

BAB II

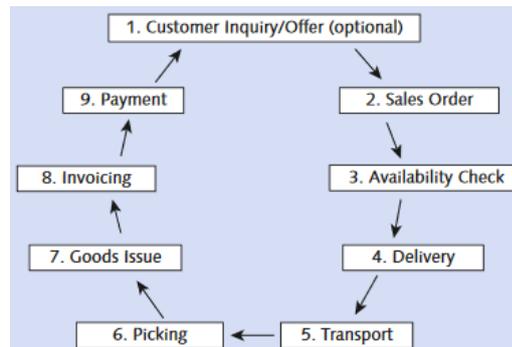
LANDASAN TEORI

2.1. Penjualan

Syarat utama berjalannya suatu usaha merupakan penjualan, karena dengan penjualan maka keuntungan akan didapatkan. Semakin besar penjualan maka keuntungan yang akan didapat akan semakin baik (Schulz, 2017). Penjualan juga merupakan penghubung antara bisnis dengan pelanggannya.

Menurut (Schulz, 2017) proses penjualan melalui proses yang meliputi:

1. *Customer inquiry*
2. *Sales order*
3. *Availability check*
4. *Delivery*
5. *Transport*
6. *Picking*
7. *Goods issue*
8. *Invoicing*
9. *Payment*



Gambar 2. 1 Proses Bisnis Penjualan

Sumber: (Schulz, 2017)

2.2.Sistem Penjualan Kredit

Transaksi dengan pembayaran secara bertahap atau melalui angsuran dengan durasi waktu yang telah ditentukan antara pembeli dan penjual dan adanya pembebanan bunga kredit kepada pembeli yang dibayarkan secara bersamaan disebut sistem penjualan kredit (Uci Fitriani, 2016).

Berikut merupakan langkah prosedur yang dibentuk sistem akuntansi penjualan kredit menurut (Iswoyo, 2013), diperlihatkan sebagai berikut:

- Penjualan
- Persetujuan Kredit
- Pengiriman
- Penagihan
- Pencatatan Piutang
- Distribusi Penjualan
- Pencatatan Bahan Penjualan

2.3.Manajemen Persediaan

Sistem untuk mengelola persediaan disebut manajemen persediaan. Bagaimana barang-barang persediaan dapat dikelompokkan dan seberapa akurat catatan persediaan dapat dijaga. Kemudian, akan diamati kontrol persediaan dalam bidang pelayanan. Manajer operasi di seluruh dunia telah mengetahui bahwa manajemen persediaan yang baik sangatlah bernilai. Di satu sisi, sebuah perusahaan dapat memperkecil biaya dengan memperkecil persediaan. Di sisi lain, produksi dapat dihentikan dan pelanggan menjadi tidak puas saat sebuah produk tidak tersedia. Menentukan keseimbangan antara investasi persediaan dengan pelayanan pelanggan adalah tujuan dari manajemen persediaan. Strategi dengan biaya rendah tidak akan bisa tercapai tanpa manajemen persediaan yang baik (Heizer Jay, Render Barry, 2015).

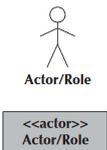
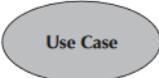
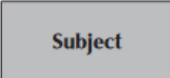
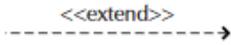
2.4.Unified Modeling Language

Unified Modeling Language atau yang sering disingkat dengan UML adalah kumpulan standar model pembuatan dan notasi yang dijelaskan oleh *Object Management Group* (OMG), merupakan organisasi standar dalam pembuatan sistem (Satzinger et al., 2016). Ahli lain menyimpulkan bahwa UML adalah sebuah kamus umum untuk *object-oriented* dan teknik membuat diagram yang kaya akan model yang digunakan dalam pengembangan sistem dari tahap perencanaan sampai implementasi (Dennis Alan et al., 2014).

2.4.1. Use Case Diagram

Use Case Diagram merupakan model UML yang dipakai guna menggambarkan penggunaan dan kaitannya dengan pengguna (Satzinger et al., 2016). Sementara (Dennis et al., 2014) berpandangan bahwa *Use Case* berguna untuk mengidentifikasi dan berkomunikasi antara sistem dengan kebutuhan kepada programmer yang akan membangun sistem. Berikut merupakan lambang yang digunakan pada *Use Case Diagram*:

Tabel 2. 1. Simbol Use Case Diagram

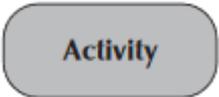
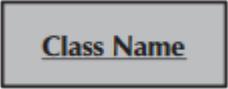
Simbol	Keterangan
	<i>Actor</i> adalah pengguna atau sistem yang mendapatkan manfaat dan adalah subjek eksternal.
	<i>Use Case</i> menggambarkan bagian utama dari fungsionalitas sistem.
	<i>A subject boundary</i> yaitu representasi <i>scope</i> dari sebuah subjek.
	<i>Association relationship</i> untuk menghubungkan interaksi antara <i>Actor</i> dengan <i>Use Case</i> .
	<i>Include relationship</i> ialah penyertaan fungsionalitas <i>Use Case</i> satu dengan lainnya.
	<i>Extend relationship</i> mengilustrasikan tambahan dari <i>Use Case</i> untuk memasukkan <i>optional behavior</i> .
	<i>Generalization relationship</i> untuk menghubungkan <i>Use Case</i> yang khusus kepada yang lebih umum.

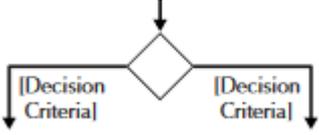
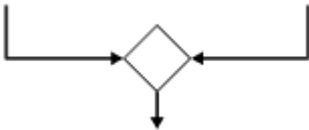
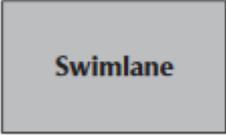
Sumber: (Dennis et al., 2014)

2.4.2. Activity Diagram

Activity Diagram menjelaskan interaksi dari setiap proses bisnis yang digambarkan menggunakan *Use Case* dengan lebih jelas dan detail (Wazlawick, 2014). Dengan diagram ini, berbagai aktivitas yang dilakukan aktor bisnis dan pekerja untuk mencapai tujuan akan dijelaskan lebih spesifik. Ahli lain berpendapat bahwa *Activity Diagram* adalah kumpulan deskripsi yang menggambarkan proses atau fungsionalitas eksternal yang butuh disediakan oleh sistem (Dennis Alan et al., 2014). Berikut adalah simbol-simbol yang digunakan pada *Activity Diagram*:

Tabel 2. 2. Simbol Activity Diagram

Gambar	Keterangan
	<i>Action</i> adalah <i>behavior</i> yang tidak terurai.
	<i>Activity</i> untuk menggambarkan aktivitas.
	<i>Object node</i> merepresentasikan sebuah objek yang terhubung dengan kumpulan aliran objek.
	<i>Control flow</i> menunjukkan bagian dari <i>behavior</i> .
	<i>Object flow</i> memperlihatkan aliran dari sebuah objek antara aktivitas.
	<i>Initial node</i> ialah permulaan awal dari serangkaian kegiatan.
	<i>Final-activity node</i> berguna untuk menghentikan serangkaian kegiatan.

Gambar	Keterangan
	<i>Final-flow node</i> digunakan untuk menghentikan kegiatan tertentu.
	<i>Decision node</i> memperlihatkan kondisi dari proses.
	<i>Merge node</i> mengembalikan alur pemilihan yang berbeda menjadi satu.
	<i>Fork node</i> membagi <i>behavior</i> menjadi kegiatan paralel.
	<i>Join node</i> mengembalikan <i>behavior</i> yang paralel.
	<i>Swimlane</i> membagi aktivitas individual <i>Activity Diagram</i> menjadi baris atau kolom kepada objek yang bertanggung jawab mengerjakannya.

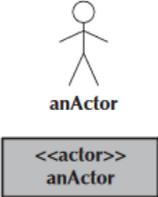
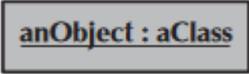
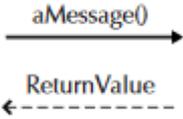
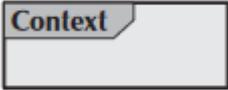
Sumber: (Dennis et al., 2014)

2.4.3. Sequence Diagram

Sequence Diagram adalah diagram yang dapat berguna untuk merepresentasikan tahapan dari proses dari sistem yaitu *events* dan *returns* dari *Use Case* (Wazlawick, 2014). Ketika *Sequence Diagram* digunakan untuk menggambarkan *Use Case Diagram* maka disebut dengan sistem *Use Case* (Wazlawick, 2014). Ahli lain menyebutkan bahwa *Sequence Diagram* mengilustrasikan objek yang ikut berpartisipasi dalam *Use Case* dan pesan yang diberikan antara keduanya dalam satu waktu dalam sebuah *Use Case*

(Dennis et al., 2014). Berikut adalah simbol-simbol yang digunakan pada *Sequence Diagram*:

Tabel 2. 3. Simbol *Sequence Diagram*

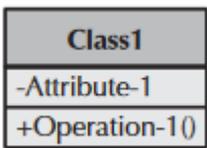
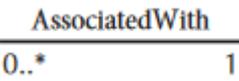
Gambar	Keterangan
	<p><i>Actor</i> adalah pengguna atau sistem yang mendapatkan manfaat dan adalah subjek eksternal.</p>
	<p><i>Object</i> ikut serta dalam proses dengan mengirim atau menerima pesan.</p>
	<p><i>Lifeline</i> menunjukkan hidup dari sebuah objek dalam proses.</p>
	<p><i>Execution occurrence</i> dipakai ketika objek mengirim atau menerima pesan.</p>
	<p><i>Message</i> berguna untuk menyampaikan pesan antara objek.</p>
	<p><i>Guard condition</i> menggambarkan sebuah tes harus dilakukan dengan mengirim pesan.</p>
	<p><i>Object destruction</i> untuk menghapus objek.</p>
	<p><i>Frame</i> dapat mengindikasikan konteks dari <i>Sequence Diagram</i></p>

Sumber: (Dennis et al., 2014)

2.4.4. Class Diagram

Class Diagram adalah ilustrasi dari unit informasi dari sebuah sistem (Wazlawick, 2014). Selain itu, model konseptual dari sebuah UML digambarkan menggunakan *Class Diagram* (Wazlawick, 2014). *Class Diagram* adalah sebuah model statis yang menunjukkan *class* dan hubungan dari banyak *class* secara konstan seiring berjalannya sistem. Berikut adalah simbol-simbol yang digunakan pada *Class Diagram*:

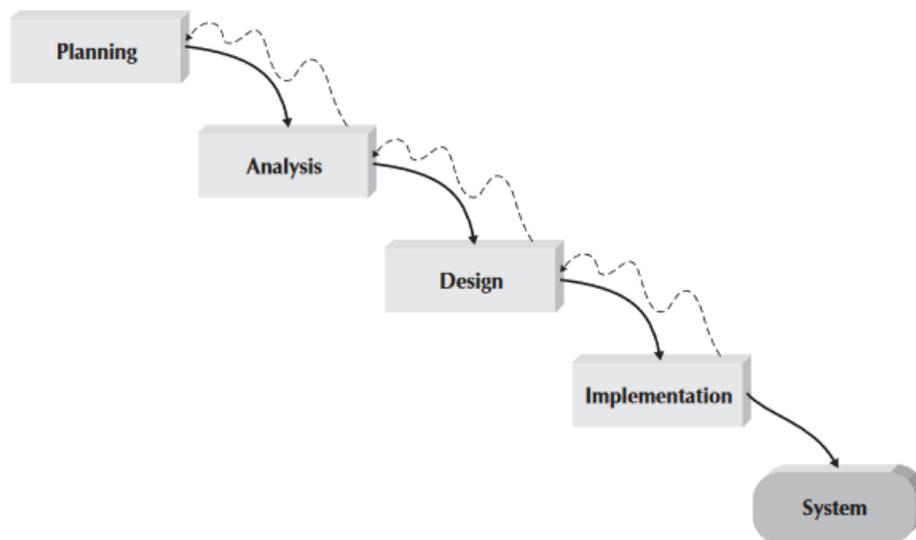
Tabel 2. 4. Simbol *Class Diagram*

Simbol	Keterangan
	<i>Class</i> menggambarkan aktor, tempat, dan penjelasan lain yang informasinya tidak akan disimpan dalam sistem.
attribute name /derived attribute name	<i>Attribute</i> sebagai <i>properties</i> yang menjelaskan keadaan dari sebuah objek.
operation name ()	<i>Operation</i> ialah aksi atau fungsi yang dapat dieksekusi oleh <i>class</i> .
	<i>Association</i> untuk membuat hubungan antara beberapa <i>class</i> .
	<i>Generalization</i> untuk menampilkan hubungan antara beberapa <i>class</i> .
	<i>Aggregation</i> untuk memperlihatkan hubungan <i>logical</i> antara <i>class</i> .
	<i>Composition</i> untuk representasi hubungan fisik antara <i>class</i> .

Sumber: (Dennis et al., 2014)

2.5. Metode Waterfall

Metode Waterfall merupakan metode pengembangan sistem asli (masih digunakan saat ini) adalah pengembangan seperti air terjun. Dengan metodologi berbasis pengembangan air terjun, analis dan pengguna melanjutkan secara berurut dari sebuah tahap ke tahap selanjutnya (Dennis Alan et al., 2014). Model Waterfall adalah aliran sekuensial linier yang kemajuan dilihat mengalir ke bawah melalui tahap implementasi *software*. Hal tersebut menunjukkan bahwa setiap tahap dalam langkah pengembangan dimulai hanya jika fase sebelumnya telah diselesaikan (Egwoh & Nonyelum, 2010). Ketika masalah dipahami dengan cukup baik dan ketika alur pekerjaan berproses linear, model Waterfall digunakan (Salve et al., 2018).

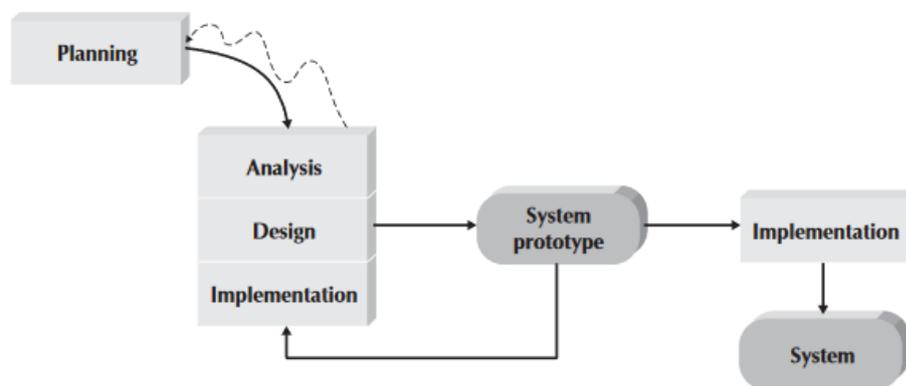


Gambar 2. 2. Gambar Metode Waterfall

Sumber: (Dennis et al., 2014)

2.6. Metode Prototyping

Prototyping merupakan metode berbasis prototyping dengan tahap analisis, desain, dan implementasi dilakukan secara bersamaan. Ketiga tahap dikerjakan secara berulang dalam sebuah proses sampai sistem telah selesai (Dennis Alan et al., 2014). Prototyping merupakan metode yang cocok untuk mendesain UI (*User Interface*) dengan baik dan salah satu penggunaan metode ini relatif cepat yang paling banyak digunakan saat ini adalah sebagai *tool* untuk merekayasa kebutuhan pengguna yang berulang dan desain UI (Salve et al., 2018). Prototipe digunakan untuk memvisualisasikan beberapa komponen perangkat lunak untuk menghilangkan kesalahpahaman *requirement* antara pengguna dengan tim *developer* (Egwoh & Nonyelum, 2010).

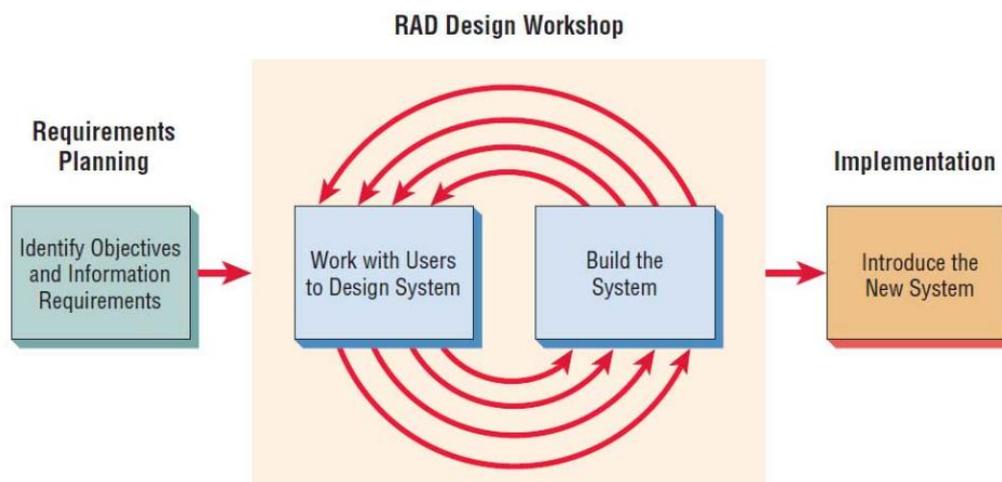


Gambar 2. 3. Gambar Metode *Prototyping*

Sumber: (Dennis et al., 2014)

2.7. Metode RAD

Metode RAD dibuat sebagai solusi dari kedua kelemahan metode desain terstruktur dengan menyesuaikan tahap pengembangan sistem untuk mendapatkan bagian-bagian dari sistem yang dibuat secara cepat hingga tangan pengguna. Dengan metode ini, sistem dapat lebih dipahami pengguna dan revisi sistem akan lebih mudah dilakukan karena pengguna dan pembang yang lebih dekat (Dennis Alan et al., 2014). RAD adalah metode yang digunakan dalam siklus pengembangan aplikasi yang cepat dan juga memberikan kualitas perangkat lunak yang baik dibandingkan dengan pendekatan rekayasa perangkat lunak tradisional. Melalui proses pengembangan aplikasi yang cepat, organisasi dapat mengurangi pengembangan perangkat lunak dan biaya pemeliharaan (Naz & Khan, 2015).



Gambar 2. 4. Gambar Metode RAD

Sumber: (Putri & Effendi, 2018)

2.8.Wawancara

Situasi berhadap-hadapan diantara pewawancara dan narasumber yang bertujuan untuk mendapatkan informasi yang diinginkan, dan bertujuan mendapatkan data tentang narasumber dengan bias yang minimum dan efisiensi maksimum disebut wawancara (Hakim, 2013).

Berdasarkan pendapat (Hakim, 2013) bahwa ada 2 jenis wawancara yaitu:

- Wawancara Formal (*Structured Interview*)

Langkah yang sistematis untuk mendapatkan informasi mengenai narasumber pada keadaan dimana beberapa pertanyaan akan ditanyakan secara berurutan dan telah dipersiapkan oleh pewawancara dan jawabannya dicatat dengan format yang telah ditentukan.

- Wawancara Informal (*Unstructured Interview*)

sebuah wawancara dimana belum dipersiapkan terlebih dahulu pertanyaannya, pertanyaan yang tidak berurut, dan pewawancara yang memiliki kuasa untuk memilih pertanyaan sesuai dengan topik utama.

2.9.HTML (*HyperText Markup Language*)

HTML merupakan bahasa web. Ketika membuka *web page* di browser web seperti Internet Explorer, Firefox, atau Safari, browser mengunduh dan menampilkan HTML (Steven & Janet, 2013). HTML merupakan standar bahasa yang digunakan untuk mengakses halaman web. Berdasarkan (Steven & Janet, 2013) HTML dapat melakukan beberapa hal sebagai berikut:

- mengurus tampilan dari halaman web dan kontennya.

- Membuat tabel.
- Mempublikasikan web secara daring.
- Membuat formulir yang dipakai untuk registrasi dan transaksi.

2.10. CSS (*Cascading Style Sheet*)

CSS berpasangan HTML dengan memberikan hiasan atau *style* pada halaman web. Halaman HTML yang dibuat tanpa CSS akan tampak cukup sederhana, dengan *font* dan ukuran *font default*. Menggunakan CSS, dapat meningkatkan tampilan web, menambahkan gambar pada latar belakang dan warna, mengubah *font style* dan *font size*, menambahkan gambar batas di sekitar area, bahkan mengubah letak halaman tersebut (Steven & Janet, 2013).

2.11. JavaScript

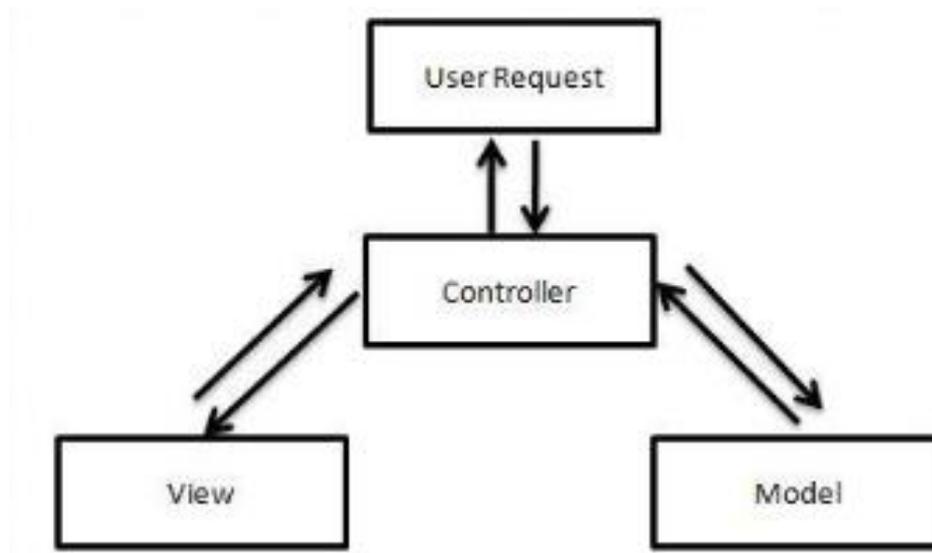
JavaScript digunakan oleh banyak web *browser* dan tak membutuhkan instalasi lainnya. Membuat halaman menjadi baik, dengan formulir terisi secara otomatis, dan jenis elemen yang menambahkan pengalaman pengguna dapat dilakukan dengan Javascript (Steven & Janet, 2013).

2.12. PHP

Bahasa *scripting* yang dibuat secara khusus untuk dipakai di web adalah PHP. PHP mempunyai fitur yang berguna saat pemrograman tugas yang diperlukan untuk pengembangan web yang dinamis (Steven & Janet, 2013). Sementara menurut (Ayu & Permatasari, 2018) PHP adalah skrip bahasa yang dapat digunakan atau disisipkan pada HTML.

2.12.1. CodeIgniter

CodeIgniter merupakan salah satu *framework* PHP. *Framework* ini dibuat didasarkan pada konsep *model view controller* (MVC). *CodeIgniter* mempunyai *library* yang cukup lengkap untuk mengerjakan tugas yang umum dan digunakan oleh rata-rata aplikasi berbasis web contohnya mengakses *database*, validasi formulir sehingga sistem dapat lebih mudah dikembangkan (Destiningrum & Adrian, 2017).



Gambar 2.5 Model View Controller

Sumber : (Destiningrum & Adrian, 2017)

2.13. MySQL

Banyak situs web dinamis membutuhkan *database backend*. Basis data dapat berisi informasi yang ditampilkan halaman web kepada pengguna, atau tujuan *database* untuk menyimpan informasi yang disediakan oleh pengguna dan MySQL

merupakan *database* paling populer untuk digunakan di situs web, dikembangkan menjadi cepat dan kecil, khusus untuk situs web (Steven & Janet, 2013). (Ayu & Permatasari, 2018) berpandangan MySQL merupakan sistem yang dapat mengatur *database SQL, open source*, dan paling populer saat ini. Fitur – fitur seperti *multi-user, multithreaded*, dan sistem manajemen *database SQL* didukung oleh MySQL.

2.14. Visual Studio Code

Visual Studio Code adalah *tools* untuk *text editor* yang memiliki banyak fitur dan fungsi. Meskipun baru secara resmi dirilis selama tiga tahun, Visual Studio Code dengan cepat menjadi salah satu *editor* top dalam hal popularitas, bersaing dengan *Sublime Text, Atom, dan UltraEdit* untuk posisi teratas (Johnson, 2013).

2.15. Xampp

XAMPP adalah *tools all-in-one* populer yang menginstal Apache, MySQL, dan PHP dalam satu prosedur. XAMPP juga menginstal phpMyAdmin, aplikasi web yang dapat digunakan untuk mengelola *database MySQL* dan XAMPP biasa digunakan untuk pengembangan aplikasi web pada komputer lokal pengembang aplikasi (Steven & Janet, 2013). Xampp adalah sebuah paket kumpulan *software* yang terdiri dari Apache, MySQL, PhpMyAdmin, PHP, Perl, Filezilla, dan lain (Ayu & Permatasari, 2018).

2.16. PhpMyAdmin

Aplikasi atau perangkat lunak yang bebas digunakan (*open source*) dan ditulis dengan menggunakan bahasa pemrograman PHP untuk dapat menangani

administrasi *database* MySQL dengan jaringan lokal maupun internet yaitu phpMyAdmin. berbagai tugas MySQL dapat dijalankan menggunakan phpMyAdmin, diantaranya (mengatur data, tabel, bidang (*fields*), hubungan (*relations*), indeks, pengguna, perizinan (*permissions*), dan masih banyak lagi (Standisyah & N.S Restu, 2017).

2.17. Penelitian Terdahulu

Tabel 2. 5. Penelitian Terdahulu

No	Nama Journal, Vol, No, Tahun	Penulis	Hasil Penelitian	Kontribusi di Penelitian ini
1	Jurnal Evolusi, 2018, Vol. 6, No. 2.	Oky Irnawati, Galih Bayu Aji Listianto.	Sistem berbasis web untuk <i>Inventory</i> .	Referensi untuk perancangan sistem.
2	Jurnal IDEALIS, 2018, Vol. 1, No .3.	Muhammad Alif Ramadhansyah, Dian Anubhakt.	Sistem informasi penjualan dan pembelian bahan bangunan.	Referensi untuk fitur-fitur yang akan ditanamkan ke dalam sistem yang dibuat.
3	ULTIMA InfoSys, 2019, Vol. 10, No. 1	Listiani, Friska Natalia	Aplikasi <i>Manpower Request</i> untuk bagian <i>Recruitment</i> .	Referensi untuk pembangunan sistem dengan metode <i>Rapid Application Development</i>

Sumber : (Irnawati & Listianto, 2018; Listiani & Natalia, 2019; Ramadhansyah & Anubhakti, 2018)

Perbedaan penelitian ini dengan penelitian-penelitian sebelumnya. Beberapa perbedaannya yaitu dari segi integrasi dan pengingat menggunakan notifikasi. Sistem yang saat ini dirancang mengintegrasikan seluruh proses bisnis penjualan yang melibatkan bagian penjualan dan gudang. Ada juga fitur yang belum ada pada penelitian sebelumnya yaitu pengingat berupa notifikasi. Fitur notifikasi berguna untuk memberikan pemberitahuan ketika ada proses yang harus segera diproses sehingga pengguna tidak harus melakukan pengecekan seluruh data. Selain itu, penggunaan metode RAD pada pengembangan sistem ini mengambil dari penelitian sebelumnya yang digunakan untuk membuat sistem HR sebagai panduan untuk alur pengembangan sistem.