

# Model Waterfall Pada Sistem Informasi Pemantauan dan Rekapitulasi Tagihan Vendor

Januar Rama Fitra<sup>1</sup>, Kresna Ramanda<sup>2\*</sup>, Arief Rusman<sup>3</sup>

<sup>1,3</sup> Universitas Nusa Mandiri

e-mail: <sup>1</sup>[januar12161193@nusamandiri.ac.id](mailto:januar12161193@nusamandiri.ac.id), <sup>3</sup>[arief.aef@nusamandiri.ac.id](mailto:arief.aef@nusamandiri.ac.id)

<sup>2</sup> Universitas Bina Sarana Informatika

e-mail: <sup>2</sup>[kresna.kra@bsi.ac.id](mailto:kresna.kra@bsi.ac.id)

## Abstrak

Permasalahan pemantauan riwayat transaksi jual beli antara suatu perusahaan dengan vendor tidak dilakukan dengan efektif, sehingga menimbulkan kasus pembayaran berulang yang dapat merugikan sebuah perusahaan. Permasalahan utama adalah tidak ada pemantauan atau rekapitulasi terhadap semua transaksi yang pernah dilakukan. Sistem informasi pemantau tagihan vendor dibutuhkan untuk penyelesaian masalah tersebut. Model Waterfall diterapkan dalam pengembangan sistem informasi, dimulai dari menganalisa kebutuhan sistem informasi, kemudian membuat sebuah desain baik berupa antarmuka pengguna maupun desain basis data, melakukan pengembangan sistem dengan media bahasa pemrograman yang menerjemahkan desain dengan komputer, proses ujicoba dan perawatan merupakan tahapan terakhir yang dilakukan pada metode pembuatan sistem informasi pemantau tagihan vendor. Membuat sebuah proses bisnis yang baik sehingga secara tidak langsung akan membantu memudahkan proses pemantauan. Masih terdapat beberapa fitur yang harus dilakukan perbaikan. Sistem informasi mampu melakukan pemantauan terhadap faktur jual beli mana saja yang sudah diproses maupun yang belum. Selain itu juga, sistem informasi menghadirkan sebuah fitur upload dokumen dan laporan seluruh riwayat transaksi.

Kata Kunci: Pemantauan Tagihan, Vendor, Waterfall

## Abstract

*The problem of monitoring the history of buying and selling transactions between a company and a vendor is not carried out effectively, giving rise to cases of recurring payments that can harm a company. The main problem is that there is no monitoring or recapitulation of all transactions that have been made. Vendor billing information system is needed to solve this problem. The Waterfall model is applied in the development of information systems, starting from analyzing information system requirements, then making a design in the form of a user interface and database design, developing a system using a programming language media that translates the design with a computer, the trial and maintenance process is the last stage carried out. on the method of making a vendor billing monitoring information system. Creating a good business process so that it will indirectly help facilitate the monitoring process. There are still some features that need to be improved. The information system is able to monitor which sales and purchase invoices have been processed or which have not. In addition, the information system provides a feature for uploading documents and reporting all transaction history.*

**Keyword** : Billing Monitoring, Vendor, Waterfall

## 1. Pendahuluan

Pihak kedua atau pihak luar yang menjamin ketersediaan barang atau jasa, kegiatan jual beli perusahaan atau untuk keberlangsungan kegiatan operasional disebut vendor. Biasanya mereka akan

menyediakan berbagai macam barang atau jasa karena vendor tahu bahwa pihaknya merupakan penyedia untuk badan usaha lain. Vendor juga memberikan keuntungan bagi pelanggan, karena mereka biasanya memberikan harga yang jauh lebih murah daripada harga di pasar dan menawarkan

fleksibilitas waktu ketika melakukan transaksi jual beli.

Melakukan transaksi dengan sebuah vendor tentu sudah menjadi hal biasa yang sering dilakukan oleh suatu perusahaan. Bukan hanya vendor yang mampu mensuplai kebutuhan sesuai dengan bidang bisnis perusahaan tersebut tetapi sering kali justru perusahaan membutuhkan vendor untuk dapat memenuhi kebutuhan operasional diluar dari bidang usahanya. Hal tersebut juga berlaku pada PT Wijaya Karya (Persero) Tbk Divisi Bangunan Gedung yang memiliki banyak vendor untuk dapat memenuhi kebutuhan operasionalnya. Tetapi dengan banyaknya vendor dan transaksi yang dilakukan menyebabkan pemantauan terhadap tagihan yang sudah dibayarkan atau belum menjadi kendala utama, (Anggraeni, 2017) seringkali juga terdapat keterlambatan administrasi ketidaktepatan informasi antara vendor dengan perusahaan. Sehingga hal tersebut dapat memunculkan potensi kerugian bagi perusahaan, yaitu dapat terjadinya tagihan berulang atau tagihan yang seyogyanya sudah dibayarkan menjadi muncul kembali dalam keadaan belum dibayarkan.

(Ginantra et al., 2020) Informasi dan teknologi adalah dua hal yang tak dapat untuk dipisahkan, hal ini pula yang dibutuhkan oleh para pelaku usaha beserta dengan konsumennya. Solusi tepat guna yang ditawarkan dalam mengatasi masalah pemantauan tagihan ini yaitu dengan menerapkan sistem informasi yang mampu memantau seluruh tagihan dari vendor tersebut apakah sudah berstatus terlunasi atau belum. Menurut (Mulyani, 2016) Sistem informasi menerima masukan data, instruksi dan mengolah data sesuai dengan perintah untuk mengeluarkan hasilnya, ini merupakan sebagian dari peristiwa yang terjadi pada sistem informasi. Dengan menggunakan sistem informasi ini diharapkan mampu meningkatkan efisiensi dalam proses pemantauan tagihan.

Dalam (Nurhidayah et al., 2020) Teknologi Informasi dan Komunikasi pada saat ini sudah berkembang pesat di seluruh bidang kegiatan. Perkembangan tersebut memicu banyaknya perangkat lunak yang muncul sesuai kebutuhan bidang kegiatan, salah satunya perusahaan. Berdasarkan analisis tersebut, sistem informasi berbasis website pemantauan tagihan vendor untuk menyelesaikan permasalahan yang ada

pada PT Wijaya Karya (Persero) Tbk Divisi Bangunan Gedung.

## 2. Metode Penelitian

Suatu cara untuk memperoleh penemuan yang signifikan secara operasional terhadap suatu sistem yang saat ini berlangsung pada PT Wijaya Karya (Persero) Tbk Divisi Bangunan Gedung.

Adapun beberapa Teknik Pengumpulan Data yang dilakukan antara lain :

Observasi, yaitu pengamatan yang dilakukan secara langsung untuk mengetahui sistem monitoring yang selama ini berjalan pada PT Wijaya Karya (Persero) Tbk Divisi Bangunan Gedung yang beralamat di Jl. DI. Panjaitan No.Kav. 9-10, RT 001/ RW 011, Kelurahan Cipinang Cempedak, Kecamatan Jatinegara, Kota Jakarta Timur, Daerah Khusus Ibukota Jakarta 13340.

Wawancara, metode yang digunakan adalah dengan memberikan beberapa pertanyaan secara langsung kepada responden yang merupakan PIC untuk melakukan transaksi dengan vendor.

Studi Pustaka, sebagai metode untuk mendapatkan benchmarking dan landasan teoritis yang menjadi sumber data guna mendukung penelitian dalam pengembangan sistem usulan.

Sementara itu, beberapa model pengembangan sistem yang dilakukan melalui beberapa tahapan berikut :

Analisa kebutuhan software, sistem informasi yang dapat membuat seorang pengguna login dengan menggunakan username dan password, pengguna melakukan masukan berupa informasi detail mengenai faktur jual beli, melakukan pemantauan terhadap faktur jual beli. Sementara pada saat faktur jual beli ingin dibayarkan, maka sistem informasi mampu memberikan detail informasi mengenai faktur jual beli mana saja yang akan diproses dan memberikan laporan mengenai riwayat transaksi yang sudah dilakukan.

Desain, bahasa pemodelan yang digunakan pada sistem informasi ini adalah Unified Modelling Language (UML) dengan Use Case Diagram. Sementara untuk desain basis data menggunakan Entity Relational Diagram (ERD).

Code Generation, sebagai media penerjemah antara desain dengan sebuah komputer, menggunakan bahasa

pemrograman HTML, CSS dan PHP. MySQL digunakan sebagai pengolahan basis data.

Testing, pada tahap pengujian menggunakan metode blackbox testing. Melakukan pengujian terhadap seluruh sistem informasi yang bekerja, dengan menguji coba seluruh fitur yang ada pada sistem informasi mulai dari memasukan data hingga menampilkan informasi mengenai laporan transaksi dan memastikan bahwa website terhubung dengan baik oleh basis data.

Support, ketika sistem informasi sudah selesai dibangun, maintenance akan tetap berlangsung dengan memperbaiki kesalahan yang belum ditemukan sebelumnya dan senantiasa meningkatkan kualitas layanan pada sistem informasi.

### 3. Hasil dan Pembahasan

#### 3.1. Definisi Sistem Informasi

Kalaulah Sistem informasi itu dapat didefinisikan sebagai sekumpulan sub sistem yang berhubungan satu dengan yang lainnya demi tercapainya suatu tujuan akibat daripada data yang diolah menjadi sebuah informasi berguna (Rahmayu et al., 2019). Pengelompokan beberapa komponen dan bagaimana cara mereka berkelompok merupakan bagian dari sub-sub sistem.

Sistem informasi tidak akan terwujud atau tidak akan terbentuk apabila terdapat salah satu unsur yang tidak tersedia terlepas dari bagaimana pengelompokan itu dilakukan. Komponen-komponen sistem informasi sebagai berikut :

- a. Jaringan komunikasi (*Communication network*)
- b. Manusia (*Brainware*)
- c. Basis data (*Database*)
- d. Prosedur (*Procedure*)
- e. Perangkat lunak
- f. Perangkat keras

Dalam referensi (Anggraeni, 2017, p. 1) sistem dapat dianalogikan sebagai beberapa orang yang sedang bekerja sama demi mencapai suatu tujuan dengan ketentuan terstruktur dan sistematis yang sudah ditetapkan aturannya demi terbentuknya satu kesatuan yang solid. Komponen sistem, batasan sistem, lingkungan luar sistem, penghubung sistem, masukan sistem, keluaran sistem, pengolahan sistem dan sasaran sistem merupakan karakteristik yang harus dimiliki oleh sistem. Sedangkan informasi adalah

data yang diolah untuk mengurangi ketidakpastian dalam proses pengambilan keputusan terhadap suatu hal agar lebih berguna dan berarti bagi penerimanya.

Fungsi sistem informasi adalah sebagai berikut :

- a. Meningkatkan aksesibilitas kepada pengguna berupa penggunaan data.
- b. Memanfaatkan sistem informasi secara kritis untuk menjadi tersedianya kualitas dan keterampilan.
- c. Sistem informasi membutuhkan keterampilan untuk dapat diidentifikasi.
- d. Konsekuensi ekonomi yang harus dipahami dan diantisipasi.
- e. Sistem informasi harus menjadi investasi yang ditetapkan di waktu yang akan datang.
- f. Efektif dalam proses perencanaan.

#### 3.2. Perancangan Basis Data

##### a. Konsep Dasar Basis Data

Menurut sumber (Ginatra et al., 2020, p. 1) Pemanfaatan dan perkembangan perangkat lunak di berbagai bidang harus mampu menyeimbangkan perkembangan teknologi 4.0 yang dengan pesat berkembang. Salah satu faktor penentu kualitas perangkat lunak adalah tentang bagaimana pemanfaatan perangkat lunak dalam bidang pengolahan data. Pengolahan data yang baik akan memberikan informasi yang baik dalam mendukung keputusan bagi pihak terkait. Apa itu data dan informasi? Kumpulan fakta, angka dan kata yang belum memiliki kegunaan merupakan definisi dari data, sementara informasi adalah kumpulan data yang akan digunakan sebagai bahan pengambilan keputusan yang telah melalui proses pengolahan.

Istilah basis data (*database*) pada dasarnya berkaitan dengan perangkat lunak dalam hal pengolahan data secara digital. Kebutuhan akan aplikasi pengolahan basis data merupakan hal yang tidak dapat dihilangkan dalam perkembangan baru ini dari sektor kepentingan bisnis dan lainnya. Secara umum basis data dapat di definisikan sebagai item data yang saling berhubungan satu sama lain yang diorganisasikan secara struktur dan dengan memanipulasi data menggunakan perangkat lunak untuk kegunaan tertentu. Basis data (*database*) juga dapat diartikan kumpulan *records* yang saling berhubungan yang disimpan secara bersama sedemikian rupa tanpa

pengulangan (*redundancy*) yang tidak perlu untuk memenuhi kebutuhan. Pengertian ini digambarkan seperti lemari arsip yang menyimpan beberapa arsip di sebuah ruangan yang dikelola secara konvensional sedangkan basis data penyimpanan data secara digital yang diproses dengan menggunakan *Database Management System* (DBMS).

Beberapa istilah dalam sistem basis data adalah sebagai berikut :

1) *Enterprise*

Yang merupakan suatu bentuk organisasi. Beberapa contoh dari *enterprise* adalah Sekolah, Rumah Sakit.

2) Entitas

Objek yang mampu membedakan satu dengan lainnya.

Contoh :

*Database* Nilai → entitas : Mahasiswa, Mata Pelajaran.

3) *Field/Atribute*

Setiap entitas mempunyai suatu sebutan untuk mewakili suatu entitas.

Contoh :

Entitas siswa → field = Nim, nama, alamat dan lain-lain.

Entitas nasabah → field = Kode\_nasabah, nama\_nasabah dan lain-lain.

4) *Data value*

Isi daripada Field merupakan *data value*.

Contoh :

Atribut nama\_karyawan → Sutrisno, Budiman dan lain-lain.

5) *Record/tuple*

Suatu entitas yang dapat diinformasikan secara lengkap melalui beberapa kumpulan elemen-elemen.

6) *File*

*Data value* yang berbeda pada tiap *field*.

7) Kunci Elemen

Mengidentifikasi suatu entitas melalui nomor yang bersifat unik.

b. Sistem Manajemen Basis Data (DBMS)

Melakukan penyimpanan dan pengaturan basis data yang didesain dalam bentuk perangkat lunak merupakan sistem manajemen basis data. Pengamanan data, pemakaian data secara bersama dan pemaksaan keakuratan data adalah mekanisme yang diterapkan oleh sistem manajemen basis data (Pamungkas, 2017, p. 4).

Menurut (Widodo & Kurnianingtyas, 2017, p. 3) DBMS juga dapat diartikan

sebagai data yang saling berhubungan satu sama lain yang kemudian dikelompokkan ke dalam beberapa tabel dan mengatur cara mengakses data tersebut melalui sebuah program aplikasi. Disebut sebagai basis data karena merupakan kumpulan dari data yang diolah menjadi sebuah informasi nyata yang akan digunakan oleh sebuah perusahaan sebagai bahan untuk mengambil suatu kebijakan. Menyediakan sebuah cara untuk mengambil dan menyimpan informasi basis data secara efisien dan nyaman adalah tujuan utama dari DBMS.

Hal-hal yang meliputi manajemen data diantaranya mekanisme untuk memanipulasi informasi yang ada dan struktur informasi penyimpanan dalam basis data. Salah satu yang menjadi penting di dalam manajemen basis data data mengenai keamanan data yang harus terjamin, walaupun pengaksesan tidak diizinkan atau sistem dalam keadaan rusak.

c. Model Data Relasional

Berdasar (Yanto, 2012, p. 26) pendekatan yang dilakukan oleh model data relasional dengan menawarkan keunggulan-keunggulan dibandingkan dengan sistem pemrosesan berkas tradisional. Keunggulan-keunggulan diantaranya adalah sebagai berikut:

- 1) Kemandirian akan data dan program. Pemisahan antara data dan program tersebut dinamakan kemandirian data (*data independence*). Dengan lokasi terpusat yang dinamakan repositori sebagai wadah untuk menyimpan data. Salah satu karakteristik unggulan dari basis data ini adalah mampu mengubah organisasi data (sampai batas-batas tertentu) tanpa berpengaruh terhadap program aplikasi.
- 2) Meminimalisir pengulangan data (*Redundancy*). Menyatukan berkas-berkas data pada suatu struktur logika yang tunggal merupakan sasaran perancangan dengan pendekatan basis data. Setiap fakta primer (*Primary Key*) hanya direkam pada satu tempat di basis data. Sebagai contoh Nomor Induk Mahasiswa (NIM) mahasiswa hanya tersimpan pada table atau entitas mahasiswa. Hal tersebut tidak serta merta mampu menghilangkan *redundancy* data sama sekali, tetapi mengizinkan perancang untuk lebih berhati-hati mengendalikan *redundancy*.

Model data relasional menggunakan beberapa tabel untuk menghubungkan sejumlah data sebagai gambaran. Setiap tabel memiliki field/atribut yang masing-masing terdapat kode unik (*primary key*).

Tabel 1. Tabel Mahasiswa

NIM	Nama	Umur
2008010001	Dwi Susanto	18
2009010002	Endang Etriyanti	20
2011010003	Muhammad Mehmed	19
2011020012	Yoga Pratama	19

Sumber: (Yanto, 2012)

Tabel 2. Tabel Mata Kuliah

Kd_Matkul	Nm_Matkul	SKS
MKK0011	Basis Data	4
MKB0212	Algoritma Lanjutan	4
MPB1201	Matematika Diskrit	2
MKK0013	Struktur Data	4

Sumber: (Yanto, 2012)

Tabel 3. Tabel Nilai

NIM	Kd_Matkul	Nilai
2008010001	MKK0011	A
2009010002	MKB0212	A
2011010003	MPB1201	B
2011020012	MKK0013	A

Sumber: (Yanto, 2012)

Dalam (Yanto, 2012, p. 28) pada tabel 1 memperlihatkan data-data mahasiswa, sebagai contoh mahasiswa dengan NIM 2008010001 bernama Dwi Susanto, mahasiswa dengan NIM 2011010003 bernama Muhammad Mehmed dan seterusnya. Selanjutnya pada tabel 2 memperlihatkan data mengenai mata kuliah, semisal mata kuliah Algoritma Lanjutan memiliki Kd\_matkul MKB0212 dan seterusnya. Kemudian pada tabel 3 memperlihatkan data-data mahasiswa yang memperlihatkan data-data mahasiswa yang mengambil mata kuliah tertentu beserta dengan hasil nilai yang di peroleh tiap mahasiswa, contoh NIM 2011010003

bernama Muhammad Mehmed mengambil mata kuliah Matematika Diskrit dengan nilai B dan seterusnya.

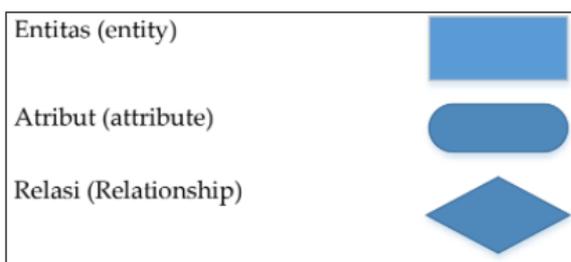
Contoh model data berbasis rekaman ini merupakan model data relasional. Mengapa dinamakan tersebut karena basis data memiliki struktur rekaman berformat tertentu yang masing-masing memiliki tipe yang berbeda. Dalam hal ini setiap kolom pada tabel mencerminkan field/atribut setiap entitas yang bersangkutan yang juga akan kita jumpai pada model konseptual ERD.

#### d. *Entity Relationship Diagram* (ERD)

Robi (Yanto, 2012, p. 31) berpendapat pemodelan data konseptual yang paling sering digunakan dalam proses pengembangan basis data bertipe relasional salah satunya adalah ERD. Kemudahan penggunaan secara luas atau biasa disebut *Computer Aided Software Engineering* (CASE), dukungan konsep matematika yang tangguh, hubungan antar entitas merupakan konsep pemodelan alamiah yang sesuai dengan keadaan dunia nyata, faktor tersebutlah yang menjadikan penggunaan ERD sangat luas.

Mengkonstruksi model data agar terlihat konseptual, mencerminkan struktur data dan batasan-batasan dari basis data, dari perangkat lunak pengelolaan basis data (DBMS) dan erat hubungannya dengan model data yang langsung dapat digunakan untuk mengimplementasikan basis data secara logika maupun secara fisik dengan menggunakan DBMS yang dipilih pada tahap implementasi biasa digunakan pada permodelan E-R.

Pada (Yanto, 2012, p. 32) secara sederhana ERD dapat di analogikan sebagai suatu diagram yang menggambarkan desain konseptual dari suatu basis data relasional. Gambaran antara objek yang satu dengan objek lain dari objek realitas yang dikenal dengan hubungan antar entitas. Semisal apabila kita ingin membuat ERD dari Sistem Perpustakaan maka bahan sebagai objek ERD dapat berupa buku, anggota, peminjaman, pengembalian dan sebagainya. Terdapat 3 (tiga) komponen pada ERD, dapat dilihat pada Gambar 1 berikut :



Gambar 1. Komponen *Entity Relationship Diagram*

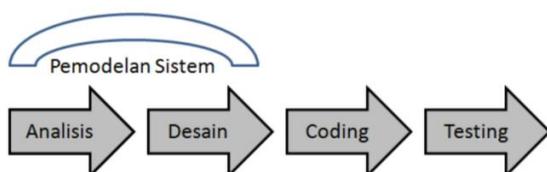
Sumber: (Yanto, 2012)

### 3.3. *Waterfall Model*

Indra Rianto (Rianto, 2021, pp. 44–47) mengemukakan bahwa model air terjun adalah sebutan yang sering dikenal dengan model *waterfall*, model proses pertama yang diperkenalkan. Mudah digunakan dan dimengerti merupakan sifat yang melekat pada model ini. Pada permodelan *waterfall* ini, setiap fase harus diselesaikan terlebih dahulu sebelum memulai fase selanjutnya sehingga menghindari terjadinya fase yang saling tumpang tindih.

Pada *waterfall* ini, keseluruhan proses pengembangan perangkat lunak dibagi menjadi beberapa fase terpisah. Inputan fase merupakan hasil dari satu fase secara berurutan. Ini artinya bahwa setiap fase selanjutnya akan dimulai jika fase sebelumnya sudah selesai. Salah satu yang mendasari mengapa permodelan ini disebut *waterfall* atau air terjun karena proses desain dijalankan secara berurutan dimana *progress* kegiatan dilihat sebagai bentuk aliran dari atas ke bawah melalui beberapa fase.

Sekuensial linier untuk pengembangan rekayasa perangkat lunak, sering juga dilekatkan dengan model *waterfall*. Berikut gambar yang menggambarkan tentang sekuensial linier :



Gambar 2. Model Sekuensial Linier (*Waterfall Model*)

Sumber: (Rianto, 2021)

Pendekatan yang dilakukan pada sekuensial linier ini dengan pengembangan perangkat lunak yang sistemik, yang dimulai pada tingkat dan kemajuan sistem pada seluruh fase analisis, desain, kode, pengujian serta pemeliharaan. Berikut ini adalah penjelasan lebih rinci pada masing-masing fase :

#### 1) Analisis Kebutuhan Perangkat Lunak

Langkah awal dalam proyek pembuatan perangkat lunak dengan menganalisis kebutuhan yaitu dengan menentukan gambaran perangkat lunak yang akan dihasilkan ketika proses pengembangan perangkat lunak dilakukan. Keberhasilan dalam melakukan analisis kebutuhan akan sangat bergantung pada perangkat lunak yang baik dan sesuai dengan kebutuhan pengguna. Proyek-proyek perangkat lunak dengan ruang lingkup yang besar, setelah tahap sistem *information engineering* dan *software project planning* diselesaikan baru tahap analisis kebutuhan.

Proses analisis kebutuhan yang baik belum tentu akan mampu menghasilkan perangkat lunak yang baik pula, akan tetapi analisis kebutuhan yang tidak tepat akan menghasilkan perangkat lunak yang tidak berguna. Mengetahui adanya kesalahan analisis kebutuhan pada tahap awal memang jauh lebih baik, ketimbang mengetahui kesalahan ketika sudah sampai pada tahap penulisan kode atau bahkan tahap pengujian, merupakan kesalahan besar yang harus diantisipasi oleh pengembang perangkat lunak. Dengan adanya hal tersebut, tentu akan mengakibatkan kerugian dari sisi waktu dan biaya.

Sebagai langkah untuk memitigasi kerugian atas kesalahan itu, seorang analis (perekayasa perangkat lunak) harus memahami domain permasalahan, tingkah laku, unjuk kerja dan *interface* yang diperlukan. Kebutuhan yang berkaitan dengan sistem akan didokumentasikan dan dilihat kembali oleh klien.

#### 2) Desain

Proses desain sebenarnya berfokus pada empat atribut program yang berbeda mulai dari struktur data, arsitektur perangkat lunak, representasi *interface* dan algoritma yang procedural. Menerjemahkan kebutuhan ke dalam sebuah representasi perangkat lunak yang diperkirakan demi menunjang

kualitas sebelum dimulainya penulisan kode (*coding*). Proses desain ini juga harus didokumentasikan, sebagaimana pada proses analisis.

- 3) Menghasilkan Kode  
Ketika tahap desain sudah selesai dikerjakan, maka langkah selanjutnya adalah dengan menerjemahkan desain tersebut ke dalam bentuk bahasa mesin yang dapat dibaca oleh perangkat keras. Penulisan kode ini dapat dilakukan secara mekanis.
- 4) Pengujian (*Testing*).  
Pengujian program dapat dimulai ketika sekali kode dibuat. Fokus dari pengujian ada pada logika internal perangkat lunak, memastikan bahwa semua pernyataan sudah diuji dan pada eksternal fungsional, yaitu pengujian untuk menemukan kesalahan-kesalahan dan memastikan bahwa input yang dibatasi akan mampu memberikan hasil sesuai dengan yang dibutuhkan.
- 5) Pemeliharaan  
Tentu seiring berjalannya waktu, perubahan dan penyesuaian akan dialami perangkat lunak setelah disampaikan kepada pelanggan. Perubahan dapat terjadi karena kesalahan yang ada pada perangkat lunak atau pelanggan membutuhkan pengembangan dari aspek fungsional yang harus diakomodir. Pemeliharaan perangkat lunak diaplikasikan pada tiap fase sebelumnya dan tidak membuat kembali dari awal.

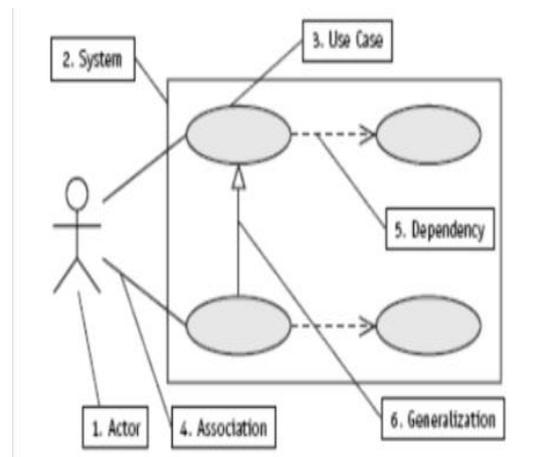
### 3.4. Unified Modeling Language

(Mulyani, 2016, pp. 42–43) UML atau *Unified Modeling Language* adalah teknik pengembangan sistem untuk melakukan spesifikasi dan pendokumentasian pada sistem dengan menggunakan bahasa grafis sebagai alatnya. Adalah Booch dan James Rumbaugh di tahun 1994 yang menggabungkan dua metodologi terkenal yaitu Booch dan OMT, kemudian untuk standar UML sendiri dikelola oleh *Object Management Group* (OMG).

Untuk melakukan pemodelan data dan sistem, UML memiliki banyak sekali variasi diagram. Salah satunya adalah *Use Case Diagram*.

*Use Case Diagram* digunakan untuk menggambarkan hubungan antara sistem dengan aktor. Elemen-elemen yang terdapat dalam *Use Case Diagram* dapat dikatakan jumlahnya tidak banyak, hal itu

disebabkan karena *Use Case Diagram* hanya menggambarkan sistem secara global. Pada Gambar 3 ini adalah elemen-elemen yang terdapat pada *Use Case Diagram*.



Gambar 3. Elemen *Use Case Diagram*

Sumber: (Mulyani, 2016)

**Sistem**, yaitu batasan pada proses yang telah kita tentukan dalam sebuah sistem. Secara sederhana untuk memperlihatkan batasan dalam dan luar sistem.

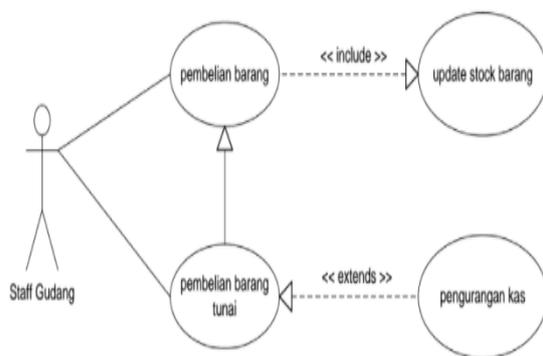
**Aktor**, dapat berupa orang, mesin ataupun sistem lain yang berinteraksi langsung dengan *use case*. Aktor juga dapat dikatakan sebagai elemen pemicu sistem.

**Use Case**, sebuah proses yang merupakan bagian dari sistem.

**Association**, sebagai jembatan penghubung yang menggambarkan interaksi antara *use case* dan aktor.

**Dependency**, terdapat 2 (dua) tipe yang menggambarkan relasi pada *dependency* yaitu, *include* dan *extends*. *Include* merupakan tipe yang menghubungkan 2 (dua) *use case* dimana, salah satu dari *use case* membutuhkan *use case* yang lain. Sementara pada *extends*, tipe yang menghubungkan 2 (dua) *use case* akan tetapi pada kondisi tertentu *use case* yang satu akan memanggil *use case* yang lain.

**Generalization**, menggambarkan pewarisan antara dua aktor atau *use case* dimana salah satu aktor atau *use case* mewarisi *properties* ke aktor atau *use case* yang satunya. Untuk lebih jelas, perhatikan Gambar 4 berikut.



Gambar 4. Contoh Use Case Diagram

Sumber: (Mulyani, 2016)

### 3.5. HTML

Dalam (Purnama & Watrianthos, 2018, pp. 1–2) *Hypertext Mark Up Language* adalah sebuah bahasa pemrograman untuk membuat website yang dapat diakses melalui jaringan internet. Sederhananya, bahasa ini digunakan untuk diterjemahkan oleh komputer agar dapat dipahami oleh penggunaannya dan kemudian itu semua disusun sehingga menjadi tampilan website yang mampu kita lihat dan baca.

HTML disusun dengan kode dan simbol tertentu yang dimasukkan ke dalam suatu file atau dokumen. Pada saat kita membuka browser kemudian kita mencari suatu website yang ini kita tuju maka website tersebut dibuat dengan menggunakan HTML.

Ada beberapa makna dibalik dari singkatan HTML yang berarti *Hypertext Mark Up Language*. Suatu metode yang digunakan untuk berpindah dari satu web ke web yang lain dengan mengklik suatu teks atau simbol adalah definisi dari kata *Hypertext* itu sendiri. Sementara istilah *Mark Up* dapat diartikan sebagai suatu hal yang dilakukan tag HTML terhadap sebuah tulisan yang berada di dalamnya, semisal jika kita mengetik suatu kalimat di dalam tag `<b>` maka kalimat tersebut akan tampil dengan huruf tebal pada tampilan website. Terakhir adalah kata *Language* dapat di definisikan sebagai bahasa pemrograman yang disusun dari tag-tag tertentu yang nantinya akan diterjemahkan ke dalam teks atau visual yang tampil pada halaman website.

### 3.6. CSS

Menurut (Purnama & Watrianthos, 2018, p. 17) apabila HTML adalah bahasa pemrograman untuk membuat sebuah

website, maka CSS dapat diartikan sebagai kode program yang bertujuan untuk mempercantik tampilan pada website. Mengubah desain dari teks, warna, gambar hingga latar belakang dapat kita lakukan dengan CSS.

### 3.7. PHP

Pada (Nurhidayah et al., 2020, p. 16) menyatakan bahwa *Hypertext Preprocessor* atau PHP adalah bahasa yang berbentuk skrip dan memiliki sifat *server side* yang mana kode program akan dilakukan pada *server* dan hasilnya akan tampil di *browser*.

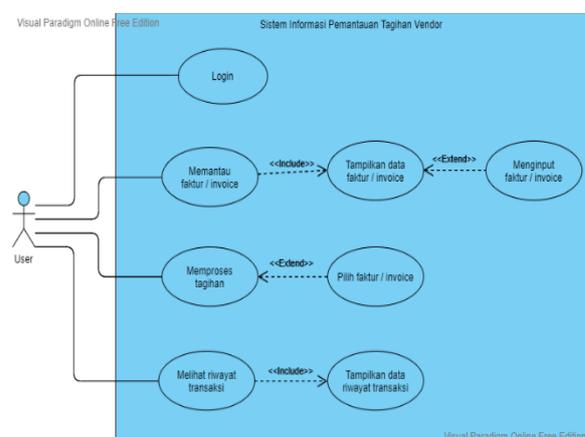
### 3.8. Analisa Sistem Berjalan

PT Wijaya Karya (Persero) Tbk Divisi Bangunan Gedung sering kali menggunakan sebuah vendor untuk memenuhi kebutuhan operasionalnya. Kebutuhan akan vendor yang bergerak dibidang jasa travel, alat-alat yang berkaitan dengan komputer dan alat tulis kantor merupakan contohnya. Dalam proses transaksi, diawali dengan melakukan pemesanan kepada vendor, kemudian vendor akan memenuhi kebutuhan sesuai dengan yang dipesankan oleh PIC (*Person In Charge*) dari PT Wijaya Karya (Persero) Tbk Divisi Bangunan Gedung dan akan menerbitkan sebuah faktur jual beli atau *invoice* apabila barang atau jasa sudah diterima. Transaksi akan dianggap selesai apabila pihak divisi melalui tim keuangannya sudah melakukan pembayaran kepada vendor.

### 3.9. Rancangan Sistem

#### a. Analisa Kebutuhan Sistem

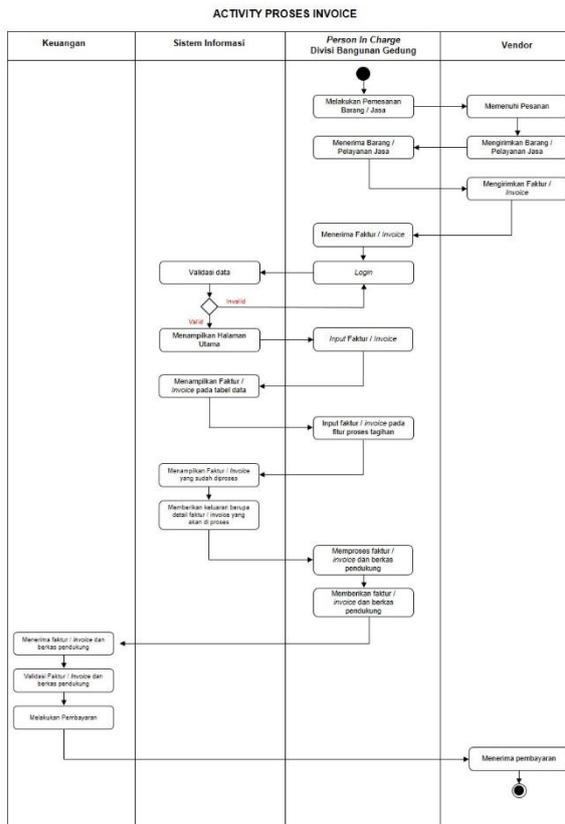
##### 1) Use Case Diagram



Gambar 5. Use Case Diagram

Sumber: Penelitian (2021)

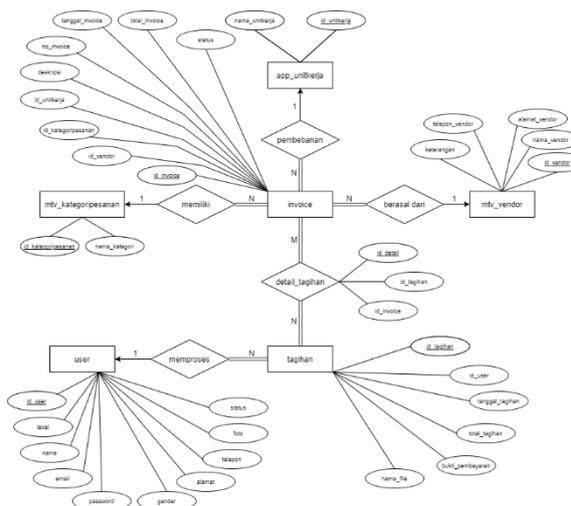
2) Activity Diagram



Gambar 6. Activity Diagram

Sumber: Penelitian (2021)

b. Desain  
3) Entity Relationship Diagram



Gambar 7. Entity Relationship Diagram

Sumber: Penelitian (2021)

4) Rancangan Basis Data

Tabel 4. Tabel Kategori Pesanan

No	Elemen Data	Nama Field	Type	Size	Ket
1	Id Kategori Pesanan	id_kategoripesanan	Int	2	Primary Key
2	Nama Kategori	nama_kategori	Varchar	30	

Sumber: Penelitian (2021)

Tabel 5. Tabel Unit Kerja

No	Elemen Data	Nama Field	Type	Size	Ket
1	Id Unit Kerja	id_unitkerja	Int	4	Primary Key
2	Nama Unit Kerja	Nama_unitkerja	Varchar	50	

Sumber: Penelitian (2021)

Tabel 6. Tabel Vendor

No	Elemen Data	Nama Field	Type	Size	Ket
1	Id Vendor	id_vendor	Int	2	Primary Key
2	Nama Vendor	nama_vendor	Varchar	30	
3	Alamat Vendor	alamat_vendor	Text		
4	Telepon Vendor	telepon_vendor	Varchar	15	
5	Keterangan	keterangan	Text		

Sumber: Penelitian (2021)

Tabel 7. Tabel Invoice

No	Elemen Data	Nama Field	Type	Size	Ket
1	Id Invoice	id_invoice	Int	4	Primary Key
2	Id Vendor	id_vendor	Int	2	Foreign Key
3	Id Kategori Pesanan	id_kategoripesanan	Int	2	Foreign Key
4	Id Unit Kerja	id_unitkerja	Int	4	Foreign Key
5	Deskripsi	deskripsi	Text		
6	Nomor Invoice	no_invoice	Varchar	30	
7	Tanggal Invoice	tanggal_invoice	Date		
8	Total Invoice	total_invoice	Float		
9	Status	status	Tinyint	1	

Sumber: Penelitian (2021)

Tabel 8. Tabel Detail Tagihan

No	Elemen Data	Nama Field	Type	Size	Ket
1	Id Detail	id_detail	Int	4	Primary Key
2	Id Tagihan	id_tagihan	Varchar	10	Foreign Key
3	Id Invoice	id_invoice	Int	4	Foreign Key

Sumber: Penelitian (2021)

Tabel 9. Tagihan

No	Elemen Data	Nama Field	Type	Size	Ket
1	Id Tagihan	id_tagihan	Varchar	10	Primary Key
2	Id User	id_user	Int	4	Foreign Key
3	Tanggal Tagihan	tanggal_tagihan	Date		
4	Total Tagihan	total_tagihan	Float		
5	Bukti Pembayaran	bukti_pembayaran	Varchar	50	
6	Nama File	nama_file	Varchar	50	

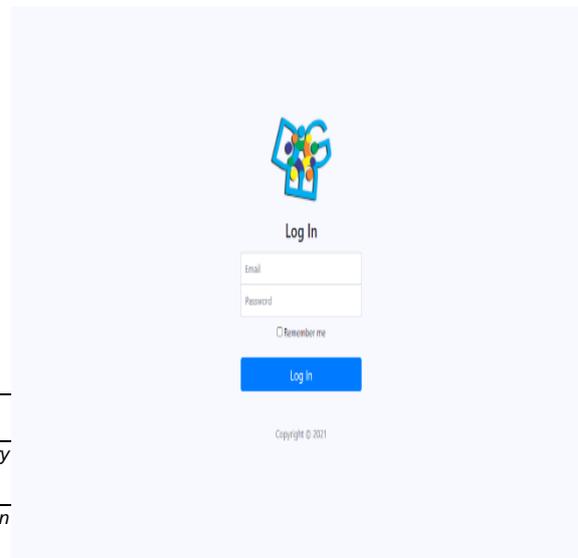
Sumber: Penelitian (2021)

Tabel 10. Tabel User

No	Elemen Data	Nama Field	Type	Size	Ket
1	Id User	id_user	Int	4	Primary Key
2	Level User	level	Tinyint	1	
3	Nama	nama	Varchar	30	
4	Email	email	Varchar	40	
5	Password	password	Varchar	35	
6	Jenis Kelamin	gender	Tinyint	1	
7	Alamat	alamat	Text		
8	Nomor Telepon	telepon	Varchar	15	
9	Foto	foto	Varchar	50	
10	Status Keaktifan	status	Tinyint	1	

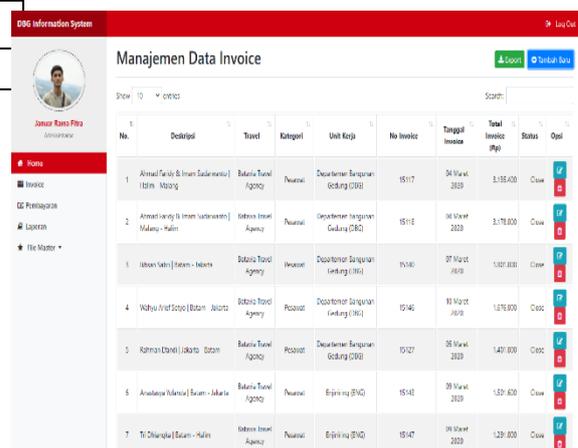
Sumber: Penelitian (2021)

## 5) User Interface



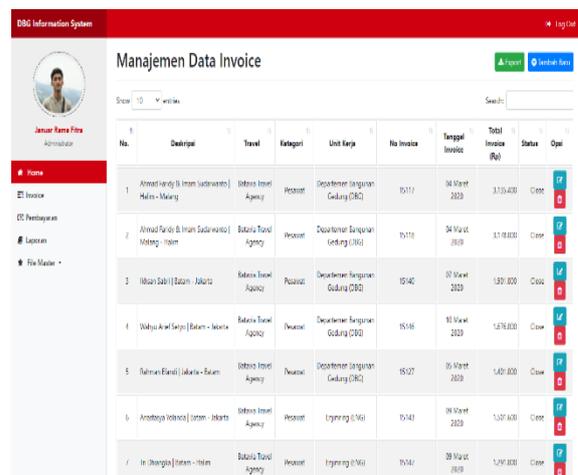
Gambar 8. Tampilan Login

Sumber: Penelitian (2021)



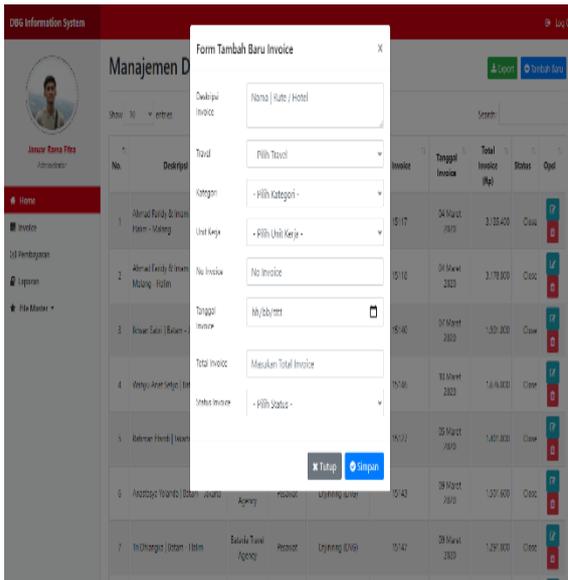
Gambar 9. Tampilan Menu Home

Sumber: Penelitian (2021)

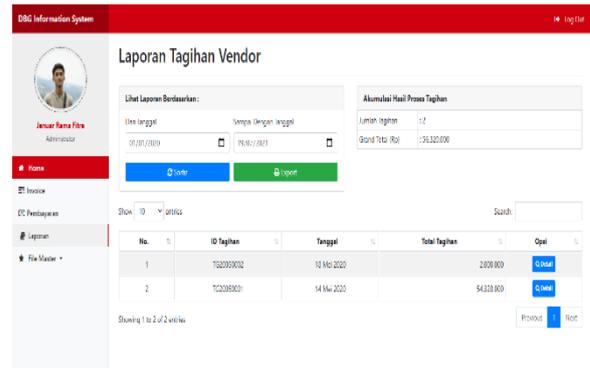


Gambar 10. Tampilan Menu Invoice

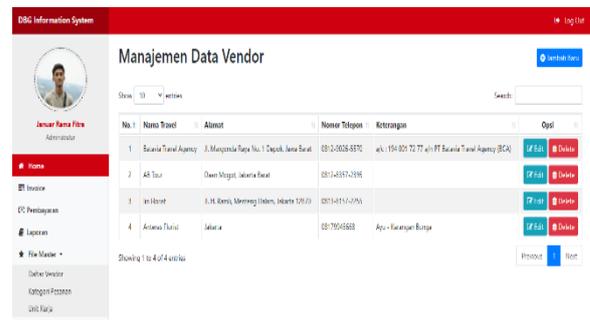
Sumber: Penelitian (2021)



Gambar 11. Tampilan Formulir Tambah Baru Invoice  
Sumber: Penelitian (2021)



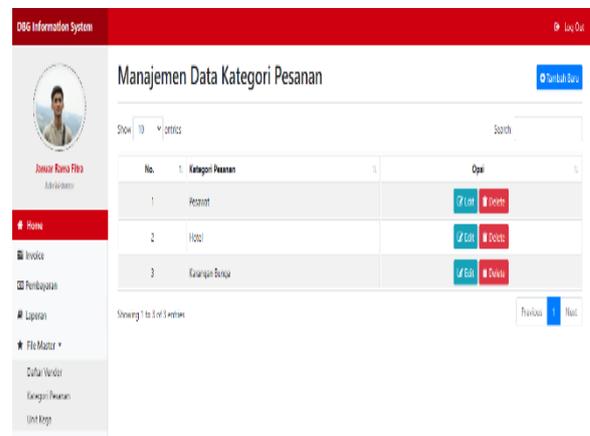
Gambar 14. Tampilan Menu Laporan  
Sumber: Penelitian (2021)



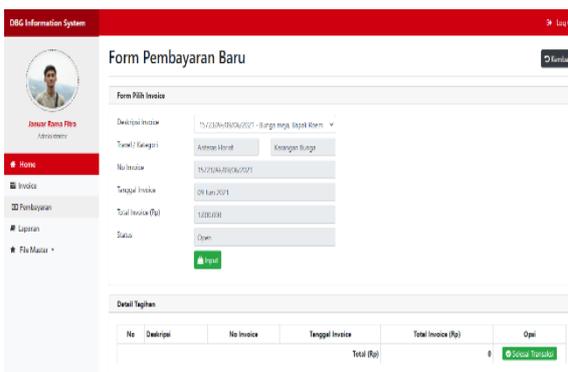
Gambar 15. Tampilan Menu Daftar Vendor  
Sumber: Penelitian (2021)



Gambar 12. Tampilan Menu Pembayaran  
Sumber: Penelitian (2021)



Gambar 16. Tampilan Menu Kategori Pesanan  
Sumber: Penelitian (2021)



Gambar 13. Tampilan Formulir Pembayaran Baru  
Sumber: Penelitian (2021)

No.	Nama Unit Kerja	Opil
1	Human Capital (HC)	[Edit] [Hapus]
2	Keuangan (FI)	[Edit] [Hapus]
3	Keperawatan (KPR)	[Edit] [Hapus]
4	Keperawatan (KPR)	[Edit] [Hapus]
5	Quality Safety Health & Environment (QSHE)	[Edit] [Hapus]
6	Manajemen Risiko (BAM/MS)	[Edit] [Hapus]
7	Keamanan (PS)	[Edit] [Hapus]
8	Keamanan (PS)	[Edit] [Hapus]
9	Keamanan & Pengendalian Populasi (KPP)	[Edit] [Hapus]
10	Keamanan & Pengendalian Lingkungan (KPL)	[Edit] [Hapus]

Gambar 17. Tampilan Menu Unit Kerja  
Sumber: Penelitian (2021)

#### 4. Kesimpulan

Dari hasil penelitian yang dilakukan, beberapa kesimpulan dihasilkan, yaitu sebagai berikut :

- Sistem informasi Pemantauan Tagihan Vendor ini, nantinya pengguna akan lebih mudah melakukan monitoring terhadap seluruh invoice yang dihasilkan dari transaksi antara perusahaan dengan vendor.
- Sistem informasi yang dibangun dengan menggunakan bahasa pemrograman PHP dan MySQL untuk pengelolaan basis data, memungkinkan seluruh data yang akan dimasukkan ke dalam sistem informasi ini akan menjadi terpusat di dalam satu wadah yang disebut dengan *Database Management System*.
- Model waterfall yang diterapkan memudahkan dalam pengembangan sistem informasi, setiap fase yang dijalankan sesuai dengan kebutuhan dan pengembangannya.
- Sistem informasi yang dihasilkan dapat memantau dan tersedianya rekapitulasi terhadap semua transaksi, sehingga memudahkan perusahaan dalam memantau aktivitas transaksi dari vendor.

#### Referensi

- Anggraeni, E. Y. (2017). *Pengantar Sistem Informasi*. CV. Andi Offset.
- Ginatra, N. L. W. S. R., Wardani, N. W., Aristamy, I. G. A. A. M., Suryawan, I. W. D., Ardiana, D. P. Y., Sudipa, I. G. I., Dirgayusari, A. M., Mahendra, G. S., Ariasih, N. K., & Parwita, W. G. S. (2020). *Basis Data Teori dan*

- Perancangan*. Yayasan Kita Menulis.
- Mulyani, S. (2016). *Analisis dan Perancangan Sistem Informasi Manajemen Keuangan Daerah: Notasi Pemodelan Unified Modeling Language (UML)*. Abdi Sistematika.
- Nurhidayah, S., Fauzan, M. N., & Rahayu, W. I. (2020). *Implementasi Metode Analytic Hierarchy Process (AHP) dengan PHP*. Kreatif Industri Nusantara.
- Pamungkas, C. A. (2017). *Pengantar dan Implementasi Basis Data*. Deepublish.
- Purnama, I., & Watrionthos, R. (2018). *Sistem Informasi Kursus PHP dan MySQL*. Uwais Inspirasi Indonesia.
- Rahmayu, M., Serli, R. K., & Hanawi, S. A. (2019). Sistem Pelayanan E-Invoicing Bulanan Pada Apartemen Permata Surya 1 Jakarta. *JUSIM (Jurnal Sistem Informasi Musirawas)*, 4(1), 18–26. <https://doi.org/10.32767/jusim.v4i1.407>
- Rianto, I. (2021). *Rekayasa Perangkat Lunak*. Lakeisha.
- Widodo, A. W., & Kurnianingtyas, D. (2017). *Sistem Basis Data*. UB Press.
- Yanto, R. (2012). *Manajemen Basis Data menggunakan MySQL*. Deepublish.