

BAB II TINJAUAN PUSTAKA

2.1. Teori Dasar

Sebelum penulis menguraikan konten laporan ini, diperlukan adanya dasar teori yang kokoh agar penulis dapat memiliki pemahaman menyeluruh tentang isi laporan. Oleh karena itu, penulis akan menyajikan atau menjelaskan landasan teori berdasarkan pandangan para ahli sesuai dengan judul penyusunan laporan ini.

2.1.1. Sistem

Menurut Kurnia Cahya Lestari dan Arni Muarifah Amri (2020), sistem berdasarkan dari dua atau lebih elemen yang saling terkoneksi dan berinteraksi satu sama lain, membentuk suatu kesatuan yang berguna untuk kelompok agar mencapai tujuan tertentu. Dari perspektif Universitas Dian Nuswantoro, karakteristik sistem dapat dijelaskan sebagai berikut:

1. Komponen Sistem (Component):

Sebuah sistem terdiri dari beragam elemen yang berkaitan dan bekerja sama untuk membentuk sebuah keseluruhan.

2. Subsistem (Subsystem):

Elemen-elemen sistem dapat membentuk subsistem yang memiliki atribut sistem, melaksanakan fungsi khusus, dan berdampak pada proses sistem secara keseluruhan. Sistem juga dapat merupakan bagian dari sistem yang lebih besar yang dikenal sebagai supra sistem.

3. Batasan Sistem (Boundary):

boundary yaitu area yang memisahkan suatu sistem dari sistem lainnya atau sistem dari lingkungan luar. Pembatasan sistem ini menciptakan pandangan bahwa sistem adalah suatu entitas yang tidak dapat dipisahkan.

4. Lingkungan di Luar Sistem (Environment):

Semua entitas yang berada di luar cakupan atau batas sistem yang mempengaruhi kinerja sistem disebut sebagai lingkungan luar sistem.

5. Penghubung Sistem (Interface):

Penghubung sistem memiliki peran menjadi mediator yang menghubungkan satu subsistem dengan subsistem lainnya. Fungsi *Interface* ini memudahkan pemindahan sumber daya dari satu subsistem ke subsistem lainnya. Selain itu, *Interface* menghubungkan keluaran dari satu subsistem dengan masukan dari subsistem lain.

6. Masukan Sistem (Input):

Input sistem bisa berupa input perawatan (*maintenance input*) atau input sinyal (*signal input*). Program, yang dipakai pada sistem komputer untuk menjalankan operasinya adalah input perawatan, dan data yaitu input sinyal yang diubah menjadi informasi.

7. Keluaran Sistem (Output):

Keluaran sistem yaitu hasil dari pemrosesan input yang telah diorganisir. Output ini kemudian menjadi masukan untuk subsistem selanjutnya.

8. Sasaran Sistem (Object):

Sistem mempunyai fungsi yang jelas dan bersifat deterministik. Suatu sistem dibidang berhasil bila dapat menggapai sasaran atau maksud yang sudah di *planning*.

9. Pengolah Sistem (Process):

Sistem bisa mempunyai tahap yang mengubah input menjadi output. Proses ini merupakan elemen kunci yang memungkinkan sistem mencapai tujuannya.

2.1.2. Informasi

Menurut Tukino (2020), informasi yaitu hasil ubahan data yang diubah menjadi sebuah entitas yang mempunyai nilai lebih tinggi bagi penerima dengan fungsi membantu pada proses pengambilan keputusan. Informasi memiliki sifat yang jelas, yaitu keakuratan, ketepatan waktu, dan relevansi, yang menjadikannya memiliki nilai penuh. Selain untuk memberikan pengetahuan informasi juga mempunyai beberapa fungsi yaitu :

1. Dukungan Keputusan:

Informasi yang tepat dan akurat memungkinkan pengguna untuk menilai berbagai opsi, memilih solusi terbaik, dan merencanakan langkah-langkah ke depan.

2. Komunikasi:

Dengan menyajikan data yang telah diolah, pesan atau ide dapat disampaikan secara jelas dan dapat dipahami oleh pihak lain.

3. Pengendalian:

Dengan pemahaman yang komprehensif tentang situasi, manajemen dapat mengambil langkah-langkah pengendalian yang diperlukan.

4. Perencanaan:

Dengan memahami kondisi dan tren, organisasi dapat merencanakan langkah-langkah ke depan untuk mencapai tujuan mereka.

5. Inovasi dan Pengembangan:

Melalui analisis data dan tren, organisasi dapat mengidentifikasi peluang baru, menyesuaikan strategi, dan mengembangkan solusi kreatif.

6. Peningkatan Efisiensi dan Produktivitas:

Dengan akses ke informasi yang diperlukan, pekerja dapat bekerja lebih efisien dan membuat keputusan dengan cepat.

7. Pelacakan dan Evaluasi:

Dengan membandingkan data aktual dengan target atau standar tertentu, organisasi dapat mengevaluasi keberhasilan dan membuat perbaikan jika diperlukan.

2.1.3. Sistem Informasi

Sistem informasi bertujuan untuk mengelola data dan informasi agar dapat digunakan dengan efisien dan efektif dalam konteks bisnis atau institusi. Menurut (Kenneth C. Laudon dan Jane P. Laudon) Sistem informasi dapat diartikan semacam gabungan atau kombinasi antara orang, perangkat keras, perangkat lunak, data, prosedur, dan pemrosesan informasi yang dioakai agar menghasilkan informasi yang diperlukan untuk mensupport operasi dan pengambilan keputusan pada suatu institusi/perusahaan.

2.1.2. Rancang Bangun

Dalam perspektif (Roger S. Pressman), perancangan merujuk pada serangkaian langkah prosedural yang digunakan untuk mengubah output analisis sistem menjadi kode pemrograman, dengan maksud menggambarkan dengan rinci bagaimana aspek-aspek sistem diwujudkan.

Sementara itu, konsep membangun atau pembangunan sistem, menurut (Roger S. Pressman), mencakup kegiatan pembuatan sistem baru atau modifikasi serta perbaikan sistem yang sudah ada, baik secara keseluruhan maupun sebagian.

Tahap rancang bangun (*design*) merujuk pada fase krusial dalam pengembangan perangkat lunak di mana perancang perangkat lunak merencanakan secara terperinci bagaimana perangkat lunak akan dibuat. Ini mencakup pemilihan struktur sistem, algoritma, tata letak komponen, dan berbagai keputusan teknis lainnya yang memungkinkan perangkat lunak mencapai persyaratan dan tujuannya. Proses rancang bangun ini memiliki signifikansi besar karena menentukan cara kerja perangkat lunak dan keberlanjutan fungsinya dalam jangka panjang. Kesalahan pada tahap ini dapat mengakibatkan masalah yang sulit diperbaiki pada tahap pengembangan perangkat lunak berikutnya. Oleh karena itu, pemahaman yang mendalam tentang prinsip-prinsip desain perangkat lunak dan praktik terbaik sangat diperlukan dalam rekayasa perangkat lunak

2.1.3. Aplikasi

Menurut pandangan (Roger S. Pressman), aplikasi merujuk pada perangkat lunak yang memberikan layanan atau fungsi tertentu untuk memenuhi kebutuhan atau tujuan tertentu. Jenis aplikasi dapat mencakup perangkat lunak bisnis, permainan komputer, aplikasi mobile, dan berbagai perangkat lunak lainnya. Dalam era teknologi informasi modern, aplikasi dapat bersifat beragam, termasuk aplikasi desktop, aplikasi web, aplikasi mobile, dan sebagainya. Aplikasi dirancang untuk memenuhi berbagai keperluan, mulai dari keperluan bisnis hingga hiburan, dan memiliki peran kunci dalam memenuhi kebutuhan pengguna dalam berbagai konteks.

2.1.4. Inventori

Inventori adalah kegiatan pengelolaan informasi yang mencakup barang-barang yang dipunyai oleh suatu institusi atau perusahaan, dimana bertujuan untuk mencatat, menjual dalam

sebuah kurun waktu usaha tertentu / menyimpan barang-barang yang masih dalam proses pembuatan atau pengerjaan (Rangkuti, 2004: 1).

Dengan penerapan sistem inventori, manajemen aset di perusahaan atau organisasi dapat beroperasi efisien. Pengawasan terhadap pergantian status aset, baik keluar maupun masuk, dapat dipantau dengan baik, dan keberadaannya tercatat secara akurat. Hal ini juga dapat mengurangi potensi penurunan jumlah aset karena faktor-faktor seperti kehilangan atau tidak dikembalikannya aset ke lokasi semula setelah digunakan.

2.1.5. SDLC (System Development Life Cycle)

SDLC atau Siklus Hidup Pengembangan Perangkat Lunak/Sistem yaitu suatu teknik yang diterapkan pada tahap pengembangan perangkat lunak, yang merinci tahapan-tahapan yang harus dilalui, mulai dari perancangan, pengembangan, pengujian, implementasi, hingga pemeliharaan perangkat lunak.

SDLC juga dapat diartikan suatu pendekatan yang digunakan oleh seorang pengembang sistem dalam merancang suatu sistem informasi, yang melibatkan aspek-aspek seperti kebutuhan, validasi, pelatihan, dan pemilik sistem (Prof. Dr. Sri Mulyani, 2017). Terdapat beberapa metode SDLC yang sering digunakan, yaitu:

1. *Waterfall Model:*

Fase-fase dalam model ini dilaksanakan secara berurutan, satu demi satu. Setiap langkah harus selesai sebelum memulai ke langkah berikutnya. Cocok pada proyek dengan kebutuhan yang sudah diketahui dan seimbang.

2. *Iterative Model:*

Pengembangan perangkat lunak dilakukan secara berulang-ulang (iteratif). Semua iterasi menyangkut fase analisis, design, implementasi, dan pengujian. Masing-masing iterasi meningkatkan fungsionalitas produk.

3. *Incremental Model:*

Proses pengembangan dipecah menjadi bagian-bagian kecil atau inkremental. Setiap inkrement memberikan penambahan fungsionalitas ke produk. Inkrement-inkrement berurutan hingga produk lengkap tercapai.

4. *Spiral Model:*

Model ini menggabungkan aspek iteratif dari pengembangan perangkat lunak dengan elemen kontrol dan risiko. Setiap iterasi berbentuk spiral dan melibatkan perencanaan, analisis risiko, teknis, dan evaluasi pelanggan.

5. *Agile Model:*

Agile merupakan pendekatan pengembangan perangkat lunak yang menekankan kerjasama tim dan responsibilitas terhadap perubahan.

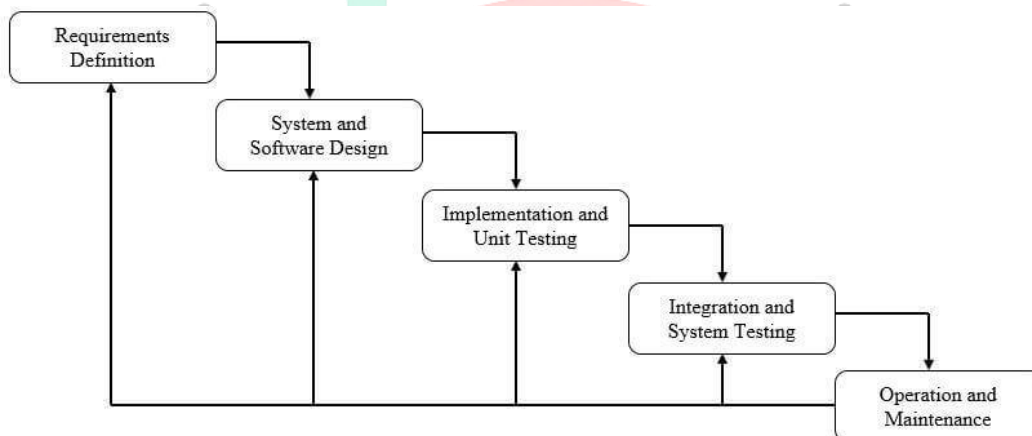
Proses iteratif dan inkremental, dengan fokus pada pengembangan perangkat lunak yang dapat dengan cepat beradaptasi dengan perubahan kebutuhan.

6. RAD (*Rapid Application Development*):

Pendekatan pengembangan yang mempromosikan penggunaan prototipe dan pengembangan cepat. Fokus pada pengembangan iteratif dan perubahan cepat berdasarkan umpan balik pengguna.

2.1.6. *Waterfall Model*

Dalam metode Waterfall, baik *Object-Oriented Analysis and Design* (OOAD) maupun *Structured Systems Analysis and Design* (SSAD) dapat digunakan tergantung pada kebutuhan dan karakteristik spesifik dari proyek yang sedang dijalankan. Dalam metode ini setiap tahap harus dikerjakan secara berurutan tanpa ada kembali ke tahap sebelumnya. Metode *Waterfall* mengacu pada siklus hidup proyek yang berjalan seperti aliran air dari atas ke bawah, mengalir melalui setiap tahap secara berurutan. Metode *waterfall* sebagai tahapan utama yang langsung menggambarkan dasar dalam pengembangan sistem. Ada beberapa tahapan perancangan dalam metode *waterfall* antara lain adalah analisis kebutuhan perangkat lunak, desain, implementasi dan unit *testing*, integrasi dan sistem *testing*, dan terakhir yaitu pemeliharaan (sommerville,2011).



Gambar 2. 1 Tahapan Metode *Waterfall*

Sumber : ranahresearch.com

1. *Requirements analysis*

Tahap ini bertujuan mengumpulkan kebutuhan secara lengkap untuk mengidentifikasi dan mendefinisikan kebutuhan sistem apa saja yang harus dicapai. Informasi dapat dilakukan antara lain melalui wawancara maupun melakukan survey kepada *user* nantinya.

2. *System and software Design*

Pada tahap ini, tim pengembangan perangkat lunak merencanakan struktur dan desain keseluruhan sistem berdasarkan persyaratan yang telah didefinisikan dalam tahap *Requirements Analysis* (Analisis Kebutuhan) sebelumnya.

3. *Implementation and Unit Testing*

Tahap *Implementation* adalah tahap di mana konsep dan perencanaan dari tahap *Requirements Analysis* dan *System Design* diwujudkan menjadi perangkat lunak yang sebenarnya. Pada tahap ini, fokus utama adalah pada penulisan kode perangkat lunak (*Coding*) yang berkualitas tinggi, pengujian unit, dan integrasi komponen sehingga sistem nantinya dapat berjalan dengan baik dan sesuai yang diinginkan.

4. *Intergration and System Testing*

Tahap ini bertujuan untuk memeriksa dan mengidentifikasi apakah komponen-komponen perangkat lunak yang telah dibangun terdapat masalah atau berinteraksi dengan baik satu sama lain saat digabungkan. Ini memastikan bahwa integrasi antara komponen berjalan dengan baik.

5. *Operation and Maintenance*

Tahap ini adalah tahap akhir dalam model pengembangan perangkat lunak *Waterfall*. Setelah perangkat lunak telah selesai dikembangkan dan diuji pada tahap sebelumnya, tahap ini berfokus pada penggunaan, pemeliharaan, dan dukungan jangka panjang pada perangkat lunak tersebut.

2.1.7. OOAD (*Object Oriented Analysis and Design*)

OOAD yaitu salah satu teknik dalam perancangan sistem. Metode ini berfokus pada pemodelan objek dan konsep pemrograman berorientasi objek. Melibatkan identifikasi objek, hubungan antar objek, dan pembuatan kelas untuk merancang sistem. OOAD biasanya didukung oleh bahasa pemodelan *Unified Modeling Language* (UML).

2.1.8. UML (*Unified Modeling Language*)

UML sering digunakan dalam konteks analisis dan desain berorientasi objek (OOAD). Melibatkan pemodelan konsep-konsep objek, hubungan antar objek, dan dinamika sistem menggunakan diagram-diagram UML. UML atau *Unified Modeling Language* adalah bentuk bahasa yang diterapkan dalam rekayasa perangkat lunak untuk tujuan perancangan, dokumentasi, dan pemodelan sistem perangkat lunak. UML adalah sebuah standar industri yang dipakai sebagai memvisualisasikan, merancang, dan mengkomunikasikan berbagai aspek dari perangkat lunak dan sistem informasi. *Unified Modeling Language* (UML), yang menyediakan notasi grafis untuk memodelkan objek, kelas, interaksi, dan komponen sistem secara visual. *Unified Modeling Language* (UML) dapat digunakan untuk mempermudah dalam mengembangkan aplikasi yang berkelanjutan (Hasanuddin, 2016). Dalam bentuk UML atau *Unified Modeling Language* dapat terdiri dalam kerangka *Use Case Diagram*, *Activity Diagram*, *Class Diagram* dan *Sequence Diagram* (Siti Mujilahwati, 2018).

2.1.9. Database

Database dapat diartikan sebagai sekumpulan data yang telah diatur secara terstruktur dan tersimpan dengan baik di dalam suatu sistem komputer. Keberadaan database memungkinkan pengelolaan data yang efisien serta mempermudah proses penyimpanan, akses, dan pembaruan informasi dengan menggunakan struktur yang terstruktur. Umumnya, sistem database terdiri dari tabel, kolom, dan baris yang membentuk struktur terorganisir.

2.1.10. Spesifikasi Basis Data

Spesifikasi basis data merupakan penjelasan terperinci mengenai persyaratan dan sifat-sifat sistem basis data yang akan dibuat. Isi spesifikasi melibatkan beragam unsur, seperti susunan data, keterkaitan antar tabel, keberlanjutan data, jenis data yang digunakan, dan ketetapan serta kebijakan yang mengatur manajemen data.





2.1.11. Alat Pendukung (*Tools System*)



Teori Pendukung merupakan alat bantu dalam mengilustrasikan suatu sistem. *Tools System* bertujuan untuk memberikan pemahaman kepada pengguna mengenai cara sistem informasi dapat beroperasi dengan menggunakan model logis (sistem yang sedang dikembangkan) dan model fisik (sistem yang sudah beroperasi).

1. Use Case Diagram

Use case diagram salah satu diagram dalam UML dimana dipakai untuk menjelaskan atau menggambarkan hubungan antara sistem yang dibuat dan dikembangkan dengan aktor-aktor yang berinteraksi pada sistem tersebut.

Tabel 2. 1 *Use Case Diagram* Simbol

Simbol	Nama	Keterangan
	<i>Actor</i>	Subjek yang mewakili peraninteraksi pada sistem.
	<i>Use Case</i>	Gambaran utama antara sistem danaktor.
	<i>Include relationship</i>	Include digunakan untuk menunjukkanhubungan antara dua kasus penggunaan.
	<i>Extend relationship</i>	Extend digunakan ketika salah satu use case memiliki aksi tambahan yang dapatterjadi, bersifat opsional.



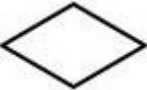


	<i>generalization</i>	Digunakan untuk spesialisasi aktor agar dapat berpartisipasi dalam Use Case.
	<i>association</i>	Penghubung antara <i>actor</i> dan <i>use case</i> .


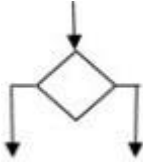
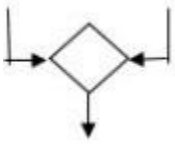

Sumber : Dokumentasi Pribadi

2. Activity Diagram

Activity Diagram yaitu diagram dalam UML dimana dipakai sebagai gambaran aliran kerja (*workflow*) pada suatu sistem atau proses bisnis. *Activity Diagram* bertujuan untuk memahami, merancang, dan mendokumentasikan bagaimana suatu tugas atau aktivitas dilakukan dalam suatu sistem.

Tabel 2. 2 *Activity Diagram* Simbol

Simbol	Nama	Keterangan
	<i>Initial state</i>	Simbol dimulainya alur suatu diagram aktivitas.
	<i>activities</i>	Aktivitas yang terjadi pada alur kerja.
	<i>Decision</i>	Simbol dimana adanya kemungkinan pada alur sistem.
	<i>Final state</i>	Simbol akhirnya alur suatu diagram aktivitas.
	<i>Swimlane</i>	Pembagian alur bisnis atas aktivitas yang terjadi.

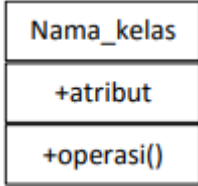
Simbol	Nama	Keterangan
	<i>Association</i>	Tanda untuk menunjukkan alur dari aktivitas diagram tersebut.
	<i>Decision node</i>	Mengambarkan adanya kemungkinan lebih dari satu dari alur pada sistem.
	<i>Merge node</i>	Menggabungkan kembali alur yang dipisah oleh <i>decision node</i> .
	<i>Final flow node</i>	Akhir dari alur.

Sumber : Dokumentasi Pribadi

3. Class Diagram

Class Diagram yaitu diagram lainnya dalam UML yang dipakai pada rekayasa perangkat lunak untuk mendeskripsikan struktur statis dalam sistem. Diagram yang berfokus pada entitas-entitas yang ada dalam sistem, hubungan antara entitas-entitas tersebut, dan atribut serta metode yang dimiliki oleh entitas-entitas tersebut.

Tabel 2. 3 *Class Diagram* Simbol

Simbol	Nama	Keterangan
	<i>Class</i>	Deskripsi dari struktur sistem.

Simbol	Nama	Keterangan
	<i>Association</i>	Garis yang menunjukkan bahwakelas-kelas tersebut berinteraksi atau berhubungan dalam suatu sistem.
	<i>Aggregation</i>	Hubungan antar kelas dengan makna semua bagian.
	<i>Generalization</i>	Hubungan antara dua atau lebih kelas yang menunjukkan bahwa kelas-kelas tersebut memiliki sifat atau perilaku yang sama atau tingkatan yang berbeda dalam suatu sistem.
	<i>Dependency</i>	Hubungan antara duakelas di mana perubahan pada kelas yang satu akan mempengaruhi kelas yang lainnya.


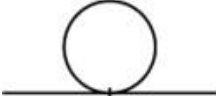
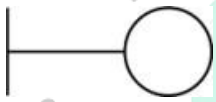


Sumber : Dokumentasi Pribadi


4. Sequence Diagram

Sequence Diagram salah satu jenis diagram dalam UML juga dimana dipakai untuk mendeskripsikan interaksi antara objek-objek pada sistem atau aplikasi. Biasanya alur berjalan objek-objek dari sistem yang digambarkan dalam diagram ini dibuat dari arah kiri ke arah kanan.

Tabel 2. 4 *Sequence Diagram* Simbol

Simbol	Nama	Keterangan

	<p><i>Actor</i></p>	<p>Subjek yang mewakili peran interaksi pada sistem.</p>
	<p><i>Entity Class</i></p>	<p>Simbol yang menggambarkan bagaimana pesan ataupun panggilan metode melibatkan objek dari kelas entitas tersebut.</p>
	<p><i>Boundary Class</i></p>	<p>Simbol yang digunakan untuk berinteraksi dengan sistem lain, baik berupa <i>user interface</i> dan lain sebagainya.</p>
	<p><i>Control Class</i></p>	<p>Digunakan untuk menggambarkan logika aplikasi yang tidak memiliki tanggung jawab kepada entitas.</p>
	<p><i>Recursive</i></p>	<p>Simbol di mana sebuah objek memanggil metode atau berinteraksi dengan objek yang memiliki tipe yang sama dengan dirinya sendiri.</p>

	<i>Lifeline</i>	Menandakan panjang objek.
---	-----------------	---------------------------

Sumber : Dokumentasi Pribadi

2.2. Tinjauan Studi

Literature review dilakukan agar penulis dapat mempelajari studi kasus atau pun pelajaran dari penelitian sebelumnya, baik dalam hal metodologi maupun hasil, dan menggunakan informasi ini untuk mengembangkan penelitian yang penulis buat untuk menjadi lebih baik.

1. Jurnal berjudul "**SISTEM INFORMASI INVENTORY BARANG PADA PT.MEDAN SMART JAYA BERBASIS WEB,**" yang ditulis oleh Teuku Yusrizal, Buyung Solihin Hasugian, dan Amru Yasir, diterbitkan dalam Universitas Dharmawangsa *Journal of Information System, Computer Science, and Information Technology, Volume 1, Nomor 2, Desember 2020*. Dalam jurnal tersebut, dijelaskan bahwa saat ini, sistem yang digunakan di PT. Medan Smart Jaya masih memiliki beberapa kelemahan dalam Penulisan catatan dan Penyusunan laporan barang keluar dan masuk secara manual. Hal ini membuat hasil kinerja perusahaan mengalami hambatan sehingga tidak dapat memenuhi semua keperluan yang dibutuhkan oleh perusahaan. Meskipun identifikasi masalah dijelaskan dalam artikel, namun tidak secara rinci. Kesimpulan yang diambil dari artikel ini adalah bahwa Sistem Informasi Inventori Barang di Peseroan Terbatas. Medan Smart Jaya yang berlandas *website* yang telah dibuat dan diimplementasikan oleh OS Windows, XAMPP sebagai server, MySQL untuk sistem database, dan PHP digunakan untuk bahasa pemrograman yang terintegrasi dengan HTML, serta Visual Studio Code sebagai editor web.
2. Jurnal kedua yang berjudul "**SISTEM INFORMASI MANAJEMEN INVENTORY SECARA ONLINE MENGGUNAKAN FRAMEWORK EASYUI,**" yang ditulis oleh Abdul Bari dan Kasmawi, dipublikasikan dalam jurnal Inovtek Polbeng - Seri Informatika, Volume 1, Nomor 1, Juni 2016. Dalam jurnal ini dijelaskan bahwa ada kebutuhan untuk memiliki pengelolaan barang dalam sistem informasi manajemen persediaan di UPTD kesehatan per-kecamatan di Kabupaten Bengkalis dan Dinas Kesehatan setempat. Meskipun tidak dijelaskan secara rinci, jurnal ini bertujuan untuk menciptakan sebuah sistem informasi manajemen *inventory control* secara *online* yang bisa menghasilkan informasi tentang persediaan, pencatatan penerimaan dan pengeluaran, dan laporan semester mengenai pemakaian inventaris yang dibuat sistematis sesuai dengan Peraturan Bupati Bengkalis nomor 44 tahun 2012. Kesimpulan dari jurnal ini adalah penggunaan sistem manajemen *inventory control* menyajikan penataan persediaan berlandas web yang dapat digunakan secara *online* oleh UPTD kesehatan per kecamatan di Kabupaten Bengkalis dan Dinas Kesehatan setempat.
3. Jurnal ketiga berjudul "**SISTEM INFORMASI PERSEDIAAN BARANG BERBASIS WEB PADA CV. PATRIOT KENCANA MEDIKA KUDUS,**" ditulis oleh Rizki Putra Wicaksono dan Anteng Widodo, diterbitkan dalam Jurnal Sistem Informasi & Manajemen Basis Data (SIMADA), Volume 3, Nomor 1, Maret 2020, oleh Jurusan Sistem Informasi Institut Informatika dan Bisnis Darmajaya. Pada jurnal ini dijelaskan bahwa CV. Patriot Kencana Medika menghadapi

masalah saat mencatat laporan barang masuk dan keluar, memakai aplikasi *stand-alone* dengan penyimpanan internal yang tidak terakses secara efektif untuk direktur, perusahaan, dan pemasok yang dapat memperbarui harga sewaktu-waktu. Sistem lama tersebut menggunakan metode Pre Order (PO) untuk memeriksa ketertidaksediaan barang sehingga proses order dilakukan secara manual via aplikasi pesan *mobile WhatsApp* oleh tim pemasaran. Dengan diterapkannya sistem informasi ini, PT. Patriot Kencana Medika di Kudus, direktur, dan tim penjualan dapat memperoleh informasi transparan mengenai jumlah barang di gudang, yang nantinya akan dipasarkan kepada pelanggan.

4. Jurnal selanjutnya berjudul "**RANCANG BANGUN SISTEM INFORMASI INVENTORY BARANG (SINBAR) BERBASIS WEBSITE,**" yang disusun oleh Agyztia Premana dan diterbitkan di *Information Technology Journal (INTECH) of UMUS, Volume 1, Nomor 02, November 2019, halaman 51-61.* Dalam jurnal ini, diuraikan permasalahan yang dihadapi oleh pusat teknologi informasi dan komunikasi di Universitas Muhadi Setiabudi, ydimana saat itu masih menggunakan pelaporan data inventaris secara manual. Pegawai melakukan pelaporan yang melibatkan kegiatan seperti pendataan barang yang baru maupun barang lama yang tetap dapat dipakai, bahkan barang rusak yang sudah harus diganti. Lalu data ini diproses menjadi laporan inventaris via aplikasi Microsoft Excel, lalu tahap pengumpulan data semacam ini dianggap kurang efisien. Kesimpulan yang diambil adalah bahwa dalam pengelolaan data inventaris di UPT. Pusat Teknologi Informasi dan Komunikasi di Universitas Muhadi Setiabudi, aplikasi yang dirancang lalu dibangun memakai metode UML (*Unified Modelling Language*) untuk representasi sistem yang telah dirancang. Dengan menerapkan sistem inventory ini, proses input, pemrosesan, dan output (laporan) menjadi lebih mudah, dan pengambilan data dapat dilakukan dengan lebih cepat.
5. Jurnal terakhir berjudul "**SISTEM INFORMASI INVENTARIS BARANG BERBASIS WEB PADA SMP NEGERI 1 BUER,**" yang ditulis oleh Novi Oktaviani, I Made Widiarta, dan Nurlaily, diterbitkan dalam Jurnal JINTEKS, Volume 1, Nomor 2, November 2019. Dalam jurnal ini, dibahas permasalahan pada SMP Negeri 1 Buer yang berhubungan dengan ketidakefektifan sistem *inventory*. Setiap barang yang ada di sekolah ini belum diberi nomor aset, tidak ada pendataan dari pembelian barang, dan alur proses peminjaman barang, serta alur pengembalian barang yang masih dilakukan dengan proses pencatatan secara manual pada buku laporan atau buku besar. Selain itu, untuk menduplikasikan data laporan tahunan dilakukan secara manual, mengakibatkan proses pengolahan data yang memakan waktu cukup lama. Masalah tersebut terlihat pada pencatatan barang, baik pada proses peminjaman barang atau pun pengembalian barang yang dianggap belum efektif. Penulis berhasil merancang Sistem Informasi Inventaris Barang Berbasis Web pada SMP Negeri 1 Buer memakai *Sublime Text*, bahasa pemrograman PHP, metode pengumpulan data melalui proses observasi dan juga wawancara, dan perancangan sistem memakai metode DFD. Sehingga penulis berharap struktur yang telah dikembangkan menghasilkan kontribusi dan mempermudah SMP Negeri 1 Buer saat mengelolah pendataan inventaris barang. Dengan demikian, ke depannya akan membantu administrasi Inventaris Barang dalam penomoran aset, pendataan barang, proses alur peminjaman dan alur pengembalian barang, dan pendataan barang atau aset dari transaksi pembelian.