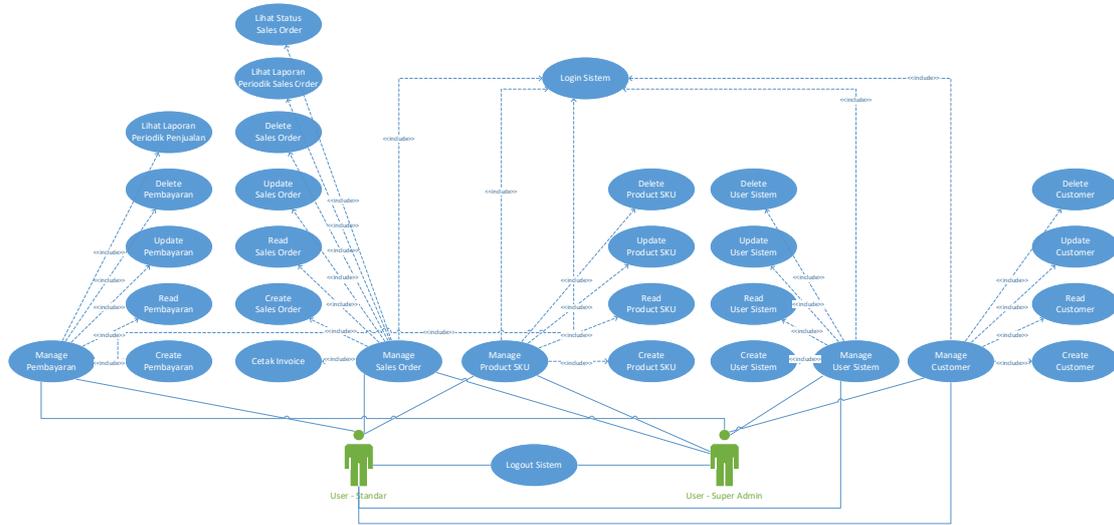
Gambar 1. Tahapan Metode *Test Driven Development*

Gambar 1 diatas merupakan proses inti dari TDD. Pada dasarnya proses ini memastikan bahwa kode program harus lolos pengujian menggunakan unit tes yang telah dibuat berdasarkan fitur-fitur yang telah direncanakan. Identifikasi Kebutuhan dan Fitur ditandai dengan menentukan kebutuhan dan fitur dari sistem informasi yang akan dikembangkan lalu fitur-fitur ditentukan perilakunya. Alat yang digunakan pada tahap ini adalah diagram UML[7] dan BPMN[8]. Pilih unit atau komponen yang akan dikembangkan terlebih dahulu. Unit dapat berupa fungsi, kelas, atau modul kecil yang dapat diuji secara terisolasi. Tulis unit tes berupa pembuatan kelas tes yang sesuai dengan unit atau komponen yang akan dikembangkan. Lalu tulis skenario tes yang menjelaskan bagaimana unit atau komponen seharusnya berperilaku. Selain itu, perlu dijelaskan situasi input dan hasil yang diharapkan. Setelah tes siap maka dijalankan. Pastikan tes mengalami kegagalan karena implementasi fungsionalitas belum ada. Pada tahapan implementasi membuat fitur atau fungsi dengan cara sederhana yang paling memungkinkan untuk memenuhi kebutuhan tes. Dalam pembuatan fitur ini difokuskan pada usaha untuk membuat tes berubah dari gagal menjadi berhasil. Setelah fitur siap, jalankan kembali unit tes. Jika tes berhasil, maka menandakan bahwa implementasi awal sudah memenuhi persyaratan.

Kemudian melakukan refactoring atau perbaikan kode jika diperlukan untuk meningkatkan kejelasan atau efisiensi tanpa mengubah perilaku yang diuji (fitur). Dalam hal ini pastikan bahwa tes masih berhasil dijalankan setelah refactoring. Pilih unit atau komponen yang akan dikembangkan berikutnya. Kemudian buat tes baru untuk fitur atau fungsionalitas berikutnya. Lanjutkan langkah-langkah implementasi dan refactoring seperti sebelumnya.'

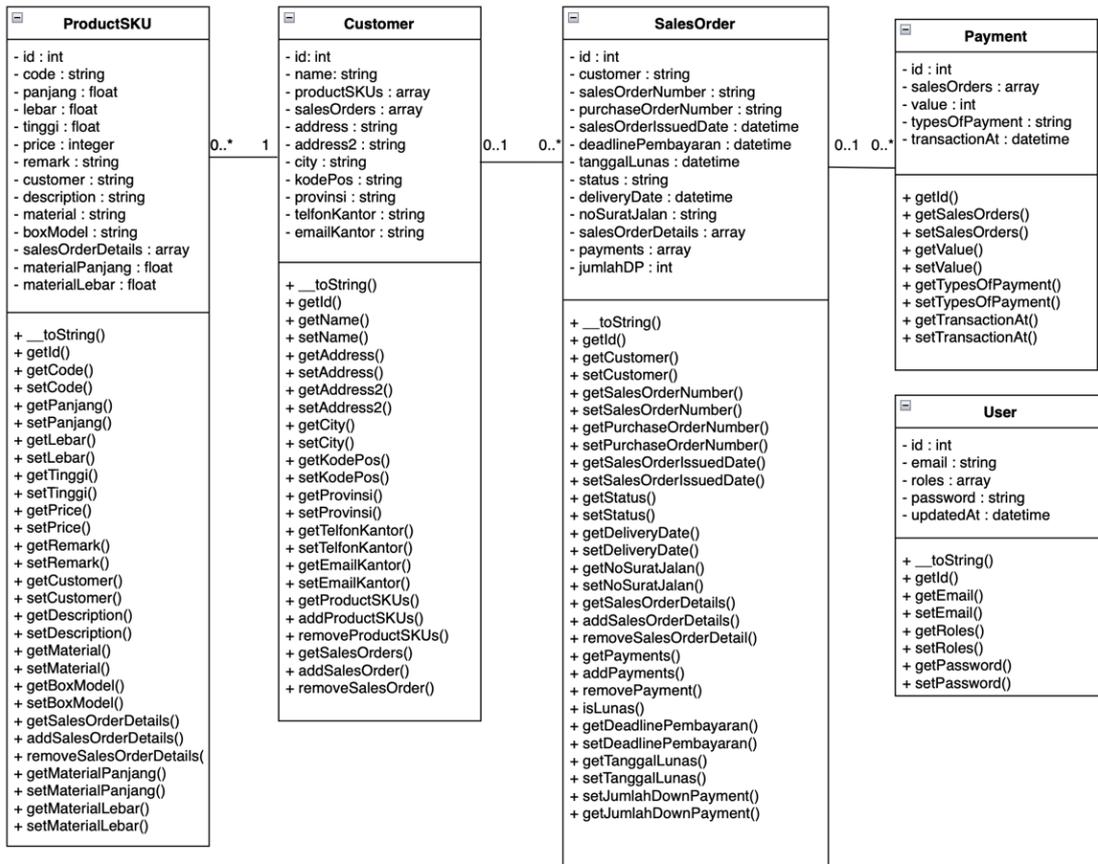
Pada tahapan Tes Perangkat Lunak ini sistem informasi yang telah dibuat diuji oleh bagian kontrol kualitas agar *bug* yang ditemukan bisa segera diperbaiki agar kualitas perangkat lunak terjaga. Pada penelitian ini metode pengujian yang digunakan berupa metode Black Box Testing[9]. Pada tahap ini akan disusun skenario pengujian yang akan diterapkan terhadap fitur-fitur yang sudah dikembangkan. Skenario pengujian meliputi jenis input data yang akan digunakan dan bentuk output yang diharapkan. Selanjutnya dilakukan pengujian berdasarkan skenario yang telah disusun. Hasil pengujian kemudian dianalisis, apakah input yang diberikan sudah memberikan output yang seperti yang diharapkan. Setelah proses pengujian selesai dilakukan, langkah terakhir adalah mendokumentasikan sistem informasi yang telah dikembangkan dengan tujuan untuk kebutuhan perawatan sistem di kemudian hari.

dapat mengelola sistem secara keseluruhan yang dalam hal ini adalah user Super Admin. Jenis user ini akan dapat mengelola data SO, produk SKU, user sistem, dan kustomer.



Gambar 3. Kebutuhan Fungsional Sistem

3.3 Database Sistem



Gambar 4. Rancangan Database Sistem

Data yang dikelola sistem akan disimpan secara terstruktur dalam sebuah database yang akan dibuat berdasarkan rancangan kelas-kelas dalam diagram kelas yang ditunjukkan pada dalam gambar 4. Pada diagram tersebut terdapat 5 kelas yang dirancang yaitu ProductSKU, Customer, SalesOrder, Payment. Dan User. Kelas ProductSKU berisi informasi tentang produk pesanan customer, kelas Customer menyediakan informasi detail customer, SalesOrder berisi informasi tentang transaksi penjualan, Payment menyimpan data pembayaran, dan User berisi informasi tentang data user.

3.4 Implementasi

Pada tahap pengembangan sistem informasi penjualan berbasis web ini, metode yang diterapkan ialah metode Test Driven Development, sebagai prosedur yang ditempuh dalam tahap pengembangan sistem informasi penjualan pada CV. IN-BOX JEPARA. Berikut langkah-langkah yang dilakukan:

a) Menambah Unit Test

Pada tahap ini dilakukan pembuatan unit tes sesuai fitur yang diminta, spesifikasi fitur tersebut didapatkan berdasarkan Use Case yang dibuat dengan *Unified Modeling Language* (UML) seperti Use Case *Create, Read, Update, Delete Sales Order*. Pembuatan unit tes, yaitu pengujian perangkat lunak yang menguji setiap unit atau komponen dari sebuah perangkat lunak, yang dilakukan dengan membuat kelas *Sales Order* yang meng-*extends* kelas pada *framework test* yang dipakai, yaitu *PHPunit*, didalam *class* tersebut akan dibuat *method* yang berisi *object initialization*, serta logika dan kondisi yang diharapkan.

Dalam tes ini berisi fungsi seperti *getter* dan *setter* sesuai rancangan kelas pada *Class Diagram*. Dalam isi tes tersebut melakukan pemeriksaan apakah setiap fungsi jika diberi suatu nilai dapat menghasilkan output seperti yang diinginkan. Untuk melakukan pemeriksaan nilai tersebut menggunakan fungsi dari *class PHPUnit* yaitu *assertSame()*, fungsi tersebut juga diberi dua argumen, pertama adalah nilai keluaran yang diharapkan, kedua adalah memanggil fungsi yang sedang dites.

b) Menjalankan tes untuk memeriksa apakah ada tes baru yang gagal

Dalam tahap ini dijalankan semua tes dengan cara menjalankan perintah “*phpunit*” pada terminal perintah sistem operasi. Setelah dijalankan maka akan keluar *error* beserta penjelasan kenapa *error* tersebut muncul, dalam contoh ini terdapat *error* “*Class 'App\Entity\SalesOrder' Not Found*”, *error* tersebut muncul karena *class* tersebut memang belum ditulis, sehingga *error* ini digunakan sebagai petunjuk untuk tahap berikutnya, yaitu menulis kode program.

c) Menulis Kode Program

Berikut adalah potongan salah satu kode program yang dibuat setelah pembuatan unit tes:

```
<?php
use Doctrine\Common\Collections\ArrayCollection;
use Doctrine\Common\Collections\Collection;
use Doctrine\ORM\Mapping as ORM;

class SalesOrder
{
    #[ORM\Id] #[ORM\GeneratedValue] #[ORM\Column(type: 'integer')] private
    $id;
    #[ORM\ManyToOne(targetEntity: Customer::class, inversedBy:
    'salesOrders')] private $customer;
    #[ORM\Column(type: 'string', length: 255)] private $salesOrderNumber;
    #[ORM\Column(type: 'string', length: 255, nullable: true)] private
    $purchaseOrderNumber;
    #[ORM\Column(type: 'datetime')] private $salesOrderIssuedDate;
    #[ORM\ManyToOne(targetEntity: SalesOrderStatus::class, inversedBy:
    'salesOrders')] private $status;
```

```

#[ORM\Column(type: 'datetime', nullable: true)] private $deliveryDate;
#[ORM\Column(type: 'string', length: 255, nullable: true)] private
    $noSuratJalan;

#[ORM\OneToMany(mappedBy: 'salesOrder', targetEntity:
    SalesOrderDetail::class, cascade: ['persist'], orphanRemoval: true)]
private $salesOrderDetails;
#[ORM\OneToMany(mappedBy: 'salesOrders', targetEntity:
    Payment::class)] private $payments;
public function getId(): ?int{return $this->id;}
public function getCustomer(): ?Customer{return $this->customer;}
public function setCustomer(?Customer $customer): self{$this->customer
    = $customer;return $this;}
public function getSalesOrderNumber(): ?string{return $this-
    >salesOrderNumber;}
public function setSalesOrderNumber(string $salesOrderNumber):
    self{$this->salesOrderNumber = $salesOrderNumber;
    return $this;}
public function getPurchaseOrderNumber(): ?string{return $this-
    >purchaseOrderNumber;}
public function setPurchaseOrderNumber(?string $purchaseOrderNumber):
    self{$this->purchaseOrderNumber = $purchaseOrderNumber;
    return $this;}
public function getSalesOrderIssuedDate(): ?\DateTimeInterface{return
    $this->salesOrderIssuedDate;}
public function setSalesOrderIssuedDate(\DateTimeInterface
    $salesOrderIssuedDate): self{
    $this->salesOrderIssuedDate = $salesOrderIssuedDate;
    return $this;}
public function getStatus(): ?SalesOrderStatus{return $this->status;}
public function setStatus(?SalesOrderStatus $status): self{$this-
    >status = $status;return $this;}
public function getDeliveryDate(): ?\DateTimeInterface{return $this-
    >deliveryDate;}
public function setDeliveryDate(?\DateTimeInterface $deliveryDate):
    self{$this->deliveryDate = $deliveryDate;
    return $this;}

```

Pengembang melakukan pemrograman terhadap kode sesuai konteks tes yang dibuat sebelumnya. *Error* yang muncul dalam tahap tes yang gagal sebelumnya juga menjadi acuan untuk melakukan pemrograman logika, penamaan *class*, dan *function* pada kode program. Dalam pemrograman *class* yang dibuat juga dibantu dengan *framework* PHP, sehingga *function* yang diprogram, memiliki output yang diharapkan pada tes.

Pada contoh kali ini *class Sales Order* dibuat menggunakan *library Object Relational Mapping Doctrine*, sehingga pada awal *class* perlu menggunakan perintah “*use Doctrine*” untuk memanggil *library* tersebut, kemudian dibuat *class* dengan nama *SalesOrder* yang berisi *private variable*, yaitu variabel yang hanya dapat diakses oleh *class* dirinya sendiri. Kemudian dibuatkan *function getter* dan *setter* sesuai pada *error* yang muncul pada tahap menjalankan tes sebelumnya.

d) Menjalankan Tes Lagi

Setelah pengkodean unit tes yang dibutuhkan, selanjutnya menjalankan tes lagi. Tes dijalankan lagi untuk memeriksa apakah kode yang sudah ditulis dapat bekerja dengan baik dan tidak mengakibatkan *error* pada tes yang sebelumnya sudah lolos, proses ini dilakukan dengan mengirim perintah “*phpunit*”. Dengan kata lain kode yang baru berjalan sesuai spesifikasi dan tidak merusak kode-kode yang sudah ada. Jika test gagal, maka proses menulis kode program akan diulang sampai lolos test. Pada contoh ini test kali ini lolos semua.

e) Melakukan *Refactor* Kode Program

Berikut contoh kode yang perlu dilakukan *refactor*:

```

class ProductMaterialFixtures extends Fixture
{
    public function load(ObjectManager $manager): void
    {
        $material1 = new ProductMaterial();
        $material2 = new ProductMaterial();
        $material3 = new ProductMaterial();
        $material4 = new ProductMaterial();
        $material5 = new ProductMaterial();
        $material6 = new ProductMaterial();
        $material7 = new ProductMaterial();
        $material8 = new ProductMaterial();
        $material9 = new ProductMaterial();
        $material10 = new ProductMaterial();
        $material11 = new ProductMaterial();

        $material1->setName("K/M 150");
        $material2->setName("D/K 275");
        $material3->setName("D/K 200");
        $material4->setName("K/M 150-PARTISI");
        $material5->setName("K/M 150-CORNER");
        $material6->setName("D/K 150");
        $material7->setName("3 PLY");
        $material8->setName("K/M");
        $material9->setName("K/M 125");
        $material10->setName("3 PLY-150");
        $material11->setName("T/K 275");

        $manager->persist($material1);
        $manager->persist($material2);
        $manager->persist($material3);
        $manager->persist($material4);
        $manager->persist($material5);
        $manager->persist($material6);
        $manager->persist($material7);
        $manager->persist($material8);
        $manager->persist($material9);
        $manager->persist($material10);
        $manager->persist($material11);

        $this->addReference('mat1', $material1);
        $this->addReference('mat2', $material2);
        $this->addReference('mat3', $material3);
        $this->addReference('mat4', $material4);
        $this->addReference('mat5', $material5);
        $this->addReference('mat6', $material6);
        $this->addReference('mat7', $material7);
        $this->addReference('mat8', $material8);
        $this->addReference('mat9', $material9);
        $this->addReference('mat10', $material10);
        $this->addReference('mat11', $material11);

        $manager->flush();
    }
}

```

Pada kode ini terdapat duplikasi karena setiap pemanggilan fungsi “*setName*, *persist*, *addReference*”, kode ini melakukan perulangan dengan cara membuat variabel tersendiri sehingga kurang efektif. Kode ini dapat dilakukan *refactor* sehingga menjadi seperti berikut:

```

class ProductMaterialFixtures extends Fixture
{
    public function load(ObjectManager $manager): void
    {
        $i = 1;

```

```

$materialLists = [
    "K/M 150", "D/K 275", "D/K 200",
    "K/M 150-PARTISI", "K/M 150-CORNER",
    "D/K 150", "3 PLY", "K/M",
    "K/M 125", "3 PLY-150", "T/K 275",
];

foreach ($materialLists as $item) {
    $material = new ProductMaterial();
    $material->setName($item);

    $manager->persist($material);
    $this->addReference("mat-" . $i++, $material);
}

$manager->flush();
}
}

```

3.5 Hasil Implementasi

Hasil implementasi menjelaskan sistem informasi penjualan yang telah dibangun dengan menggunakan metode *Test Drive Development*.

a) Halaman Login

Pada gambar 5 halaman login tidak berbeda jauh dengan sistem-sistem yang lain, terdiri dari masukan nama user yang berupa email dari user dan password.

Gambar 5. Halaman Login Sistem

b) Halaman Indeks Sales Order

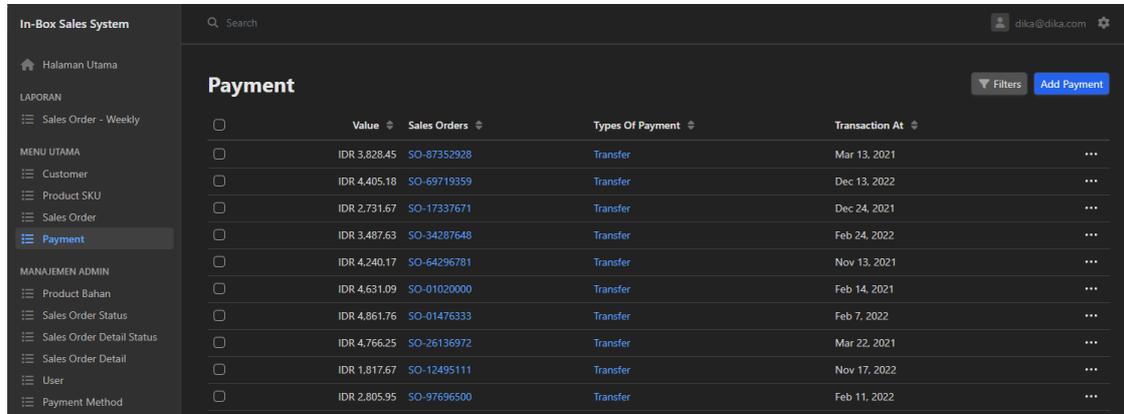
Halaman yang mencatat pesanan penjualan, memberikan informasi customer, nomor SO, nomor PO, tanggal, status, tanggal pengiriman, nomor surat jalan, dan SKU produk yang ditampilkan pada gambar 6 berikut.

Customer	Sales Order Number	Purchase Order Number	Sales Order Issued Date	Status	Delivery Date	No Surat Jalan	Product SKU
Kevin Warren	SO-23428100	aZVWJzu1rz	Jul 10, 2021	Dalam Antrian	Sep 22, 2023	rL9NX0hSG2	CD-BD-0011-H, BC-CA-7419-I, DE...
Amber Gomez	SO-49321920	cueYfmujfh	Jan 27, 2022	Sablon	Aug 30, 2023	oCUhm2mz8O	DF-AD-9962-Q, EC-CB-9226-Q, BA...
Jonathan Chavez	SO-69938960	07ew5eRRa4	Jan 14, 2023	Sablon	Sep 30, 2023	SHcbJsRr85	CC-BA-0944-C, DE-BB-7838-M, CB...
Carol Graham	SO-89240783	uHoltpbmyb	Dec 18, 2021	Dalam Antrian	Jun 12, 2022	fXTNs49Cqc	FE-BE-7392-N, FC-BF-2922-I, DF...
Mike Ross	SO-01936647	Tn600SjDTU	Nov 28, 2022	Plong	May 16, 2021	rFB09Qpu2	EE-FE-3727-O, EC-CA-1292-E, AC...

Gambar 6. Halaman Indeks Sales Order

c) Halaman Indeks Payment

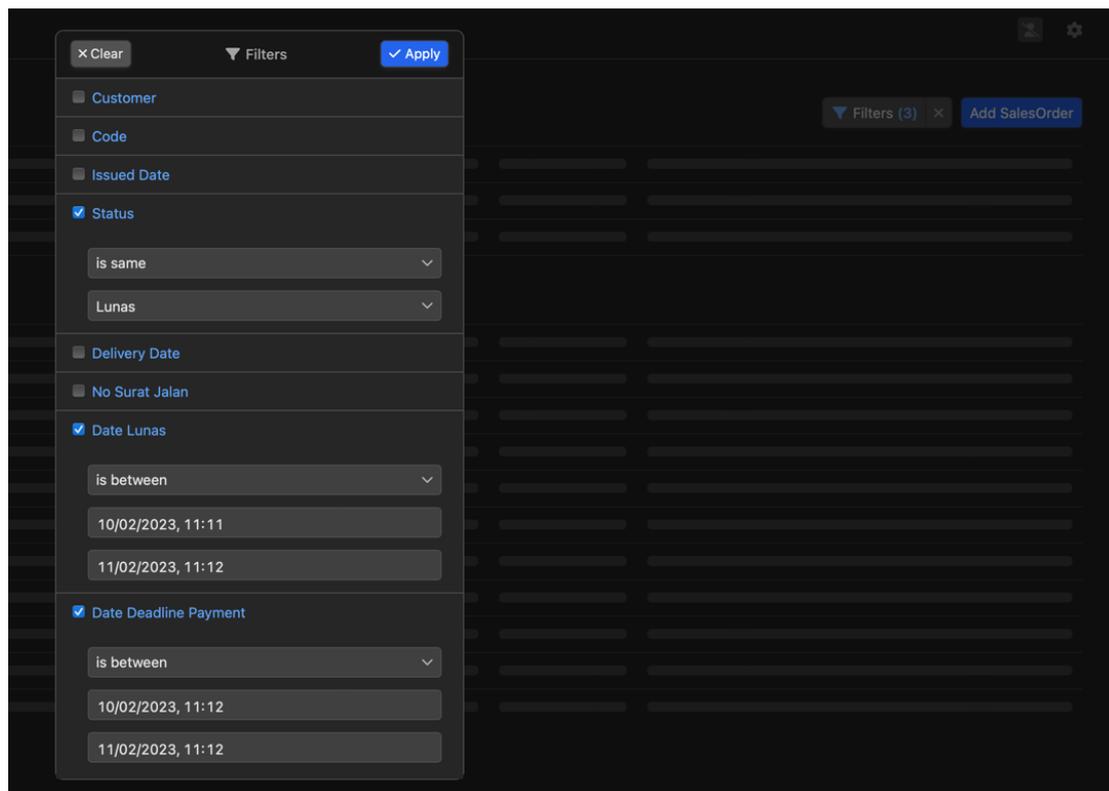
Gambar 7 menampilkan Halaman yang menyimpan informasi pembayaran meliputi nilai transaksi, nomor SO, jenis pembayaran, tanggal transaksi



	Value	Sales Orders	Types Of Payment	Transaction At	
<input type="checkbox"/>	IDR 3.828.45	SO-87352928	Transfer	Mar 13, 2021	...
<input type="checkbox"/>	IDR 4.405.18	SO-69719359	Transfer	Dec 13, 2022	...
<input type="checkbox"/>	IDR 2.731.67	SO-17337671	Transfer	Dec 24, 2021	...
<input type="checkbox"/>	IDR 3.487.63	SO-34287648	Transfer	Feb 24, 2022	...
<input type="checkbox"/>	IDR 4.240.17	SO-64296781	Transfer	Nov 13, 2021	...
<input type="checkbox"/>	IDR 4.631.09	SO-01020000	Transfer	Feb 14, 2021	...
<input type="checkbox"/>	IDR 4.861.76	SO-01476333	Transfer	Feb 7, 2022	...
<input type="checkbox"/>	IDR 4.766.25	SO-26136972	Transfer	Mar 22, 2021	...
<input type="checkbox"/>	IDR 1.817.67	SO-12495111	Transfer	Nov 17, 2022	...
<input type="checkbox"/>	IDR 2.805.95	SO-97696500	Transfer	Feb 11, 2022	...

Gambar 7. Halaman Indeks Payment

d) Halaman Filter Sales Order



Gambar 8. Halaman Filter Sales Order

Gambar 8 diatas merupakan tampilan menu filter pada halaman *Index Sales Order* yang diakses oleh aktor yang memiliki hak. Pada menu ini dapat digunakan sebagai *filter* untuk menampilkan laporan *Sales Order* sesuai *filter* yang dipilih untuk mendukung Use Case ‘Lihat Laporan Periodik Order dan Penjualan’, ‘Lihat Laporan Periodik Deadline Pembayaran’, ‘Lihat Laporan Periodik Order Status Lunas’. Jika tombol *Apply* ditekan maka akan muncul laporan yang sesuai kriteria.

4. KESIMPULAN

Metode Test Driven Development yang digunakan untuk mengembangkan sistem dalam penelitian ini memungkinkan pihak pengembang menghasilkan sistem yang lebih teruji sesuai dengan tujuan dari metode ini yang mengupayakan sistem yang selalu teruji di sepanjang proses pengembangannya. Sistem informasi penjualan berbasis web yang dihasilkan diharapkan dapat membantu menyelesaikan permasalahan kelambatan proses dan ketidakakuratan pengelolaan data yang dialami perusahaan saat ini. Upaya lebih lanjut perlu dilakukan untuk menerapkan sistem hasil penelitian ini di lapangan.

DAFTAR PUSTAKA

- [1] R. Anggraeni and I. Elan Maulani, "Pengaruh Teknologi Informasi Terhadap Perkembangan Bisnis Modern," *J. Sos. Teknol.*, vol. 3, no. 2, pp. 94–98, 2023, doi: 10.59188/jurnalsostech.v3i2.635.
- [2] J. Wan, X. Li, H. N. Dai, A. Kusiak, M. Martinez-Garcia, and D. Li, "Artificial-Intelligence-Driven Customized Manufacturing Factory: Key Technologies, Applications, and Challenges," *Proc. IEEE*, vol. 109, no. 4, pp. 377–398, 2021, doi: 10.1109/JPROC.2020.3034808.
- [3] M. R. Satria and A. P. Fatmawati, "Penyusunan Laporan Keuangan Perusahaan Menggunakan Aplikasi Spreadsheet," *Fair Value J. Ilm. Akunt. dan Keuang.*, vol. 3, no. 2, pp. 320–338, 2021, doi: 10.32670/fairvalue.v3i2.146.
- [4] R. S. Naibaho, "Pengenalan Teknologi Informasi," *Warta*, no. 52, p. 45, 2017.
- [5] A. Selay, G. Dwight, M. A. Alfarizi, and Dkk, "Sistem Informasi Penjualan," *Karimah Tauhid*, vol. 2, no. 1, pp. 232–237, 2023.
- [6] M. M. Moe and K. K. Oo, "Evaluation of Quality, Productivity, and Defect by applying Test-Driven Development to perform Unit Tests," *2020 IEEE 9th Glob. Conf. Consum. Electron. GCCE 2020*, pp. 435–436, 2020, doi: 10.1109/GCCE50665.2020.9291950.
- [7] T. B. Kurniawan and Syarifuddin, "Perancangan Sistem Aplikasi Pemesanan Makanan dan Minuman Pada Cafeteria NO Caffe di Tanjung Balai Karimun Menggunakan Bahasa Pemrograman PHP dan MySQL," *J. Tikar*, vol. 1, no. 2, pp. 192–206, 2020.
- [8] I. Ismanto, F. Hidayah, and K. Charisma, "Pemodelan Proses Bisnis Menggunakan Business Process Modelling Notation (BPMN) (Studi Kasus Unit Penelitian dan Pengabdian Kepada Masyarakat (P2KM) Akademi Komunitas Negeri Putra Sang Fajar Blitar)," *Briliant J. Ris. dan Konseptual*, vol. 5, no. 1, p. 69, 2020, doi: 10.28926/briliant.v5i1.430.
- [9] I. R. Dhaifullah, M. Muttanifudin H, A. Ananda Salsabila, and M. Ainul Yaqin, "Survei Teknik Pengujian Software," *J. Autom. Comput. Inf. Syst.*, vol. 2, no. 1, pp. 31–38, 2022, doi: 10.47134/jacis.v2i1.42.
- [10] L. Setiyani and E. Tjandra, "Analisis Kebutuhan Fungsional Aplikasi Penanganan Keluhan Mahasiswa Studi Kasus: Stmik Rosma Karawang," *J. Inov. Pendidik. dan Teknol. Inf.*, vol. 2, no. 1, pp. 8–17, 2021, doi: 10.52060/pti.v2i01.465.