

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Berdasarkan permasalahan yang ada, maka terdapat referensi-referensi yang dapat membantu dalam berjalannya penelitian ini. Berikut merupakan beberapa penelitian terdahulu.

1. PERANCANGAN SISTEM INVENTORY PADA PT. PALOH SINGKWANG STABAT BERBASIS *WEB* PHP DENGAN METODE *EXTREME PROGRAMMING.*, Dedi Wahyuman, TUGAS AKHIR FAKULTAS SAINS DAN TEKNOLOGI UNIVERSITAS PEMBANGUNAN PANCA BUDI MEDAN 2022.

Penelitian tersebut bertujuan untuk merancang sistem inventory barang yang sesuai dengan permasalahan pada kegiatan inventory pada PT. Paloh Singkwang Stabat agar mempermudah admin dan bagian gudang dalam mengelola dan mendata barang yang masuk, barang keluar, barang retur dan laporan persediaan barang. Hasil dari pembangunan sistem inventory ini dapat mempermudah admin dan staf gudang dalam mengelola dan mendata barang masuk, barang keluar, barang retur dan laporan persediaan barang.[4] Perbedaan antara penelitian tersebut dan penelitian yang akan dilakukan adalah pada penelitian tersebut berfokus pada manajemen sistem inventory barang dari PT. Paloh Singkwang Stabat, sedangkan pada penelitian yang akan dilakukan berfokus pada perancangan sistem informasi pemesanan dari PT. AIM Purwokerto.

2. METODE AGILE PADA PEMBUATAN *WEBSITE SERVICES AND SALES* PRINTER INK., WIDIARINI, FAJAR AGUSTINI, RISTYANTO SISWO WIDIYANTO, SARI HARTINI, *INFORMATION MANAGEMENT FOR EDUCATORS AND PROFESSIONALS* 2022.

Penelitian tersebut bertujuan untuk membangun *web e-commerce* yang dapat melayani *Services and sales* tinta printer dari Toko Sip Print. Dengan

pembangunan *web e-commerce* memungkinkan pelanggan untuk berbelanja atau melakukan transaksi selama 24 jam, memberikan banyak pilihan produk dari beberapa merk, melakukan perbandingan secara cepat dan pelanggan dapat menerima informasi relevan secara detail dalam hitungan jam, bukan atau minggu. Dari segi aspek manajerial dalam pembuatan laporan dapat memudahkan karyawan untuk mengelola laporan karena sudah terkomputerisasi dan secara *online*, kemudian dari segi aspek sistem dalam proses penjualan sudah menggunakan media komputer yang terhubung dengan jaringan internet yaitu sistem penjualan berbasis *web* yang dapat memudahkan customer melakukan transaksi jarak jauh.[5]

Perbedaan antara penelitian tersebut dan penelitian yang akan dilakukan adalah pada penelitian tersebut menggunakan metode agile dan berfokus pada service dan penjualan printer dari toko Sip Print sedangkan penelitian yang akan dilakukan menggunakan metode *EXTREME PROGRAMMING* (XP) dan berfokus pada perancangan sistem informasi pemesanan dari PT. AIM Purwokerto.

3. IMPLEMENTASI METODE *EXTREME PROGRAMMING* (XP) PADA APLIKASI PERSEDIA STOK GAS 3KG DI PT DINDA KARYA SABPUTRI BERBASIS *WEB.*, DINI HARI PERTIWI, MANAJEMEN INFORMASI 2017.

Penelitian tersebut bertujuan untuk membangun *web* persediaan stok (inventory) yang dapat melakukan pengecekan persediaan tabung gas dan dapat melakukan pengecekan jadwal pengiriman tabung gas. Pada penelitian ini diperoleh kesimpulan dengan adanya aplikasi ini dapat memudahkan pegawai dalam mengontrol stok tabung LPG 3kg karena stok tabung dapat di update secara otomatis, Dengan adanya aplikasi ini pangkalan dapat mengecek jadwal pengiriman karena pangkalan dapat mengetahui kapan pengiriman dilakukan dari PT Dinda Karya Sabputri.[6] sedangkan pada penelitian yang akan dilakukan berfokus pada perancangan sistem informasi pemesanan dari PT. AIM Purwokerto.

4. SISTEM INFORMASI PEMESANAN TIKET WISATA KETENGERADVENTURE BERBASIS *WEBSITE* DENGAN METODE WATERFALL STUDI KASUS :WISATA KETENGER DI DESA KETENGER., RIPTA SEPTIYANA, TUGAS AKHIR PROGRAM STUDI TEKNIK INFORMATIKA FAKULTAS INFORMATIKA INSTITUT TEKNOLOGI TELKOM PURWOKERTO 2021.

Penelitian tersebut bertujuan untuk merancang dan membangun sistem informasi pemesana tiket, pengelolaan kegiatan, pelaporan, dan pencatatan jadwal di Ketenger Adventure. Pada *website* ini menggunakan metode waterfall sebagai metode pengembangnya. Penelitian tersebut menghasilkan sistem informasi yang dapat membantu pengunjung dalam melakukan pemesanan tiket, membantu admin dalam mengelola data pesanan, memudahkan tim dalam melihat jadwal kegiatan dan membantu manajer dalam melakukan pelaporan data.[7]

Perbedaan antara penelitian tersebut dan penelitian yang akan dilakukan adalah pada penelitian tersebut menggunakan metode waterfall dan focus dari penelitian tersebut ada pada pemesanan tiket dan pengelolaan data Wisata Adventure Di Desa Ketenger, sedangkan penelitian yang akan dilakukan akan menggunakan metode *EXTREME PROGRAMMING* (XP) dan berfokus pada perancangan sistem informasi pemesanan dari PT. AIM Purwokerto.

5. PENERAPAN *EXTREME PROGRAMMING* PADA SISTEM INFORMASI PENJUALAN PAKAIAN BERBASIS *WEB* (STUDI KASUS TOKO ST JAYA)., IRMAWATI CAROLINA, ARIEF RUSMAN, NOVTEK POLBENG - SERI INFORMATIKA 2019.

Penelitian tersebut bertujuan untuk merancang untuk membuat sebuah sistem informasi berbentuk aplikasi *web* yang dapat mengolah data pakaian termasuk pada proses penjualannya dan juga pada proses laporan. Penulis menggunakan metode *Extreme programming* untuk membuat aplikasi *web* ini. Hasil dari penelitian tersebut adalah pembelian pakaian secara *online* ini

dapat memberikan salah satu alternatif dan kemudahan dalam menjalankan bisnis *online*, meningkatkan pelayanan kepada pelanggan serta dapat membuat pelanggan mudah mendapatkan informasi, Dengan menggunakan komputer beserta perangkat pendukungnya yang baik maka cara kerja dapat menjadi lebih efisien dan efektif.[8]

Perbedaan antara penelitian tersebut dan penelitian yang akan dilakukan adalah pada penelitian tersebut berfokus pada meningkatkan penjualan produk dan manajemen produk pada toko ST Jaya sedangkan pada penelitian yang akan dilakukan berfokus pada perancangan sistem informasi pemesanan dari PT. AIM Purwokerto.

Tabel 2. 1 Penelitian Terdahulu

KATEGORI	PENELITIAN SEBELUMNYA					PENULIS
PENULIS TERDAHULU	Dedi Wahyuman (2022)	Widiarini, Fajar Agustini, Ristyanto Siswo Widiyanto, Sari Hartini (2022)	Dini Hari Pertiwi (2017)	Ripta Septiyana (2021)	Irmawati Carolina, Arief Rusman (2019)	Yohanes Richardianto (2021)
JUDUL	PERANCANGAN SISTEM INVENTORY PADA PT. PALOH SINGKWANG STABAT BERBASIS WEB PHP DENGAN METODE <i>EXTREME PROGRAMMING</i>	Metode Agile Pada Pembuatan <i>Website Services and Sales Printer Ink</i>	IMPLEMENTASI METODE <i>EXTREME PROGRAMMING</i> (XP) PADA APLIKASI PERSEDIA STOK GAS 3KG DI PT DINDA KARYA SABPUTRI BERBASIS WEB	SISTEM INFORMASI PEMESANAN TIKET WISATA KETENGER ADVENTURE BERBASIS <i>WEBSITE</i> DENGAN METODE WATERFALL	Penerapan <i>Extreme programming</i> Pada Sistem Informasi Penjualan Pakaian Berbasis <i>Web</i> (Studi Kasus Toko ST Jaya) Irmawati	Perancangan sistem informasi pemesanan berbasis <i>website</i> dengan metode extreme pemrograman (XP) untuk meningkatkan mutu pelayanan pada PT AIM Purwokerto
METODE	<i>EXTREME PROGRAMMING</i>	<i>Agile</i>	<i>EXTREME PROGRAMMING</i>	<i>Waterfall</i>	<i>EXTREME PROGRAMMING</i>	<i>EXTREME PROGRAMMING</i>

KATEGORI	PENELITIAN SEBELUMNYA					PENULIS
OBJEK PENELITIAN	Pengelolaan barang	Pengelolaan data service dan penjualan tinta	Pengelolaan stok tabung gas 3 Kg	Pengelolaan Pemesanan Tiket dan Kegiatan pada wisata Ketenger <i>Adventure</i>	Pengelolaan informasi penjualan	Pengelolaan pemesanan, manajemen barang dan pelaporan
Hasil penelitian	sistem informasi yang menyajikan proses barang masuk dan barang keluar, serta laporan persediaan barang secara online, sehingga dapat di lihat oleh manajemen secara online	sistem pelayanan jasa service dan penjualan tinta printer berbasis web yang akan mempermudah pemilik dalam mengelola laporan dan pelanggan dalam melakukan pembelian atau yang akan memperbaiki printer tanpa harus datang ke toko langsung	Sistem untuk mengecek stok tabung masuk dan stok tabung keluar	sistem informasi yang dapat membantu pengunjung dalam melakukan pemesanan tiket, membantu admin dalam mengelola data pesanan, memudahkan tim dalam melihat jadwal kegiatan dan membantu manajer dalam melakukan pelaporan data. Kata	sistem informasi berbentuk aplikasi web yang dapat mengolah data pakaian termasuk pada proses penjualannya dan juga pada proses laporan.	Sistem informasi pemesanan barang, manajemen barang, dan pelaporan
Perbandingan	Hanya menyajikan manajemen barang dan pelaporan untuk internal	Membuat isitem penjadwalan dan pembelian. Juga pada metode yang digunakan	Hanya menyajikan manajemen barang untuk internal	Membuat sistem penjadwalan untuk kegiatan Ketenger <i>Adventure</i> dan metode yang digunakan	Hanya menyajikan manajemen barang untuk internal.	Membuat sistem pemesanan barang, manajemen barang dan pelaporan

2.2 Landasan Teori

Berikut ini penulis menyampaikan beberapa dasar teori yang terdapat dalam penelitian;

2.2.1 Sistem Informasi

Sistem informasi manajemen (SIM) adalah sekumpulan atau sekelompok proses untuk mengolah data, menganalisis dan menampilkan data sehingga memiliki makna dan berguna untuk kebutuhan pengambilan suatu keputusan. SIM juga sangat bermanfaat dalam mendukung manajemen dalam mengambil keputusan dalam mengendalikan kegiatan perusahaan. Tujuan utama dari SIM ini adalah mengumpulkan data dari berbagai sumber dan mengolah data tersebut sehingga menghasilkan suatu informasi untuk memudahkan pihak manajemen dalam meningkatkan kualitas keputusan yang diambil guna meningkatkan produktivitas dan profitabilitas perusahaan.[17]

2.2.2 Sistem Pemesanan

Booking/Pemesanan adalah kegiatan untuk memesan sebuah tempat sesuai waktu dan tanggal perjalanan yang di inginkan. *Online* adalah keadaan disaat seseorang terhubung kedalam suatu jaringan ataupun sistem yang lebih besar. *Online booking* adalah bagian kegiatan dari seseorang yang melakukan pemesanan atau reservasi terhadap suatu produk atau jasa melalui media *online*. Industri pariwisata telah mencapai keunggulan kompetitif dari sektor lain, karena telah mampu mentransfer lebih banyak pelanggan secara *online*. [16]

2.2.3 Web

Web adalah Sebuah *software* yang berfungsi untuk menampilkan dokumen dokumen pada suatu *web* yang membuat pengguna dapat mengakses internet melalui *software* yang terkoneksi dengan internet.[9]

2.2.4 PHP

Hypertext Preprocessor atau yang bias akita kenal dengan PHP merupakan script yang terintegrasi dengan HTML dan berada pada server (*server side HTML*

embedded scripting). PHP adalah script yang digunakan untuk membuat halaman *website* yang dinamis. Dinamis berarti halaman yang akan ditampilkan dibuat saat halaman itu diminta oleh *client*. [4] PHP memiliki 2 jenis bahasa pemrograman yaitu PHP Natif dan PHP *Framework*. Berikut ini merupakan kelebihan dan kekurangan dari PHP Natif dan *Framework* :

- a. PHP Natif merupakan bahasa pemrograman PHP dimana *developer* harus membuat kerangka kerjanya sendiri dalam membangun sebuah *website* yang dinamis, PHP Natif ini biasanya digunakan oleh para pemrograman junior sebagai basic belajar *framework*. Berikut ini beberapa kelebihan dan kekurangan pada PHP Natif.

Tabel 2. 2 kelebihan dan kekurangan PHP Natif

NO	Kelebihan	Kekurangan
1	Pemrograman yang dibangun atas dasar pemikiran seorang programmer itu sendiri. Sehingga menerbitkan suatu program yang akan membuat program akan lebih baik.	Biasanya tidak cocok untuk proses bisnis ataupun sistem yang sangat luas. Karena <i>framework</i> cukup rendah untuk suatu bisnis yang terlalu luas.
2	Bisa dibentuk dalam sebuah <i>format</i> OOP (Object Oriented Programming) sehingga dapat membentuk suatu program.	Tidak ada Security Concern <i>framework</i> , dan tidak ada security/pengamanan default pada sistem yang sudah dibangun. Alias kita harus membangun sendiri pengamanannya. Supaya terhindar dari kesalahan-kesalahan dalam hal apapun.
3	Kita bisa menerapkannya dalam skala mudah sampai kita tau tergantung tingkat kesulitannya.	Tidak ada <i>Coding</i> Style Consistence, apabila sistem kita masih dikelola oleh programmer lainnya maka akan ada perbedaan dalam penulisan source code program

- b. *framework* merupakan bahasa pemrograman dimana *developer* tidak harus membuat kerangka kerjanya sendiri dalam membangun sebuah *website* akan tetapi *developer* harus mengetahui konsep yang diterapkan dalam *framework* tersebut. Berikut ini beberapa kelebihan dan kekurangan pada *Framework*.

Tabel 2. 3 Kelebihan dan Kekurangan *Framework*

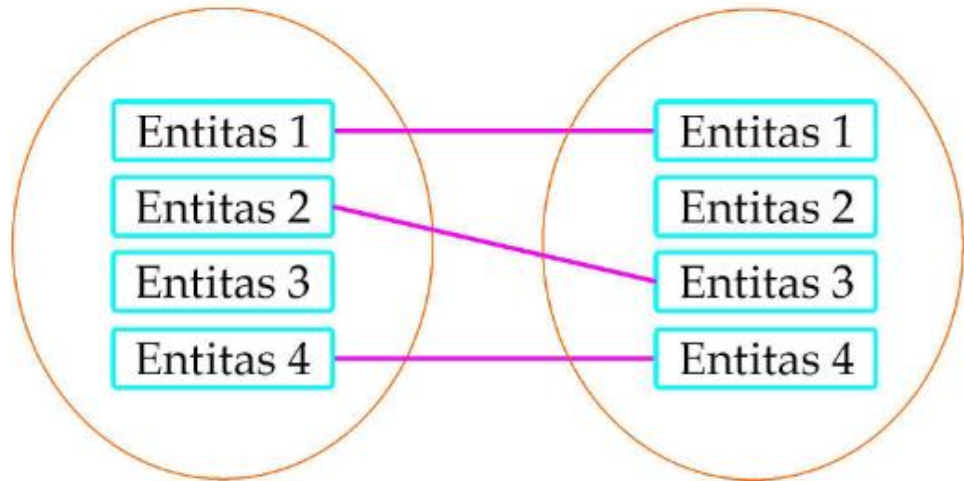
NO	Kelebihan	Kekurangan
1	Dapat membantu <i>developer/programmer</i> dalam membangun sebuah aplikasi	Programmer mungkin akan menemukan suatu kemungkinan dari batasan ketika merancang aplikasi menggunakan <i>framework</i> , contoh: belum tentu fungsi di <i>Framework A</i> ada di <i>Framework B</i>
2	Penerapan Design Patterns untuk memudahkan dalam rancangan, pengembangan dan pemeliharaan suatu sistem.	Kemungkinan juga biasanya akan ada penambahan biaya apabila pihak development kurang mendokumentasikan dan kurang disupport
3	<i>Coding Style Consistence</i> , juga akan memudahkan kita dalam membaca tentang source code karena cara pengkodean yang sama atau konsisten.	
4	Security Concern <i>Framework</i> , juga bisa mengantisipasi dalam memasang perisai terhadap adanya suatu masalah.	
5	<i>Framework</i> dapat mendisiplinkan kita untuk menulis suatu dokumentasi untuk sistem apa yang akan kita bangun. Supaya kita dapat membuat atau mempelajarinya sendiri.	

2.2.5 MySQL

MySQL adalah sebuah *database*. *Database* merupakan sebuah tempat untuk menyimpan data yang jenisnya beraneka ragam. *MySQL* merupakan tipe data relasional yang artinya *MySQL* menyimpan datanya dalam bentuk tabel-tabel yang saling berhubungan.[10] pada *MySQL* terdapat 4 relasi yaitu:

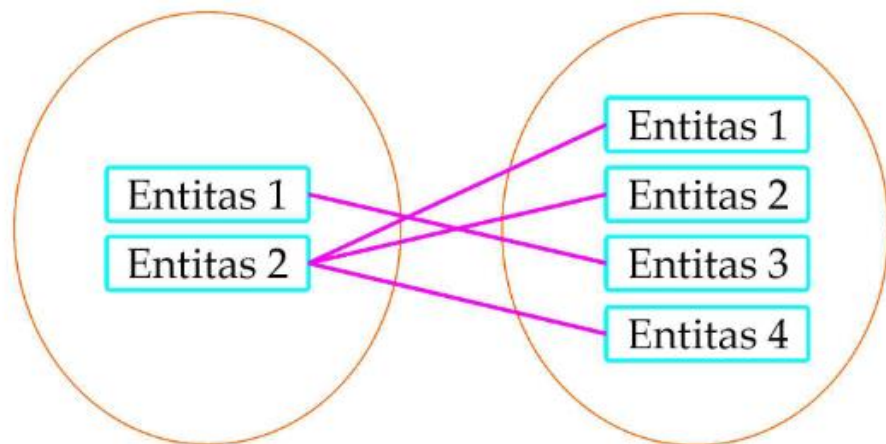
- a. satu ke Satu (*One to One*), yang berarti satu entitas A berhubungan paling banyak satu dengan entitas B. Begitupun sebaliknya, satu entitas

pada himpunan B berhubungan paling banyak satu dengan entitas A. [11]



Gambar 2. 1 Relasi One to One

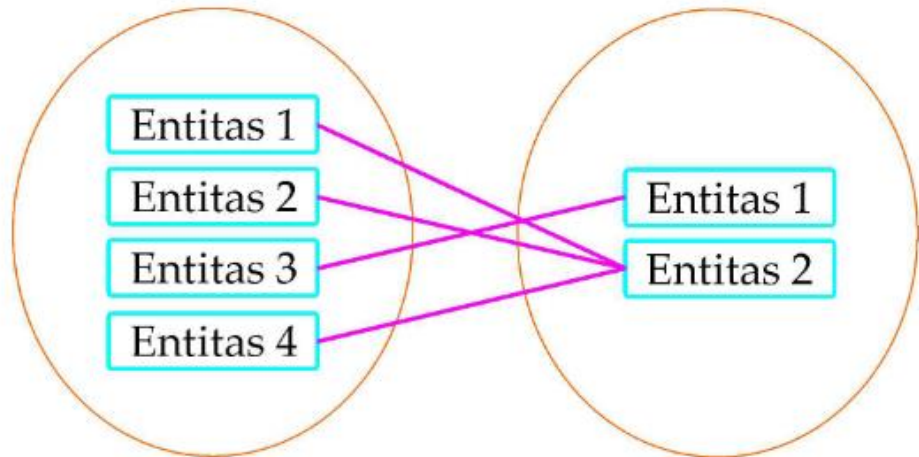
- b. Satu ke Banyak (*One to Many*), yang berarti satu entitas A dapat berhubungan lebih dari satu dengan entitas B. Namun tidak berlaku sebaliknya, dimana setiap entitas pada himpunan entitas B berhubungan dengan paling banyak dengan satu entitas pada himpunan entitas A. [11]



Gambar 2. 2 Relasi One to Many

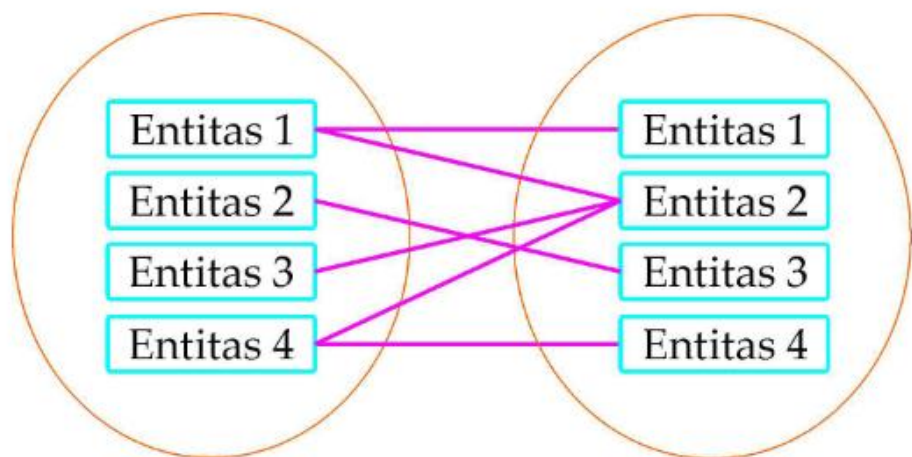
- c. Banyak ke Satu (*Many to One*), yang berarti satu entitas A berhubungan paling banyak dengan satu entitas B. Namun tidak berlaku sebaliknya,

dimana setiap entitas pada himpunan entitas B dapat berhubungan lebih dari satu dengan entitas A.[11]



Gambar 2. 3 Relasi Many to One

- d. Banyak ke Banyak (*Many to Many*), yang berarti satu entitas A dapat berhubungan lebih dari satu dengan entitas B dan sebaliknya.[11]



Gambar 2. 4 Relasi many to many

2.2.6 *CodeIgniter*

CodeIgniter merupakan *framework* PHP yang dibuat berdasarkan model view Controller (MVC). *CI* memiliki library yang lengkap untuk mengerjakan operasi-operasi yang umum dibutuhkan oleh aplikasi berbasis *web* misalnya

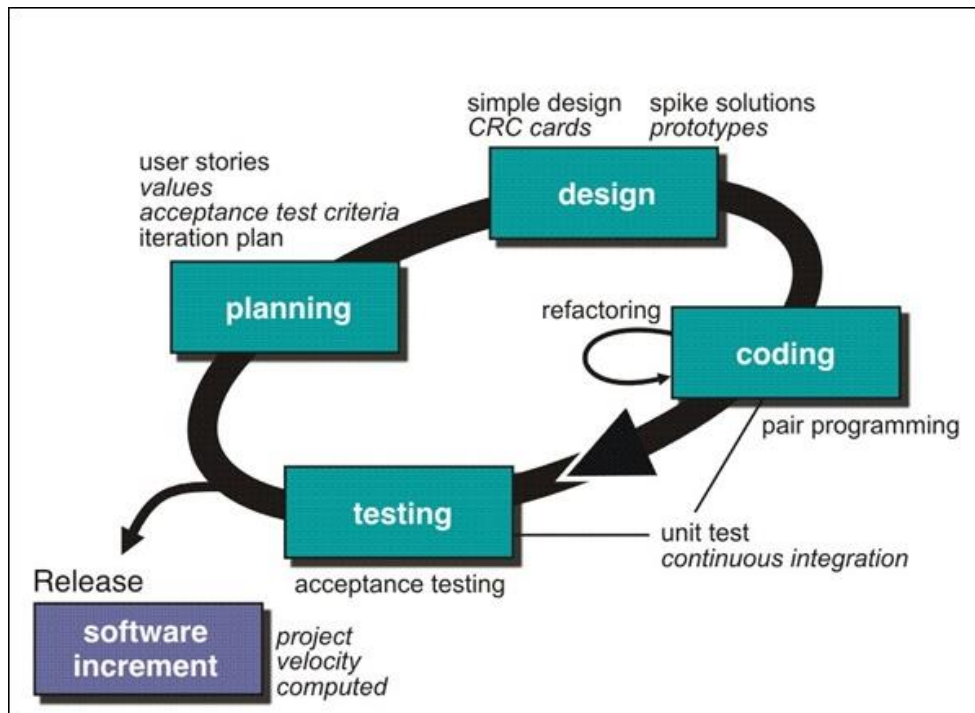
mengakses *database*, memvalidasi *form* sehingga sistem yang dikembangkan mudah.[9]

2.2.7 AdminLTE

AdminLTE adalah salah satu template yang sering digunakan oleh *web developer* sebagai template backend pada proyek yang sering dikerjakan. Jadi *adminLTE* ini adalah sebuah dashboard Administrator dibuat menggunakan bootstrap yang merupakan *framework* css yang paling banyak digunakan.[15]

2.2.8 Extreme Programming

Metode *Extreme programming* ini bersifat ringan, fleksibel dan berisiko rendah dengan kemampuan untuk mengelola requirement yang tidak jelas atau berubah dengan cepat, sehingga cocok untuk digunakan dalam tim dengan skala kecil dan menengah. Pada metode *Extreme programming* ini terdapat fitur yang membedakan dengan metode-metode lain yaitu *pair programming*. *Pair programming* merupakan teknik pengembangan perangkat lunak dimana dua orang program bekerja sama pada satu komputer untuk meminimalisir terjadinya *error* dan kesalahan pengkodean. Metode ini berfokus pada aspek teknik dibandingkan dengan manajemen proyek. *Extreme programming* sangat menekankan kepuasan pelanggan dengan menggunakan *feedback* pada saat pengembangan perangkat lunak tersebut sedang berjalan, sehingga dapat membantu meminimalisir kegagalan proyek dan menghasilkan perangkat lunak sesuai dari kebutuhan pelanggan.[12] Berikut ini adalah gambar dari tahapan-tahapan *EXTREME PROGRAMMING (XP)*.



Gambar 2. 5 Tahapan Extreme Programing

Berdasarkan gambar di atas, terdapat 4 fase dalam metode pengembangan *Extreme programming* yaitu:

- a. *Planning* yaitu Tahap *planning* dimulai dengan membuat user stories yang menggambarkan *output*, fitur, dan fungsi-fungsi dari *software* yang akan dibuat. *User stories* tersebut kemudian diberikan bobot seperti prioritas dan dikelompokkan untuk selanjutnya dilakukan proses *delivery* secara *incremental*.^[8] Terdapat beberapa aspek dalam perencanaan pengembangan sistem yaitu:

1. *User Story*

User story digunakan sebagai dokumen yang berisi kebutuhan klien dan akan digunakan oleh pengembang untuk membangun perangkat lunak dan juga untuk menentukan waktu perangkat lunak akan rilis. *User story* nantinya akan dipakai dalam *acceptance tests*. Jika pengembang dibatasi dalam mengembangkan antarmuka, *User story*

dapat dipakai sebagai skenario penggunaan perangkat lunak tersebut. *User story* seharusnya dibuat dengan detail yang cukup. Pengembang/*developer* menentukan perkiraan waktu yang dibutuhkan untuk mengimplementasikan satu *story* ke dalam code. Idealnya, setiap *story* membutuhkan 1-3 minggu (Ideal Development Time) jika *developer* tau apa yang harus dilakukan, tidak ada halangan, dan tidak ada tugas lain. Jika lebih dari tiga minggu, maka *story* harus dipecah menjadi *story* yang lebih detail. Dan jika kurang dari seminggu berarti *story* tersebut terlalu detail, maka sebaiknya digabung dengan *story* lainnya. Idealnya terdapat sekitar 80 *user story*. *Developer* sebaiknya menghindari rincian spesifik tentang teknologi yang digunakan, susunan basis data, dan algoritma. *Developer* seharusnya menjaga *User story* tetap fokus pada kebutuhan pengguna.

2. *Release Planning Creates The Release Schedule*

Setelah *User story* ditulis, *developer* dapat menggunakan *release planning meeting* untuk membuat sebuah *release plan*. Konsumer memilih *User story* yang akan dikerjakan / diimplementasikan ke dalam code pada iterasi berikutnya. Saat iterasi, *story-story* tersebut juga akan diterjemahkan ke dalam *acceptance tests*. *Acceptance tests* dilakukan saat iterasi untuk memverifikasi bila *story* sudah berhasil berjalan dengan baik, sehingga proses pun akan berjalan dengan baik. dengan begitu, waktu rilis akan terjadwal dengan.

3. *Make Frequent Small Releases*

Developer perlu merilis versi iterasi dari sistem yang telah dibuat dalam proses iterasi agar klien mengetahui proses yang telah dicapai oleh *developer*. Pada akhir iterasi, *developer* melakukan pengujian sehingga hasil tersebut kemudian dapat ditunjukkan kepada klien. Selanjutnya klien akan memutuskan apakah fungsi/sistem tersebut akan masuk produksi atau tidak. Dengan begitu, *developer* akan

mendapatkan *feedback* dari klien tanpa harus menunggu lama. Semakin lama menunggu untuk menunjukkan fitur penting kepada klien, maka semakin sedikit pula waktu untuk memperbaikinya.

4. *The project is divided into iterations*

Developer harus membagi proyek mejadi beberapa iterasi. Ini adalah hal yang sang penting dari proyek. *Developer* harus mengevaluasi proses kerja saat memasuki iterasi, jika diperkirakan tidak dapat menyelesaikan tugas ini, yang harus dilakukan adalah melakukan *iteration planning meeting*, merencanakan kembali tugas-tugas yang harus dikerjakan dalam iterasi, memperkirakan kembali, serta mengurangi beberapa tugas. Konsentrasikan pada usaha dalam menyelesaikan tugas yang dipilih oleh klien.

5. *Iteration planning starts each iteration.*

Perencanaan iterasi (*iteration planning*) dilakukan di awal iterasi untuk menentukan apa yang harus dikerjakan dalam iterasi ini. Klien memilih *user story* yang akan dikerjakan ataupun meminta untuk menyelesaikan *story* yang belum lolos *acceptance tests*. Hal ini dilakukan agar dapat memperkirakan kapan iterasi tersebut selesai dikerjakan.

- b. *Design* Pada tahap ini peneliti membuat tampilan *use case diagram*, *activity diagram*, *squence diagram* tentang aplikasi mobile berbasis android, mulai dari menu, penambahan alternatif, dan lainnya, Lalu peneliti membuat algoritma program tersebut. Pada perancangan sistem nya menggunakan UML (Unified Modelling Language).[13] Terdapat beberapa aspek dalam design pengembangan sistem yaitu:

1. *Simplicity*

Desain terlebih dahulu dibuat sesederhana mungkin (tidak menambahkan suatu fungsi) agar lebih cepat terselesaikan (Keep It Simple). Fungsi baru ditambahkan saat benar benar dibutuhkan.

2. *System Metaphor*

System metaphor merupakan bentuk komunikasi dimana kedua pihak (pengembang dan klien) memahami proses *design* perangkat lunak yang akan dibangun. Jadi, pengembang dipandu oleh klien dalam mendesain perangkat lunak. *System metaphor* akan menentukan mental model yang dimiliki anggota tim pengembang, dan akan membentuk *logical architecture* untuk sistem yang akan dibuat

3. *CRC Cards*

Class, Responsibilities, and Collaboration (CRC) cards digunakan tim pengembang mengumpulkan ide (brainstorming) dalam membuat desain perangkat lunak. *CRC card* biasanya digunakan saat pertama kali menentukan class apa yang diperlukan. *CRC cards* digunakan agar setiap anggota tim dapat berpartisipasi dalam pembuatan desain. Semakin banyak yang berpartisipasi, maka semakin banyak pula ide bagus yang didapat.

4. *Spike Solution*

Tujuan dari membuat *Spike Solution* adalah untuk memecahkan masalah teknis atau masalah dalam mendesain. Saat menemui masalah teknis, dapat menempatkan dua *developer* untuk menyelesaikan masalah itu untuk mengurangi resiko kegagalan.

5. *Never Add Functionality Early*

Seseorang biasanya berfikir untuk segera menambahkan fungsi-fungsi tambahan yang mungkin dikarenakan mudahnya fungsi tersebut atau memperkirakan fungsi tersebut akan berguna kelak. Seorang programmer harus mengingatkan dirinya sendiri untuk tidak terburu-buru menambahkan fungsi tambahan karena itu akan membuang waktu. Program dibuat siap untuk ditambahkan fungsi-fungsi setelah program tersebut selesai.

6. *Refactoring*

mengizinkan programmer melakukan perbaikan-perbaikan pada kode program walaupun hal tersebut tidak direncanakan dalam desain. Perbaikan pada proses *refactoring* pada umumnya ditujukan untuk

optimalisasi kualitas kode program (contoh: modularitas, efisiensi, reusability, kompleksitas, dan maintainability) dengan persyaratan utama tidak mengubah konsistensi output dan proses dari perangkat lunak tersebut

- c. *Coding* proses *coding* pada XP diawali dengan membangun serangkaian unit *test*. Setelah itu pengembang akan berfokus untuk mengimplementasikannya. Dalam *Extreme programming* diperkenalkan istilah Pair Programming dimana proses penulisan program dilakukan secara berpasangan. Dua orang programmer saling bekerjasama di satu komputer untuk menulis program. Dengan melakukan ini akan didapat real-time problem solving dan real-time quality assurance.[8] Terdapat beberapa aspek dalam *coding* pengembangan sistem yaitu:

1. *Code must be written to agreed standards*

Pada metode *extreme programming* sebelum melakukan proses coding. Langkah pertama adalah membuat sebuah unit *test* terlebih dahulu. Sehingga proses pengerjaan menjadi lebih cepat dan mudah. Unit *test* berfungsi untuk menjelaskan apa yang harus dibuat. Setelah unit tes selesai maka proses selanjutnya adalah membuat *code*. Lalu dibuat unit *test* selanjutnya baru membuat codenya.

2. *Only one pair integrates code at a time*

Pada metode *extreme programming* untuk menggabungkan code yang di buat oleh berbagai tim dibentuk satu pasang tim khusus. Tim tersebut diharapkan dapat menyelesaikan permasalahan saat proses penggabungan code, karena kemungkinan terjadinya ketidak cocok kode yang telah dibuat oleh tim lainnya. Pengembang perangkat lunak sebaiknya melakukan penggabungan code dari banyak tim sesering mungkin. Hal tersebut diharapkan mengurangi kesalahan sejak awal sehingga masih cepat/mudah untuk diperbaiki. Memperbaiki kesalahan sejak awal penggabungan code berbeda ketika code sudah banyak lalu digabungkan.

3. *Set up a dedicated integration computer*

Menggunakan satu komputer yang digunakan untuk mengontrol hasil rilis dari pengembang program. Jika program yang telah rilis tidak mengalami kesalahan dan sesuai dengan keinginan pelanggan, maka program akan disimpan, namun jika masih ada kekurangan maka program akan dikembalikan lagi.

4. *Use collective ownership*

Seluruh tim pengembang mempunyai hak untuk memberikan kontribusi terhadap proyek yang sedang dikerjakan. Hal tersebut diharapkan dapat meningkatkan kualitas perangkat lunak yang dikembangkan. Pada metode *extreme programming* tidak ada anggapan bahwa terdapat salah seorang yang menjadi kepala arsitek yang mengatur proyek secara keseluruhan.

- d. *Testing* Tahapan ini merupakan tahapan pengujian terhadap aplikasi yang sudah dibangun, pada tahapan ini ditentukan oleh pengguna sistem dan berfokus pada fitur dan fungsionalitas dari keseluruhan sistem kemudian ditinjau oleh pengguna sistem.[14] Terdapat beberapa aspek dalam *testing* pengembangan sistem yaitu:

1. *All code must pass all the unit test*

Kode yang dibuat oleh developer harus memiliki unit *test*. Unit *test* adalah pengujian yang dilakukan oleh *developer* dengan cara membagi-bagi kode menjadi bagian-bagian tertentu dan sehingga dapat dengan mudah memperbaiki program jika terjadi kesalahan atau ketika terjadi perubahan permintaan dari pelanggan. Program yang akan dirilis adalah program yang secara keseluruhan sudah lulus melalui tahap pengujian.

2. *When a bug is found*

Saat bug ditemukan, maka programmer fokus pada *bug* tersebut dan memperbaiki code. Sehingga perlu melakukan unit *test* kembali. Hal tersebut diulang kembali jika code masih belum benar.

3. *Acceptance test*

Pengujian dilakukan berdasarkan permintaan dari *customer*. Apakah program sesuai dengan apa yang diminta oleh *customer*. Jika masih terdapat bug atau ketidaksesuaian dengan permintaan *customer* maka proses akan kembali ke tahap *planning* / perencanaan kembali.

4. *All production code is pair programmed*

Cara kerja pair programming adalah dua orang programmer bekerja bersama memecahkan sebuah masalah dalam 1 komputer. Sehingga menghasilkan code yang lebih berkualitas karena hasil kombinasi 2 pemikiran yang mengerjakan tugas bersama-sama.

- e. *Software Increment* (Peningkatan Perangkat Lunak) Tahapan ini merupakan tahap pengembangan sistem yang sudah dibuat secara bertahap yang dilakukan setelah sistem diterapkan dalam organisasi dengan menambahkan layanan atau konten yang mengakibatkan bertambahnya kemampuan fungsionalitas dari sistem.[14]





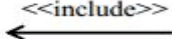

2.2.9 Black Box Testing

Pengujian *black box* merupakan teknik pengujian yang mengabaikan mekanisme internal atau struktur suatu sistem. Pengujian *black box* berfokus pada persyaratan fungsionalitas perangkat lunak. Pengujian *black box* dilakukan untuk mengevaluasi suatu sistem dengan persyaratan fungsional yang ditentukan dan hasil prediksi yang sesuai. Pengujian *black box* juga melengkapi teknik pengujian *white box* dan cenderung mengungkap kesalahan yang berbeda dengan metode *white box*. [12]

2.2.10 Use case Diagram

Diagram *use case* merupakan pemodelan untuk kelakuan (behavior) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. [18]

Use case secara sederhana sesungguhnya merupakan sebuah sarana bantu untuk mendefinisikan apa yang ada di luar sistem (aktor) dan apa yang harus dilakukan oleh sistem yang sedang dikembangkan.[19] Berikut ini adalah symbol-symbol pada use case, seperti terlihat pada gambar berikut.







Simbol	Keterangan
	Aktor : Mewakili peran orang, sistem yang lain, atau alat ketika berkomunikasi dengan <i>use case</i>
	<i>Use case</i> : Abstraksi dan interaksi antara sistem dan aktor
	<i>Association</i> : Abstraksi dari penghubung antara aktor dengan use case
	<i>Generalisasi</i> : Menunjukkan spesialisasi aktor untuk dapat berpartisipasi dengan use case
	Menunjukkan bahwa suatu use case seluruhnya merupakan fungsionalitas dari use case lainnya
	Menunjukkan bahwa suatu use case merupakan tambahan fungsional dari use case lainnya jika suatu kondisi terpenuhi

Gambar 2. 6 Simbol pada *Use Case*

2.2.11 Activity Diagram

Aktivitas merupakan kumpulan aksi- aksi. Aksi-aksi melakukan langkah sekali saja tidak boleh dipecah menjadi beberapa langkah lagi. Sebagai contoh fungsi matematika, pemanggilan perilaku, pemrosesan data. Aktivitas dapat mengakses atribut dan operasi dan operasi classifier, tiap objek yang terhubung dan parameter-parameter jika aktivitas memiliki hubungan dengan perilaku. Ketika digunakan untuk model proses bisnis, informasi itu biasanya disebut process-relevant data.[20] *Activity diagram* merupakan diagram ini menggambarkan tentang aktifitas yang terjadi pada sistem. Dari pertama sampai akhir, diagram ini

menunjukkan langkah – langkah dalam proses kerja sistem yang kita buat. 19] Berikut ini adalah symbol-simbol pada activity diagram, seperti terlihat pada gambar berikut.




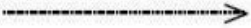

Nama	Simbol	Fungsi
Object		Menggambarkan sebuah <i>class</i> atau <i>object</i> .
Activation boxes		Menggambarkan panjang waktu yang dibutuhkan sebuah <i>object</i> dalam mengerjakan tugasnya
Actors		Menggambarkan pengguna yang berinteraksi dengan sistem
Lifeline		Menggambarkan “garis hidup” sebuah <i>object</i>
Message		Menggambarkan pesan atau interaksi antar <i>object</i>
Message to Self		Menggambarkan pesan balikan atau reaksi dari <i>object</i> sebelumnya

Gambar 2. 7 Simbol Activity Diagram

2.2.12 Class Diagram

Class Diagram bersifat statis. Diagram ini memperlihatkan himpunan kelas-kelas, antarmuka-antarmuka, kolaborasi-kolaborasi, serta relasi-relasi. Diagram ini umum dijumpai pada pemodelan sistem berorientasi objek. Meskipun bersifat statis, sering pula diagram kelas memuat kelas-kelas aktif. Penjelasan.[20]

Diagram kelas atau class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas.[19] Berikut ini adalah symbol-simbol pada *class diagram*, seperti terlihat pada gambar berikut.

asosiasi / <i>association</i> 	relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
asosiasi berarah / <i>directed association</i> 	relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
generalisasi 	relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)
kebergantungan / <i>dependency</i> 	relasi antar kelas dengan makna kebergantungan antar kelas
agregasi / <i>aggregation</i> 	relasi antar kelas dengan makna semua-bagian (<i>whole-part</i>)

Gambar 2. 8 Simbol Class Diagram