

BAB II

TINJAUAN PUSAKA

2.1 Teori Terkait

Ririt Dwiputri Permatasari, Indah Kusuma Dewi, dan Agus Suryadi Tommy Saputra (2022) melaporkan bahwa *Purchase Order (PO)* sebagai dokumen resmi digunakan untuk memesan dan mengajukan permintaan pembelian. Dalam pengadaan barang, Fakultas Teknik sedang mengalami masalah pada sistem pengadaan barang berbentuk PO yang masih dilakukan secara manual. Barang yang diminta seperti melalui pengisian formulir yang telah ditentukan manual dan perlu disetujui oleh Wakil Dekan II khususnya saat bagian aset tidak ada di tempat. Hal ini menyebabkan manajer tidak bisa secara tepat dalam menganalisis dan secara aktual dan secara tepat dan akurat dalam merencanakan pengadaan barang. Secara sistematis, hal ini pengambilan keputusan berbasis informasi yang efektif dan akurat untuk perubahan yang berbasis sistem informasi. Untuk memenuhi tujuan tersebut, terdapat beberapa fokus yang perlu dikaji seperti, sistem IT otomatis mampu mengolah, merekap, dan mempresentasikan data secara langsung. Dengan implementasi sistem yang sudah ada, pengadaan barang fakultas akan lebih 'efektif' dan presisi sasaran pada sistem hemat PO.[2]

Ardian Dwi Praba dan Maryanah Safitri Faridi Sistem melakukan sebuah studi pada tahun 2024 yang menjelaskan bahwa informasi dalam proses manajemen akuisisi berkaitan dengan nilai informasi dalam mengotomatiskan beberapa langkah proses pembelian yang mencakup, tetapi

tidak terbatas pada, pemilihan pemasok, pengajuan pesanan pembelian, serta integrasi pelacakan pengiriman. Manfaat dari pemanfaatan kerangka kerja *Laravel* adalah pengurangan kesalahan input dan akses yang lebih cepat dan lebih mudah bagi pengguna, serta integrasi yang mulus dengan sistem periferal dalam organisasi. Sebagai kerangka kerja *PHP modern*, *Laravel* sangat dikenal karena skalabilitasnya, kemudahan penggunaan, fitur keamanan data yang baik, dan manajemen transaksi yang efektif yang semuanya berkontribusi secara substansial dalam menyederhanakan pembelian. Efisiensi sistem ini terbukti dengan berkurangnya kesalahan manual, percepatan alur kerja dalam pembelian, dan transparansi yang lebih besar di semua tahap proses pembelian. Dengan demikian, sebuah organisasi dapat sangat meningkatkan manajemen waktu dan sumber dayanya dengan memanfaatkan sistem berbasis *Laravel* yang pada gilirannya membantu meningkatkan efektivitas dan efisiensi keseluruhan dari proses manajemen pembelian.[3]

Dalam penelitian yang dilakukan Aini dan Risanty (2022), dijelaskan bahwa di zaman yang serba digital ini, informasi dan transaksi perdagangan dapat dilakukan secara mudah dan cepat secara online. *Purchase Order (PO)* adalah dokumen yang berfungsi dalam perdagangan yang memerlukan perhatian lebih pada karakteristik barang yang diacuhkan. Metode ini menghadapi berbagai masalah seperti rendahnya produktivitas karyawan, kelebihan dalam berapa jumlah paper use, serta meningkatnya kemungkinan hilangnya data yang telah dipesan, apalagi di dalam perusahaan yang jumlah

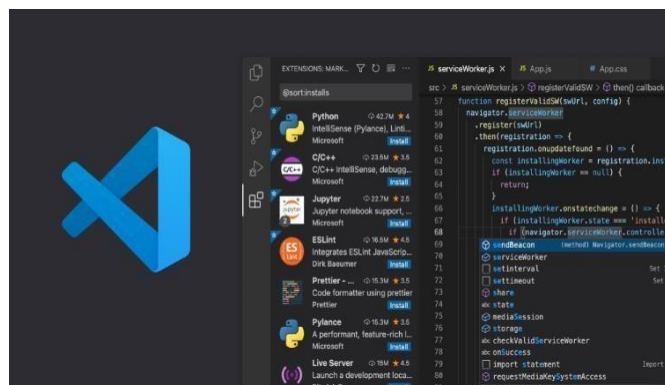
karyawannya terbatas. Oleh sebab itu, diperlukan sistem informasi yang bersifat digital untuk meningkatkan efisiensi kerja, menurunkan tingkat penggunaan kertas, menghentikan kehilangan data, dan memperbaiki dis administratif.[4]

Studi dari Azhari, Lukman Husein, Syepri Maulana Mabur, Nur Shobi Prihandoko, dan Aldi (2022) menjelaskan bahwa kebutuhan internal sebuah perusahaan dikelola oleh beberapa departemen seperti Pembelian yang bertanggung jawab untuk memenuhi kebutuhan perusahaan dan mengelola hubungan dengan pemasok. Untuk mendukung fungsi-fungsi ini, diperlukan sistem manajemen *Purchase Order* yang secara efisien mencatat, mengontrol, dan memonitor pengeluaran perusahaan. Solusi yang diusulkan adalah aplikasi berbasis *web* yang dapat diakses melalui *smartphone*, *tablet*, *PC*, atau laptop dan tidak memerlukan data disimpan secara lokal karena disimpan di *server* dan diakses menggunakan *browser*. Sistem informasi ini mengintegrasikan penangkapan dan pengolahan data untuk menghasilkan informasi yang berguna dengan kualitas tinggi dan sekaligus memberikan kebutuhan internal perusahaan akan fleksibilitas, aksesibilitas, dan pengelolaan efisien dalam proses *Purchase Order*. [5]

2.2 Landasan Teori

2.2.1 Visual Studio Code

Visual Studio Code adalah aplikasi editor kode buatan Microsoft yang dapat *digunakan* secara gratis di semua perangkat desktop. Kode editor ini sangat disukai oleh pengembang karena memiliki banyak fitur dan ekstensi. Bahkan hampir semua sistem operasi, seperti *Windows*, *Mac OS*, dan *Linux*, didukung oleh *Visual Studio Code*. [6] Mengembangkan sistem *Purchase Order* berbasis web di Els Laptop Tegal dengan Visual Studio akan meningkatkan produktivitas toko dan mengurangi ketergantungan pada sistem manual.



Gambar 2.1 *Visual Studio Code*

2.2.2 XAMPP

XAMPP adalah program open source berbasis *web server* yang dibuat oleh kelompok Apache Friend pada tahun 2002 dan tersedia secara gratis dengan lisensi *General Public License (GNU)*. *XAMPP* berfungsi sebagai *server* lokal dan mendukung berbagai sistem operasi populer, seperti *Linux*, *Windows*, *MacOS*, dan *Solaris*.

XAMPP menawarkan kemudahan dalam pengeditan, desain, dan pengembangan aplikasi dengan menjadi *server* standalone atau berdiri sendiri. *XAMPP* dianggap dapat menghemat uang karena dapat menggantikan peran *web hosting* dengan menyimpan file di *localhost*, yang dapat dipanggil atau dihubungkan melalui *browser*. Program-program yang tersedia dalam program ini dikenal sebagai *XAMPP*. [7]



Gambar 2.2 *XAMPP*

1. *X (Cross Platform)*

Kode ini berfungsi sebagai penanda untuk program yang berjalan di berbagai sistem operasi, termasuk *Windows*, *Linux*, *MacOS*, dan *Solaris*. [8]

2. *A (Apache)*

Ini berarti bahwa banyak orang dapat membuat *web server* gratis (open source) untuk membuat halaman *web* yang baik. Pengguna dapat menjalankan file *PHP* di *localhost* dengan menggunakan *web server*. [9]

3. *M (MySQL/MariaDB)*

MySQL adalah salah satu aplikasi database *server* yang menggunakan basa pemrograman *query language structured (SQL)* untuk mengelola data dengan cara yang sistematis dan terstruktur.

Mengolah, mengedit, dan menghapus daftar dari database adalah contohnya. Oleh karena itu, pengembang dan *programmer* dapat menggunakan komputer mereka untuk membuat aplikasi berbasis *web*. Sebaliknya, *MariaDB* adalah sistem manajemen database yang termasuk dalam pengembangan mandiri *MySQL*. [7]



Gambar 2.3 MySQL

4. *P (PHP)*

Hypertext Preprocessor (PHP) adalah bahasa pemrograman berbasis *web* yang digunakan untuk keperluan pada sisi *server* atau back end. Ini dapat digunakan untuk membuat halaman *web* lebih dinamis dengan menggunakan scripting sisi *server*. Selain itu, *PHP* mendukung manajemen sistem untuk *Oracle*, *Postgresql*, dan *Microsoft Access*, antara lain. [10]

5. *P (Perl)*

Perl banyak digunakan untuk pengembangan aplikasi hingga *web server* dan juga digunakan untuk pengembangan *website* pada sistem berbasis content management Sistem (CMS), seperti *PHP*. *Perl* juga sangat fleksibel dan banyak digunakan karena dapat berjalan di dalam banyak sistem operasi. [11]

2.2.3 *Laravel*

Laravel adalah *framework PHP* berlisensi *MIT* yang menggunakan konsep *Model-View-Controller (MVC)* untuk membuat aplikasi berbasis *MVP* yang menggunakan *PHP*. Tujuan dari *framework* ini adalah untuk meningkatkan kualitas perangkat lunak. Melalui sintaks yang jelas, ekspresif, dan hemat waktu, *Laravel* membantu mengurangi biaya awal pengembangan dan pemeliharaan.[3]



Gambar 2.4 *Laravel*

2.2.4 *Bootstrap*

Bootstrap adalah paket aplikasi yang dapat digunakan saat membuat front-end *website*. *Bootstrap* adalah *template* desain web dengan banyak fitur yang dirancang untuk membantu pengguna dari tingkat pemula hingga berpengalaman dalam desain web. Jika Anda memiliki pemahaman dasar *HTML* dan *CSS*, Anda dapat menggunakan *Bootstrap*. [12]



Gambar 2.5 *Bootstrap*

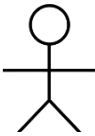
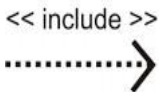
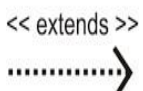
2.2.5 *UML*

Unified Modeling Language (UML) adalah sebuah teknik pengembangan sistem yang menggunakan bahasa grafis sebagai alat untuk mendokumentasikan dan menyesifikasikan sistem. Pada tahun 1994, Grady Booch dan James Rumbaugh menciptakan *UML* untuk menggabungkan dua metodologi terkenal, *Booch* dan *OMT*. Selanjutnya, Ivar Jacobson menciptakannya.[13]

Selanjutnya, *OOSE (Object Oriented Software Engineering)* bergabung. *OMG* membuat standar *UML*. Beberapa diagram *UML* yang paling umum digunakan saat mengembangkan sistem adalah:

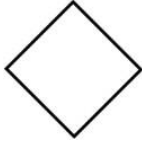
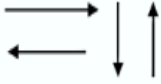
1. *Use Case: Diagram dalam UML (Unifield Modeling Language)* yang digunakan untuk menggambarkan interaksi antara pengguna (aktor) dan sistem melalui *Use Case* (fungsi atau fitur yang dapat digunakan oleh aktor).

Tabel 2.1 *Use Case*

NO	Simbol	Nama	Keterangan
1.		Actor	Menspesifikasikan himpauan peran yang pengguna mainkan ketika berinteraksi dengan <i>Use Case</i>
2.		Include	Menspesifikasikan bahwa <i>Use Case</i> sumber secara eksplisit.
3.		Extend	Menspesifikasikan bahwa <i>Use Case</i> target memperluas perilaku dari <i>Use Case</i> sumber pada suatu titik yang diberikan.
4.		Association	Apa yang menghubungkan antara objek satu dengan objek lainnya.
5.		System	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
6.		Use Case	Abstraksi dan interaksi antara sistem dan actor

2. *Activity Diagram*: Merupakan diagram yang bersifat statis, yang menggambarkan aktivitas dari suatu sistem bisnis. Untuk simbol dari diagram aktivitas bisa dilihat pada tabel.

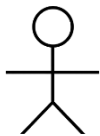
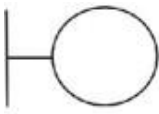
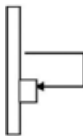
Tabel 2.2 *Activity Diagram*

NO	Simbol	Nama	Keterangan
1.		<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain.
2.		<i>Action</i>	State dari sistem yang mencerminkan eksekusi dari suatu aksi.
3.		<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.
4.		<i>Activity Final Node</i>	Bagaimana objek dibentuk atau diakhiri.
5.		<i>Decision</i>	Digunakan untuk menggambarkan suatu keputusan atau tindakan yang harus dilakukan pada kondisi tertentu.
6.		<i>Line Connector</i>	Digunakan untuk menghubungkan satu symbol.

3. *Sequence Diagram*: Adalah diagram dalam *UML (Unified Modeling Language)* yang digunakan untuk menggambarkan interaksi antar objek dalam suatu sistem berdasarkan urutan waktu.

Diagram ini menunjukkan bagaimana pesan dikirim antar objek untuk menyelesaikan suatu proses. atau fungsi. Diagramnya secara visual menampilkan aktor di sebelah kiri, sistem di tengah, dan database jika diperlukan.

Tabel 2.3 *Sequence*

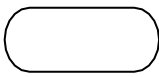
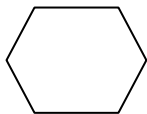
NO	Simbol	Nama	Keterangan
1.		<i>Actor</i>	Menggambarkan orang yang sedang berinteraksi dengan sistem.
2.		<i>Entity Class</i>	Gambaran sistem sebagai landasan dalam Menyusun basis data.
3.		<i>Boundary Class</i>	Menangani komunikasi antar lingkungan sistem.
4.		<i>Control Class</i>	Bertanggung jawab terhadap kelas-kelas terhadap objek yang berisi logika
5.		<i>Recursive</i>	Pesan untuk dirinya.
6.		<i>Activation</i>	Mewakili proses durasi aktivasi sebuah operasi.
7.		<i>Life Line</i>	Komponen yang digambarkan garis putus terhubung dengan objek.

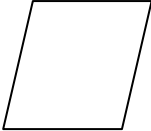
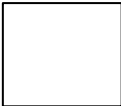
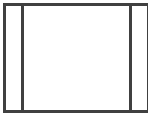
2.2.6 Flowchart

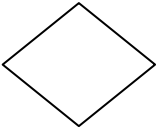
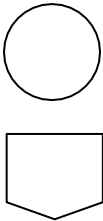
Jenis diagram yang menampilkan algoritma atau langkah-langkah instruksi yang berurutan dalam sebuah sistem disebut *flowchart*, atau juga disebut diagram alir. Seorang analis sistem menggunakan diagram flow untuk memberi *programmer* gambaran logis dari sistem yang akan dibangun. *Flowchart* membantu menyelesaikan masalah yang mungkin muncul selama proses pengembangan sistem. Sebuah *flowchart* biasanya digambarkan dengan simbol, dengan setiap simbol menggambarkan proses tertentu. Garis penghubung menghubungkan proses satu ke proses lainnya.[14]

Flowchart memudahkan penambahan proses baru dan membuat alur proses lebih mudah dipahami. Setelah *flowchart* dibuat, *programmer* akan menerjemahkan desain logis ke dalam program dengan menggunakan bahasa pemrograman yang telah diputuskan.[12]

Tabel 2. 4 *Flowchart*

NO	Simbol	Nama	Keterangan
1.		<i>Terminator</i>	Merupakan simbol yang digunakan untuk menentukan <i>state</i> awal dan <i>state</i> akhir suatu <i>flowchart</i> program.
2.		<i>Preparation</i>	Merupakan simbol yang digunakan untuk mengidentifikasi

NO	Simbol	Nama	Keterangan
			variabel-variabel yang akan digunakan dalam program.
3.		<i>Input output</i>	Merupakan simbol yang digunakan untuk memasukkan nilai dan untuk menampilkan nilai dari suatu variabel.
4.		<i>Process</i>	Merupakan simbol yang digunakan untuk memberikan nilai tertentu, apakah berupa rumus, perhitungannya <i>counter</i> atau hanya pemberian nilai tertentu terhadap suatu variabel.
5.		<i>Predefined Process</i>	Merupakan simbol yang penggunaannya seperti link atau menu. Jadi proses yang ada di dalam simbol ini harus di buatkan penjelasan <i>flowchart</i> programnya secara tersendiri yang terdiri dari terminator dan diakhiri dengan terminator.

NO	Simbol	Nama	Keterangan
6.		<i>Decision</i>	Digunakan untuk menentukan pilihan suatu kondisi (Ya atau tidak).
7.		<i>Connector</i>	Merupakan penghubung dari simbol yang satu ke simbol yang lain. Tanpa harus menuliskan arus yang panjang.
8.		<i>Arrow</i>	Merupakan simbol yang digunakan untuk menentukan aliran dari sebuah <i>flowchart</i> program.