

BAB 2

TINJAUAN PUSTAKA

Pada bab tinjauan pustaka menjelaskan tentang teori-teori terkait penelitian yang bersumber dari buku, jurnal, atau artikel ilmiah. Tujuannya adalah untuk memahami konsep dan teori penyelesaian permasalahan yang digunakan. Teori yang terdapat pada bab ini adalah tinjauan perusahaan CV. Indrajaya, sistem informasi, *Unified Model Language (UML)*, *Entity Relationship Diagram (ERD)*, *System Development Life Cycle (SDLC)*, *CodeIgniter*, *Blackbox Testing*, dan penelitian terdahulu.

2.1 CV. Indrajaya

CV. Indrajaya adalah perusahaan yang bergerak di bidang otomotif yaitu bengkel dengan bidang pelayanan khusus. CV. Indrajaya berlokasi di Jl. Mayor TNI A.D. Imat Saili RT. 56, No. 25 Kelurahan Sumber Rejo, Kecamatan Balikpapan Tengah, Kota Balikpapan. CV. Indrajaya telah beroperasi sejak tahun 2000 dan didirikan pertama kali oleh bapak Indro Cahyono. CV. Indrajaya memiliki spesialisasi dalam segi perawatan dan reparasi elemen-elemen yang ada di mobil. Pelayanan yang diberikan oleh CV. Indrajaya adalah *body repair* (reparasi mobil) pada kendaraan mobil yang baik dan lengkap yang ada di kota Balikpapan. Pengerjaan *body repair* kendaraan mobil di CV. Indrajaya prosesnya sama seperti merakit kendaraan dari awal. Para teknisi menganalisis kerusakan pada mobil terlebih dahulu, kemudian melepas panel, memperbaiki elemen yang rusak, mengecat *body* mobil, hingga tahap akhir atau *finishing*. Semua tahap pengerjaan reparasi mobil di CV. Indrajaya melewati proses kontrol yang tinggi dengan tenaga ahli khusus dalam memperbaiki mobil rusak. Saat ini, CV. Indrajaya memiliki 15 karyawan yang terdiri dari 11 orang teknisi, 2 orang admin, dan 2 orang kasir.

Sebagai salah satu bengkel reparasi yang selalu menekankan kualitas dengan cara berkomitmen untuk memberi pelayanan yang terbaik kepada pelanggan. Komitmen tersebut dicerminkan oleh visi dan misi CV. Indrajaya agar menjadi

bengkel reparasi yang amanah dan terpercaya bagi pelanggan. Adapun visi dan misi CV. Indrajaya adalah sebagai berikut:

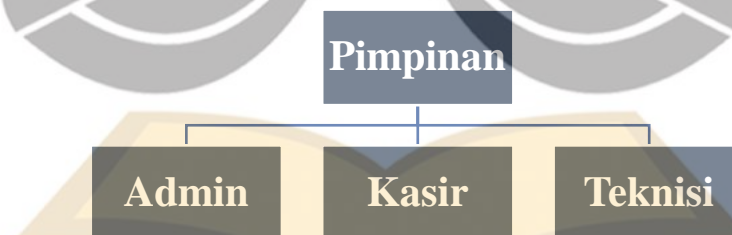
1. Visi

Menjadi pusat reparasi mobil yang menyediakan *panel body* lengkap di Kota Balikpapan, jasa pelayanan yang optimal dan terpercaya, serta selalu mengutamakan kepuasan pelanggan didukung dengan Sumber Daya Manusia (SDM) yang ahli dan kompeten di bidangnya.

2. Misi

- a. Memberikan *panel body* yang terbaik, berkualitas, serta reparasi yang terpercaya.
- b. Memberikan pelayanan terbaik dengan standar mutu pada pelanggan dengan mengerjakan proses reparasi melalui tenaga ahli di bidangnya sehingga tercapai kepuasan pelanggan.
- c. Meningkatkan semangat kerja karyawan secara optimal melalui peningkatan kemampuan kinerja (*workshop*), disiplin, kejujuran, serta memberikan penghargaan yang memadai sesuai dengan kinerjanya.

CV. Indrajaya dipimpin langsung oleh pemilik yang bertanggungjawab untuk mengawasi dan mengevaluasi kinerja karyawan di CV. Indrajaya. Struktur organisasi dan uraian jabatan pada CV. Indrajaya dapat dilihat pada gambar 2.1 di bawah ini.



Gambar 2.1 Struktur Organisasi CV. Indrajaya

Fungsi dan uraian jabatan pada CV. Indrajaya dijelaskan di bawah ini sebagai berikut :

1. Pimpinan

Pimpinan memiliki tanggung jawab terhadap keseluruhan proses reparasi yang terjadi di bengkel, seperti merencanakan *indent* barang, melakukan pembelian

panel body kepada *supplier*, serta menerima laporan bulanan yang diberikan oleh admin CV. Indrajaya. www.itk.ac.id

2. Admin

Bagian admin memiliki tanggung jawab untuk mencatat administrasi yang ada di bengkel, melakukan pemeriksaan ketersediaan *panel body* mobil di gudang, serta mengelola laporan reparasi bulanan dari setiap transaksi reparasi yang terjadi di bengkel.

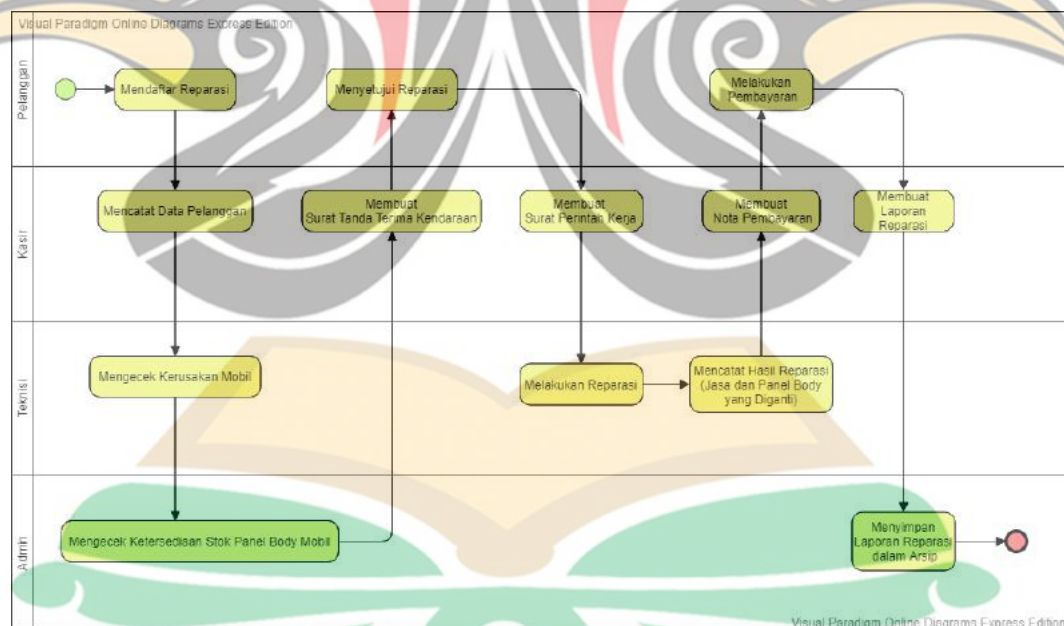
3. Kasir

Bagian kasir memiliki tanggung jawab untuk mencatat data pelanggan dan mencatat setiap transaksi reparasi yang terjadi di dalam bengkel CV. Indrajaya.

4. Teknisi

Bagian teknisi memiliki tanggung jawab untuk melakukan reparasi mobil pelanggan. Jika reparasi mobil pelanggan telah selesai dilakukan, teknisi akan melapor ke bagian kasir.

Adapun proses bisnis reparasi mobil di CV. Indrajaya dapat dilihat pada gambar 2.2 di bawah ini.



Gambar 2.2 Proses Bisnis Reparasi CV. Indrajaya

Di bawah ini akan dijelaskan mengenai alur proses bisnis reparasi mobil di CV. Indrajaya yaitu sebagai berikut :

1. Proses reparasi dimulai dengan pelanggan mendaftarkan mobil yang ingin direparasi di bagian kasir.

2. Kemudian, kasir akan meminta STNK Mobil untuk mencatat data mobil pada buku pelanggan. Kasir juga akan menuliskan kronologi kerusakan mobil berdasarkan keterangan dari pelanggan.
3. Setelah menuliskan kronologi kerusakan mobil, admin akan menghubungi salah satu teknisi untuk memeriksa kerusakan mobil apakah sudah sesuai dengan kronologi yang diceritakan oleh pelanggan.
4. Jika kerusakan mobil telah diperiksa, maka teknisi akan menghubungi admin untuk ketersediaan stok *panel body*. Jika *panel body* masih tersedia, maka proses perbaikan segera dilakukan.
5. Kemudian, kasir akan membuat Surat Tanda Terima Kendaraan untuk diberikan kepada pelanggan sebagai tanda kelengkapan mobil yang akan direparasi.
6. Teknisi akan mengkonfirmasi kerusakan mobil dan *panel body* yang harus diganti kepada pelanggan. Apabila pelanggan telah melakukan konfirmasi untuk menyetujui proses reparasi, maka teknisi akan membuat estimasi biaya reparasi mobil.
7. Kemudian, kasir akan membuat Surat Perintah Kerja (SPK) untuk teknisi. Setelah teknisi menerima SPK dari kasir, maka proses reparasi dijalankan hingga mobil selesai diperbaiki.
8. Jika mobil telah selesai direparasi, teknisi memberikan nota kepada bagian kasir dan kasir menghitung total biaya reparasi mobil.
9. Selanjutnya, kasir akan menghubungi pelanggan untuk melakukan proses pembayaran reparasi. Jika proses pembayaran telah selesai, kasir akan membuat laporan reparasi mobil.
10. Setelah itu, laporan diserahkan kepada admin untuk disimpan di dalam arsip.
- CV. Indrajaya akan melakukan pencatatan berapa jumlah mobil rusak yang masuk dalam per hari dan transaksi apa saja yang dilakukan. Catatan tersebut dapat digunakan untuk menjelaskan keuntungan yang diperoleh per hari dan *panel body* yang paling sering dibeli.

2.2 Sistem Informasi

Sistem informasi adalah suatu sistem yang terdapat di dalam suatu organisasi dan mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang sifatnya manajerial dengan kegiatan strategi dari suatu organisasi agar dapat menyediakan kepada pihak luar tertentu dengan laporan yang dibutuhkan (Sutabri, 2004).

Suatu sistem pasti memiliki tujuan ataupun sasaran yang ingin dicapai. Jika suatu sistem tidak memiliki tujuan, maka operasi dari sistem dikatakan tidak ada gunanya. Sasaran dari sistem sangat menentukan masukan yang dibutuhkan serta keluaran yang akan dihasilkan untuk dapat mencapai keberhasilan jika sistem memiliki tujuan (Hartono, 1990).

Komponen dari sistem informasi terdiri atas (Jogiyanto, 2008) :

1. *Software*, merupakan kumpulan dari perintah atau fungsi yang ditulis dengan aturan tertentu untuk memerintahkan komputer dalam melaksanakan tugas tertentu.
2. *Hardware*, merupakan perangkat keras yang terdiri dari komputer, *printer*, dan jaringan.
3. Prosedur, merupakan dokumentasi prosedur atau proses dari sistem yang berupa buku panduan operasional dan teknis.
4. Data, merupakan komponen dasar dari informasi yang diproses lebih lanjut untuk menghasilkan sebuah informasi.
5. Manusia, merupakan pengguna dari sistem informasi.

Kegiatan yang dilakukan dalam penerapan sistem informasi mencakup hal-hal sebagai berikut (Jogiyanto, 2008) :

1. *Input*, menggambarkan suatu proses/kegiatan untuk menyediakan data agar dapat diproses.
2. Proses, menggambarkan bagaimana suatu data diproses agar dapat menghasilkan suatu informasi yang bernilai lebih.
3. *Output*, suatu kegiatan yang menghasilkan laporan dari suatu proses.
4. Penyimpanan, suatu kegiatan untuk memelihara dan menyimpan data.

2.3 Unified Model Language (UML)

Unified Model Language (UML) adalah suatu metode permodelan secara visual dalam merancang suatu sistem yang berorientasi obyek dan menghasilkan dokumentasi dari sistem yang akan dibangun. Fungsi dari penggunaan UML adalah sebagai berikut (Haqi, 2019):

1. Dapat memberikan bahasa pemodelan visual kepada *user project* agar perancangan sistem mudah dipahami.
2. Dapat menyatukan *best practice* yang ada di dalam pemodelan.
3. Dapat digunakan sebagai *blue print* dari sistem yang dibuat karena dokumentasi yang lengkap dan detail dalam perancangan sistem.
4. Dapat memodelkan sistem yang berorientasi objek.
5. Dapat menciptakan bahasa pemodelan yang dapat digunakan oleh manusia ataupun mesin.

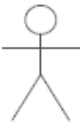
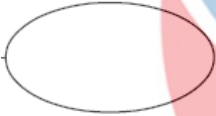


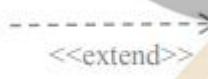

Jenis-jenis UML yang digunakan untuk mendetailkan sistem informasi yang dibangun pada penelitian ini adalah sebagai berikut :

2.3.1 Use Case Diagram

Use case diagram adalah pemodelan untuk menggambarkan kelakuan sistem/*behavior* yang akan dibangun. *Use case diagram* menggambarkan *external view* dari sistem yang akan dibuat pemodelannya. Model *use case* dapat dijabarkan dalam *use case diagram*, tetapi diagram yang digambarkan tidak identik dengan model karena model lebih luas daripada diagram. Adapun komponen pembentuk diagram *use case* adalah sebagai berikut (Muslihudin & Oktafianto, 2016):

1. Aktor (*actor*), menggambarkan pihak-pihak yang berperan dalam sistem.
 2. *Use Case*, aktivitas/sarana yang disiapkan oleh bisnis/sistem yang dibangun.
 3. Hubungan (*link*), aktor yang terlibat dalam *use case* yang dibuat.
- Simbol pada *use case* dapat dilihat dan dijelaskan pada tabel 2.1 di bawah ini (Ananda, 2018).

Tabel 2.1 Simbol *Use Case* (Ananda, 2018)

Simbol	Nama	Keterangan
	<i>Actor</i>	<i>Actor</i> adalah segala sesuatu yang berinteraksi langsung dengan sistem aplikasi komputer, seperti orang, benda, atau lainnya. Tugas <i>actor</i> adalah memberikan informasi kepada sistem dan dapat memerintahkan sistem agar melakukan suatu tugas.
	<i>Use Case</i>	<i>Use Case</i> digambarkan dalam bentuk lingkaran/ <i>elips</i> dengan nama <i>use case</i> di dalam <i>elips</i> tersebut.
	<i>Association Relationship</i>	Asosiasi digunakan untuk menghubungkan <i>actor</i> dengan <i>use case</i> .
	<i>Generalization Relationship</i>	<i>Generalization</i> menunjukkan hubungan antara elemen yang lebih umum ke elemen yang lebih spesifik.
	<i>Extend Relationship</i>	<i>Extend</i> menunjukkan bahwa suatu bagian dari elemen di garis panah bisa disisipkan ke dalam elemen yang ada di garis dengan panah.
	<i>Include Relationship</i>	<i>Include</i> menunjukkan suatu bagian dari elemen (yang ada di garis tanpa panah) dapat memicu eksekusi bagian dari elemen lain (yang ada di garis dengan panah).

2.3.2 Activity Diagram

Activity diagram menggambarkan aliran kerja/*workflow* atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Hal yang dapat diperhatikan pada *activity diagram* adalah diagram aktivitas menggambarkan aktivitas yang dapat dilakukan oleh sistem bukan yang dapat dilakukan oleh aktor (Salahuddin & Rosa, 2013). Simbol dan penjelasan tentang *activity diagram* dijelaskan pada tabel 2.2 di bawah ini (Ananda, 2018).

Tabel 2.2 Simbol *Activity Diagram* (Ananda, 2018)

Simbol	Nama	Keterangan
	Action	State dari sistem yang mencerminkan eksekusi dari suatu aksi.
	Start State	Bagaimana objek dibentuk atau dimulai.
	End State	Bagaimana objek dibentuk dan diselesaikan.
	Process	Pilihan untuk mengambil keputusan.
	Fork Node	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran.

2.3.3 Class Diagram

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, sedangkan operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas (Salahuddin & Rosa, 2013).

Class Diagram dibuat agar pengembang sistem dapat membuat kelas-kelas sesuai rancangan di dalam diagram kelas agar antara dokumentasi perancangan dan perangkat lunak sinkron. Simbol dan penjelasan *class diagram* dijelaskan pada tabel 2.3 di bawah ini (Ananda, 2018).

Tabel 2.3 Simbol *Class Diagram* (Ananda, 2018)






Simbol	Nama	Keterangan
	<i>Association</i>	Menghubungkan antara objek satu dengan objek lainnya.
	<i>Direct Association</i> (Asosiasi Berarah)	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain.
	Generalisasi	Relasi antar kelas dengan makna generalisasi dan spesialisasi (umum dan khusus).
	<i>Dependancy</i> (Kebergantungan)	Relasi antar kelas dengan makna kebergantungan antar kelas.
	<i>Aggregation</i> (Agregasi)	Relasi antar kelas dengan makna semua dan bagian (<i>whole-part</i>).

2.3.4 *Sequence Diagram*

Sequence Diagram menggambarkan kelakuan/*behavior* objek pada *use case* dengan cara mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu, dalam menggambarkan *sequence diagram* maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Pembuatan *sequence diagram* dibutuhkan agar dapat melihat skenario yang ada pada *use case* (Salahuddin & Rosa, 2013).

Sequence Diagram menggambarkan bagaimana objek berinteraksi satu sama lain melalui pesan pada eksekusi sebuah *use case* atau operasi. Diagram ini mengilustrasikan bagaimana pesan terkirim dan diterima di antara objek dan dalam *sequence* yang terlibat. Simbol dari *sequence diagram* dijelaskan pada tabel 2.4 di bawah ini (Ananda, 2018).

Tabel 2.4 Simbol *Sequence Diagram* (Ananda, 2018)

Simbol	Nama	Keterangan
	<i>Object</i>	<i>Object</i> merupakan <i>instance</i> dari sebuah <i>class</i> dan dituliskan tersusun secara horizontal. Digambarkan sebagai sebuah <i>class</i> berbentuk kotak dengan nama obyek didalamnya yang diawali dengan sebuah titik koma.
	<i>Actor</i>	<i>Actor</i> dapat berkomunikasi atau berinteraksi dengan sistem.
	<i>Lifeline</i>	<i>Lifeline</i> mengindikasikan keberadaan sebuah object dalam basis waktu. Notasi untuk <i>Lifeline</i> adalah garis putus-putus vertical yang ditarik dari sebuah obyek.
	<i>Activation</i>	<i>Activation</i> dinotasikan sebagai sebuah kotak segi empat yang digambar pada sebuah <i>lifeline</i> . <i>Activation</i> mengindikasikan sebuah objek yang akan melakukan sebuah aksi.
	<i>Message</i>	<i>Message</i> digambarkan dengan anak panah horizontal antara <i>Activation</i> . <i>Message</i> mengindikasikan komunikasi antara <i>object</i> .

2.4 Entity Relationship Diagram

Entity Relationship Diagram (ERD) adalah gambaran dari sistem yang memiliki hubungan antara *entity* beserta relasinya. *Entity* (entitas) merupakan sesuatu yang ada dan terdefiniskan di dalam suatu organisasi. *Entity Relationship*

Diagram (ERD) dibutuhkan agar dapat menggambarkan hubungan antara entitas dengan jelas, dapat menggambarkan batasan jumlah entitas, dan keterlibatan antar entitas, mudah dimengerti oleh pengguna dan mudah disajikan oleh perancang *database*. Oleh karena itu, ERD dibagi menjadi dua jenis model, yaitu sebagai berikut (Trong, 2017) :

a. *Conceptual Data Model*

Conceptual Data Model (CDM) adalah jenis model data yang menggambarkan hubungan antar tabel secara konseptual.

b. *Physical Data Model*

Physical Data Model (PDM) adalah jenis model data yang menggambarkan relasi antara tabel secara *physical*.

2.5 System Development Life Cycle (SDLC)

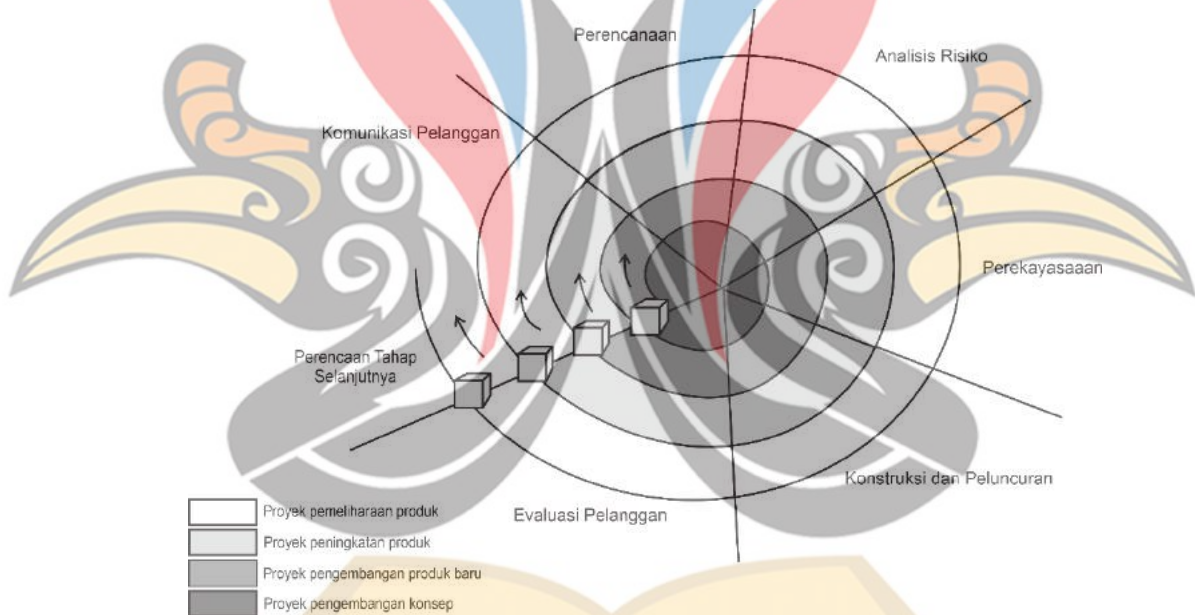
System Development Life Cycle (SDLC) atau siklus hidup pengembangan sistem dalam rekayasa sistem dan rekayasa perangkat lunak adalah proses pembuatan dan perubahan sistem serta model dan metodologi yang digunakan untuk mengembangkan sistem. SDLC merupakan pola yang diambil untuk mengembangkan sistem perangkat lunak, yang terdiri dari tahap-tahap : perencanaan (*planning*), analisis (*analysis*), desain (*design*), implementasi (*implementation*), uji coba (*testing*), dan pengelolaan (*maintenance*). Dalam rekayasa perangkat lunak, SDLC mendasari berbagai jenis metodologi pengembangan perangkat lunak. Metodologi ini membentuk suatu kerangka kerja untuk tahap perencanaan dan pengendalian pembuatan suatu sistem informasi (Susanto & Andriana, 2016). Terdapat beberapa jenis *System Development Life Cycle* (SDLC) yaitu : *Agile SDLC*, *Waterfall SDLC*, *Scrum SDLC*, *Iterative SDLC*, *Spiral SDLC*, *V SDLC*, *Big Bang SDLC*, *Rational Unified Process (RUP) SDLC*, *Prototype SDLC*, *Rapid Application Development (RAD) SDLC*, dan *Unified Process SDLC* (Munthe, 2017). Pada penelitian ini, digunakan metode *Spiral* untuk melakukan pengembangan sistem dan dijelaskan pada sub bab di bawah ini.

2.5.1 *Spiral Model*

Spiral model adalah proses model dari perangkat lunak yang memadukan wujud pengulangan dari model *prototyping* dengan aspek pengendalian dan

sistematika dari *waterfall model*. Dalam model *spiral*, perangkat lunak dikembangkan dalam *incremental release* (Nugroho dkk., 2009). Model *spiral* menyediakan kemungkinan untuk pengembangan perangkat lunak yang cepat, yang bergerak dari versi perangkat lunak yang kurang lengkap menjadi versi yang sangat lengkap (Pressman, 2002).

Bentuk *spiral* memberikan gambaran bahwa semakin besar iterasi yang dilakukan, maka akan menunjukkan versi lengkap dari perangkat lunak yang akan dibangun. Dalam pengembangan sistem informasi berbasis web, model ini digunakan untuk menyelesaikan sistem secara global terlebih dahulu, lalu fitur dari sistem dikembangkan setelahnya. Model ini dapat mempercepat penerapan proyek dan cocok digunakan dalam sistem informasi berbasis *web* (Sopian dkk, 2013). Model *spiral* digambarkan pada Gambar 2.3 di bawah ini.



Gambar 2.3 Model *spiral* (Pressman, 2002)

Pada gambar 2.3 adalah proses dari setiap untaian model *spiral* yang setiap *loop* mewakili satu tahap dari *software process*. *Loop* paling dalam berfokus pada kelayakan dari sistem, *loop* selanjutnya tentang definisi kebutuhan pelanggan, *loop* berikutnya berkaitan dengan desain sistem dan seterusnya. Adapun penjelasan pada tahapan model *spiral* dijelaskan di bawah ini (Pressman, 2002).

1. Komunikasi Pelanggan

Tugas-tugas yang diperlukan dalam membangun komunikasi yang efektif diantara pengembang dan pelanggan.

2. Perencanaan

Tugas-tugas yang dibutuhkan untuk mendefinisikan sumber daya yang dibutuhkan dalam membangun sistem, ketepatan waktu, serta proyek informasi yang berhubungan dengan pembangunan sistem.

3. Analisis Risiko

Tugas-tugas yang dibutuhkan untuk melakukan analisis dan identifikasi risiko yang terjadi baik secara teknis maupun manajemen.

4. Perencanaan

Tugas-tugas yang dibutuhkan dalam membangun satu atau lebih representasi dari sistem yang akan dibangun.

5. Konstruksi dan Peluncuran

Tugas-tugas yang dibutuhkan untuk melakukan konstruksi, menguji, memasang, dan memberikan pelayanan kepada pemakai. Misalnya adalah dengan melakukan *user training* dan dokumentasi.

6. Evaluasi Pelanggan

Tugas-tugas yang dibutuhkan agar dapat mendapatkan umpan balik dari pelanggan berdasarkan pada evaluasi representasi perangkat lunak yang dibangun dan diimplementasikan selama masa pemasangan.

Adapun kelebihan pada model *spiral* yang digunakan dalam penelitian ini adalah sebagai berikut (Narafa, 2013):

1. Dapat disesuaikan pada pembangunan maupun perubahan yang terjadi pada perangkat lunak secara sistematis dalam skala besar.
2. Pengembang dan pengguna dapat mudah memahami dan cepat memerhatikan risiko pada setiap tingkat pengembangan sistem karena perangkat lunak terus bekerja pada setiap proses.
3. Menggunakan *prototype* sebagai mekanisme pengurangan risiko dan pada setiap keadaan di dalam pengembangan produk.

4. Selalu mengikuti langkah-langkah dalam siklus hidup dan memasukkannya ke dalam kerangka kerja yang iteratif.
5. Membutuhkan pertimbangan langsung terhadap risiko teknis sehingga mengurangi risiko sebelum terjadi permasalahan yang lebih serius.

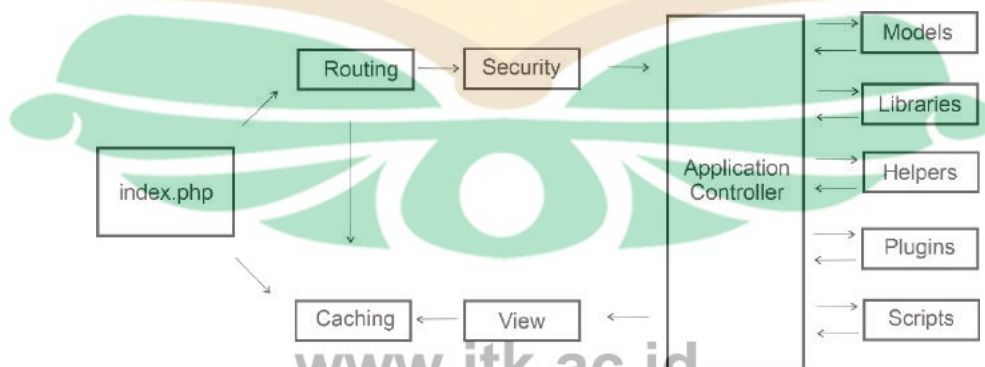
Kelemahan yang terdapat pada model *spiral* adalah sebagai berikut (Nugroho dkk, 2009) :

1. Sulit untuk meyakinkan pengguna bahwa penggunaan pendekatan ini akan dapat dikendalikan.
2. Membutuhkan tenaga ahli yang dapat memperkirakan risiko yang dapat menjadi masalah serius jika risiko paling besar (*major*) tidak ditemukan dan diatur.
3. Membutuhkan waktu yang lama untuk menerapkan model ini.

Penelitian ini menggunakan model *spiral* karena model ini dapat menjadi pendekatan yang realistis untuk perkembangan perangkat lunak, dapat mengecek kemungkinan risiko, serta dapat mencerminkan keadaan sebenarnya. Perangkat lunak terus bekerja seiring dengan proses yang terus bergerak. Oleh karena itu, antara pengembang dan pemakai sistem dapat memahami dan menanggapi lebih baik terhadap risiko dari setiap proses pengembangan perangkat lunak dengan model *spiral*.

2.6 CodeIgniter

CodeIgniter (CI) merupakan salah satu kerangka kerja yang digunakan untuk membangun sebuah sistem atau aplikasi berbasis *Hypertext Preprocessor* (PHP). Cara kerja *CodeIgniter* dijelaskan pada gambar 2.4 di bawah ini (Subagia, 2017).



Gambar 2.4 Cara Kerja *CodeIgniter* (Subagia, 2017)

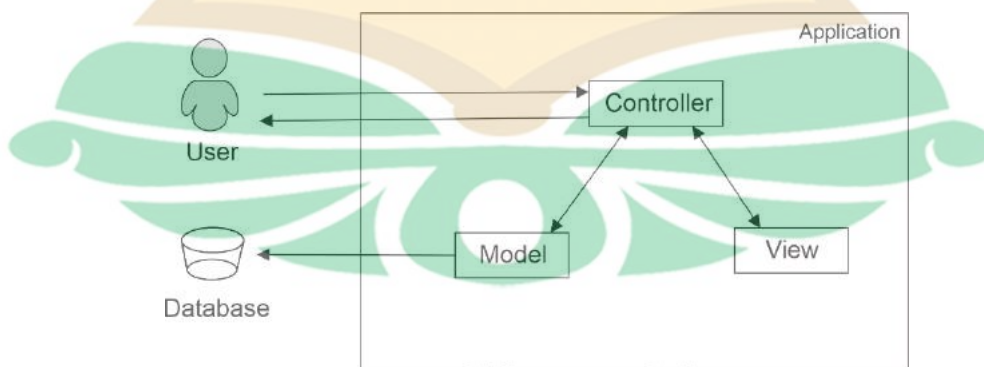
Di bawah ini merupakan penjelasan dari gambar 2.4 tentang cara kerja *CodeIgniter* yaitu sebagai berikut :

1. *Index.php*, bertindak sebagai *controller* terdepan dan mengisi *resource* yang dibutuhkan untuk menjalankan CI.
2. *Router*, memeriksa *http request* agar dapat menentukan apa yang harus dikerjakan.
3. Jika *cache file* ada, maka akan ditampilkan langsung dengan melewati eksekusi normal sistem.
4. Sebelum memuat *controller*, *http request* akan memeriksa apa yang dikirim oleh pengguna dan memfilter keamanan.
5. *Controller*, berisi *model*, *core libraries*, *plugin*, *helper*, dan *resource* lainnya untuk memproses permintaan tertentu.
6. *View* ditampilkan di *browser* sesuai dengan proses yang dikerjakan oleh *controller*. Jika *caching* sedang berjalan, maka *view* akan dilakukan *cache* terlebih dahulu agar dapat ditampilkan di permintaan selanjutnya.

Kerangka kerja CI memiliki kelebihan sebagai berikut (Subagia, 2017):

1. *Free*, karena berada di bawah lisensi *open souce* sehingga pengembang dapat melakukan apapun dengan menggunakan CI.
2. *Fast*, berdasarkan performa yang dimiliki oleh CI terbukti cepat dibandingkan dengan kerangka kerja lainnya menurut dokumentasinya.
3. Menggunakan kaidah *Model-View-Controller* (MVC) yang memisahkan bagian *logic* dan *presentation* dari sistem atau aplikasi yang dibangun.

Pada gambar 2.5 di bawah ini menjelaskan tentang cara kerja dari *Model-View-Controller* (MV) (Subagia, 2017):



Gambar 2.5 Cara Kerja MVC

Adapun penjelasan dari gambar 2.5 adalah sebagai berikut :

- a. *Model*, menggambarkan struktur data. Biasanya, kelas model akan berisi fungsi yang digunakan untuk mengambil, menambah, dan memperbaharui informasi yang ada di *database*.
- b. *View*, informasi yang diperlihatkan kepada pengguna. *View* adalah halaman dari *web* yang terdiri dari HTML, CSS, dan *Javascript*. Namun, pada *CodeIgniter*, *view* dapat juga sebagai potongan halaman contohnya seperti *header* atau *footer*.
- c. *Controller*, sebagai perantara atau penghubung antara *Model*, *View*, dan *resource* lainnya yang dibutuhkan untuk menangani *http request* dan menghasilkan halaman *web*.

2.7 Black Box Testing

Black box testing atau *behavioral testing* berfokus pada spesifikasi fungsional dari suatu perangkat lunak. Pada pengujian *black box* dimungkinkan untuk menentukan set kondisi *input* yang akan sepenuhnya menjalankan semua kebutuhan fungsional dari sebuah program. Pengujian *black box* mengupayakan untuk menemukan kesalahan dengan kategori sebagai berikut (Pressman, 2002):

1. Fungsi-fungsi yang salah atau hilang
2. Kesalahan antarmuka/*interface errors*
3. Kesalahan struktur data atau akses basis data eksternal
4. Kesalahan dari kinerja sistem
5. Kesalahan inisialisasi atau terminasi

Black box testing dilakukan dengan cara membuat kasus uji (skenario) yang bersifat mencoba semua fungsi dengan menggunakan perangkat lunak apakah sesuai dengan spesifikasi yang dibutuhkan. Kasus uji yang dibuat untuk menguji harus dibuat dengan skenario benar dan salah. Adapun contoh dari *black box testing* adalah sebagai berikut (Salahuddin & Rosa, 2013) :

1. Jika pengguna memasukkan nama pengguna (*username*) dan kata sandi (*password*) yang benar.

Jika pengguna memasukkan nama pengguna (*username*) dan kata sandi (*password*) yang salah, contohnya nama pengguna benar tetapi kata sandi salah, atau sebaliknya, atau keduanya salah.

2.8 Penelitian Terdahulu

Tabel 2.5 di bawah ini merupakan rangkuman hasil penelitian terdahulu yang memiliki keterkaitan dengan penelitian yang telah dilakukan.

Tabel 2.5 Penelitian terdahulu

No.	Peneliti	Tahun	Metode	Kesimpulan
1.	St. Magfirah	2018	Waterfall	Permasalahan yang terjadi pada penelitian ini adalah sistem pengelolaan data yang berjalan pada bagian servis PT. Toyota Hadji Kalla Alauddin yang masih menggunakan <i>Microsoft excel</i> untuk pendataan dimana dalam proses memasukkan data memakan waktu yang lama. Data yang tidak tersimpan dalam satu penyimpanan <i>database</i> akan memungkinkan terjadinya kerusakan atau kehilangan data. Dengan adanya sistem yang dibuat dapat memberikan solusi sebagai pemecahan masalah dalam pengelolaan data <i>service</i> mobil dan memberikan kemudahan bagi pelanggan yang ingin mengetahui jadwal <i>service</i> yang harus dilakukan selanjutnya.
2.	Satria	2018	Waterfall	Permasalahan yang terjadi pada penelitian ini adalah tidak ada

No.	Peneliti	Tahun	Metode	Kesimpulan
3.	Zulhamdi Apriyadi, Samsinar	2018	<i>Waterfall</i>	<p>sistem untuk pencatatan <i>service</i> kendaraan dimana sistem ini dibutuhkan agar kepala bengkel dapat menganalisa kinerja dari teknisi bengkel agar dapat meningkatkan pelayanan <i>service</i>. Dengan adanya aplikasi pencatatan <i>service</i> kendaraan di PT. Astrido Jaya Mobilindo, aplikasi dapat memudahkan pegawai dalam melaksanakan pekerjaannya dan mengurangi kesalahan dalam perhitungan laporan pendapatan karena perhitungan dilakukan oleh sistem.</p> <p>Permasalahan yang terjadi pada penelitian ini adalah proses pencatatan transaksi yang ada di Bengkel Cipta Prima Motor masih menggunakan sistem manual dalam pengolahan data transaksi pelanggan. Hal ini dapat menimbulkan penyajian data kurang baik sehingga terjadi kesalahan dalam penulisan keluhan pelanggan dan kekeliruan dalam penghitungan biaya servis yang menyebabkan kerugian.</p>

No.	Peneliti	Tahun	Metode	Kesimpulan
				Dengan adanya aplikasi tersebut dapat memudahkan dalam menghitung transaksi <i>service</i> . Selain itu, aplikasi berfungsi untuk mengetahui stok persediaan <i>panel body</i> dan memudahkan admin dalam membuat laporan karena semua data disimpan dalam <i>database</i> .
4.	Cornelis Dehotman Trong	2017	Waterfall	Permasalahan yang terjadi pada penelitian ini adalah proses pendaftaran <i>service</i> masih dicatat secara manual dan tidak terdapat riwayat dari perbaikan sebelumnya yang dilakukan oleh pelanggan di Bengkel Selasar Motor. Hal tersebut dapat menyebabkan jika ada kerusakan atau keluhan dari pelanggan, mekanik tidak dapat melihat riwayat perbaikan sebelumnya. Dengan adanya aplikasi ini dapat mengelola data yang berkaitan dengan kegiatan reparasi sepeda motor.
5.	Aera Santiana, Herlawati	2018	Waterfall	Permasalahan yang terjadi pada penelitian ini adalah segala proses pencatatan transaksi di Bengkel Cipta Prima Motor

No.	Peneliti	Tahun	Metode	Kesimpulan
				<p>masih menggunakan sistem manual dalam pengolahan data transaksi pelanggan. Hal ini menyebabkan penyajian data yang kurang baik. Selain itu, terdapat kesalahan penulisan keluhan pelanggan dan kekeliruan dalam penghitungan biaya <i>service</i> yang menyebabkan kerugian. Dengan adanya sistem informasi yang baru, maka dapat meminimalisir kesalahan pada transaksi, memudahkan dalam mencari data barang yang tersedia, serta keakuratan dalam pembuatan laporan transaksi.</p>
6.	<p>Moch. Fatchur Rozy, A. Prasita Nugroho, Moch. Nurcholis</p>	2017	<i>Waterfall</i>	<p>Permasalahan yang terjadi pada penelitian ini adalah sistem pembukuan yang digunakan pada Bengkel Paimo masih menggunakan cara manual, sehingga menghambat proses pelayanan terhadap pelanggan. Dengan adanya aplikasi ini, maka pengembangan sistem yang terkomputerisasi dapat membantu memudahkan kegiatan operasional</p>

No.	Peneliti	Tahun	Metode	Kesimpulan
7.	Ega Bangun Saputro, Robby Rachmatullah	2017	PIECES	<p>perusahaan. Pengolahan data dan pembuatan laporan <i>service</i> melalui sistem menjadi lebih mudah dan efisien. Selain itu, dengan adanya aplikasi ini dapat memudahkan dalam mencari data pelanggan.</p> <p>Permasalahan yang terjadi pada penelitian ini adalah pelayanan administrasi di Bengkel Bowo Motor masih mengadopsi sistem manual, dimana pendataan <i>spare part</i> masih dicatat dengan menggunakan tulisan tangan dan disimpan pada map (<i>snell holder</i>). Hal tersebut dapat menimbulkan pemrosesan data menjadi informasi yang diperlukan oleh bagian gudang tidak berjalan dengan baik. Dengan menerapkan Sistem Informasi Manajemen Bengkel Bowo Motor memudahkan dalam pencatatan dan pengelolaan data pelanggan dan data <i>panel body</i> sehingga pelayanan yang diberikan kepada pelanggan berjalan secara maksimal.</p>

No.	Peneliti	Tahun	Metode	Kesimpulan
8.	Yohannes Yahywa Welim, T.W. Wisjhnuadji, Rasip Firmansyah	2015	Waterfall	<p>Permasalahan yang terjadi pada penelitian ini adalah proses pelayanan transaksi di Bengkel KFMP masih menggunakan sistem manual, mulai dari proses pendaftaran pelanggan pelanggan, pendataan tanda bayar, serta pembuatan laporan yang ditunjukkan kepada bengkel. Hal ini dapat menyebabkan terhambatnya proses pelayanan terhadap pelanggan dan keterlambatan informasi yang akan disampaikan kepada pemilik Bengkel KFMP. Dengan adanya sistem yang komputerisasi dapat membantu manajemen dalam mengambil keputusan dengan menyajikan informasi yang akurat, relevan, dan tepat waktu. Sistem komputerisasi juga dapat membantu dalam pengolahan data dan pembuatan laporan penjualan <i>panel body</i> dan <i>service</i>.</p>
9.	Aris Winardi, Ida Farida	2014	Waterfall	<p>Permasalahan yang terjadi pada penelitian ini adalah sistem</p>

No.	Peneliti	Tahun	Metode	Kesimpulan
				<p>yang ada di bengkel CV. Anugerah mengalami beberapa kendala, mulai dari proses transaksi penjualan dan pelayanan <i>service</i> serta penyimpanan data lainnya yang berhubungan dengan proses penjualan dan pelayanan <i>service</i>, pemesanan, pembelian, dan penerimaan barang barang hingga pembuatan laporan. Dengan adanya aplikasi ini memudahkan dalam proses transaksi penjualan, pelayanan <i>service</i>, proses pemesanan, pembelian dan penerimaan stok barang menjadi lebih terorganisir.</p>
10.	Firzaldy Hanaf	2011	<i>Waterfall</i>	<p>Permasalahan yang terjadi pada penelitian ini adalah terjadi kendala dalam melakukan sebuah transaksi di Bengkel Surya Kencana yang berjalan tidak sesuai dengan rencana sehingga menyebabkan berbagai keluhan pelanggan. Dengan adanya aplikasi yang dibuat dapat memberikan informasi riwayat mobil pelanggan dan memudahkan</p>

No.	Peneliti	Tahun	Metode	Kesimpulan
				dalam melihat kondisi stok barang terkini yang sesuai dengan stok di gudang.

Pada tabel 2.5 menjelaskan tentang beberapa penelitian terdahulu terkait dengan pengembangan sistem informasi reparasi bengkel kendaraan. Penelitian pertama dilakukan oleh St. Magfirah yang membangun sistem pengelolaan data servis mobil berbasis web pada PT. Toyota Hadji Kalla Alauddin Makassar. Penelitian St. Magfirah menggunakan metode *waterfall*. Berdasarkan penelitian dari St. Magfirah, penelitian ini berhasil membangun sistem informasi untuk melakukan pengolahan data servis mobil di PT. Toyota Hadji Kalla Alauddin dan memudahkan pelanggan dalam mengetahui jadwal servis yang harus dilakukan selanjutnya. Hasil pengujian sistem informasi ini menggunakan pengujian *blackbox* dan menunjukkan hasil yang sesuai dengan yang diharapkan oleh peneliti.

Penelitian kedua dilakukan oleh Satria yang membuat sistem informasi berbasis web dengan metode *waterfall*. Sistem informasi ini dibangun karena tidak ada sistem laporan untuk memudahkan kepala bengkel dalam menganalisa kinerja perbaikan yang telah dilakukan oleh bengkel. Dengan adanya aplikasi pencatatan *service* kendaraan di PT. Astrido Jaya Mobilindo, aplikasi tersebut dapat membantu pekerjaan teknisi atau pimpinan dalam melihat laporan produktifitas grafik di bengkel.

Penelitian ketiga dilakukan oleh Zulhamdi Apriyadi dan Samsinar yang membuat sistem informasi jasa *service* dan penjualan *panel body* di bengkel Hafis Motor dengan metode *waterfall*. Dibangun sistem ini karena sistem masih dilakukan secara manual baik dalam melakukan pencatatan jasa *service*, penjualan *spare part*, dan pembuatan laporan bulanan. Data yang dikelola pada transaksi masih menggunakan tulis tangan dan dapat terjadi kesalahan dalam penulisan dan perhitungan pembayaran. Dengan adanya aplikasi yang dibangun dengan menggunakan bahasa pemrograman *Microsoft Visual Basic 2008*, aplikasi ini dapat memudahkan dalam menghitung transaksi *service*. Selain itu, aplikasi berfungsi

untuk mengetahui stok persediaan *panel body* dan memudahkan admin dalam membuat laporan karena semua data disimpan dalam *database*.

Penelitian keempat dilakukan oleh Cornelis Dehotman Trong yang membangun aplikasi penerimaan jasa reparasi sepeda motor pada bengkel Selaras Motor yang berfungsi untuk mengelola data penerimaan servis pada motor dengan metode *waterfall*. Berdasarkan hasil uji coba yang dilakukan, aplikasi jasa reparasi sepeda motor dapat mengelola data yang berkaitan dengan kegiatan reparasi sepeda motor di bengkel Selaras Motor. Aplikasi ini juga dilengkapi dengan laporan *service per periode*.

Penelitian kelima dilakukan oleh Aera Santiana dan Herlawati yang membuat sistem informasi di bengkel Cipta Prima Motor dengan menggunakan metode *Waterfall*. Penelitian ini menghasilkan sistem yang berguna untuk meminimalisir kesalahan pada transaksi, memudahkan dalam mencari data barang yang tersedia, serta keakuratan dalam pembuatan laporan transaksi.

Penelitian keenam dilakukan oleh Moch. Fatchur Rozy, A. Prasita Nugroho, Moch. Nurcholis yang membuat aplikasi pelayanan dan pengelolaan data bengkel secara elektronik berbasis *web* dengan metode *waterfall*. Penelitian ini menghasilkan sistem yang terkomputerisasi yang dapat membantu memudahkan kegiatan operasional perusahaan. Pengolahan data dan pembuatan laporan *service* melalui sistem menjadi lebih mudah dan efisien. Selain itu, dengan adanya aplikasi ini dapat memudahkan dalam mencari data pelanggan.

Penelitian ketujuh dilakukan oleh Ega Bangun Saputro dan Robby Rachmatullah yang merancang dan membangun sistem informasi manajemen bengkel Bowo Motor dengan metode *Performance, Information, Economic, Control, Efficiency, Service* (PIECES). Berdasarkan hasil penelitian yang dilakukan, sistem informasi manajemen bengkel Bowo Motor dapat membantu bagian gudang dalam melakukan pencatatan maupun pencarian data *sparepart* motor sehingga pelayanan penjualan *sparepart* dapat menjadi lebih cepat dari sebelumnya. Selain itu, sistem informasi tersebut dapat membantu bagian *customer service* dalam mengatur antrian pelanggan jasa *service* sehingga tidak akan ada pelanggan yang terlompati dalam antrian jasa pelayanan *service*.

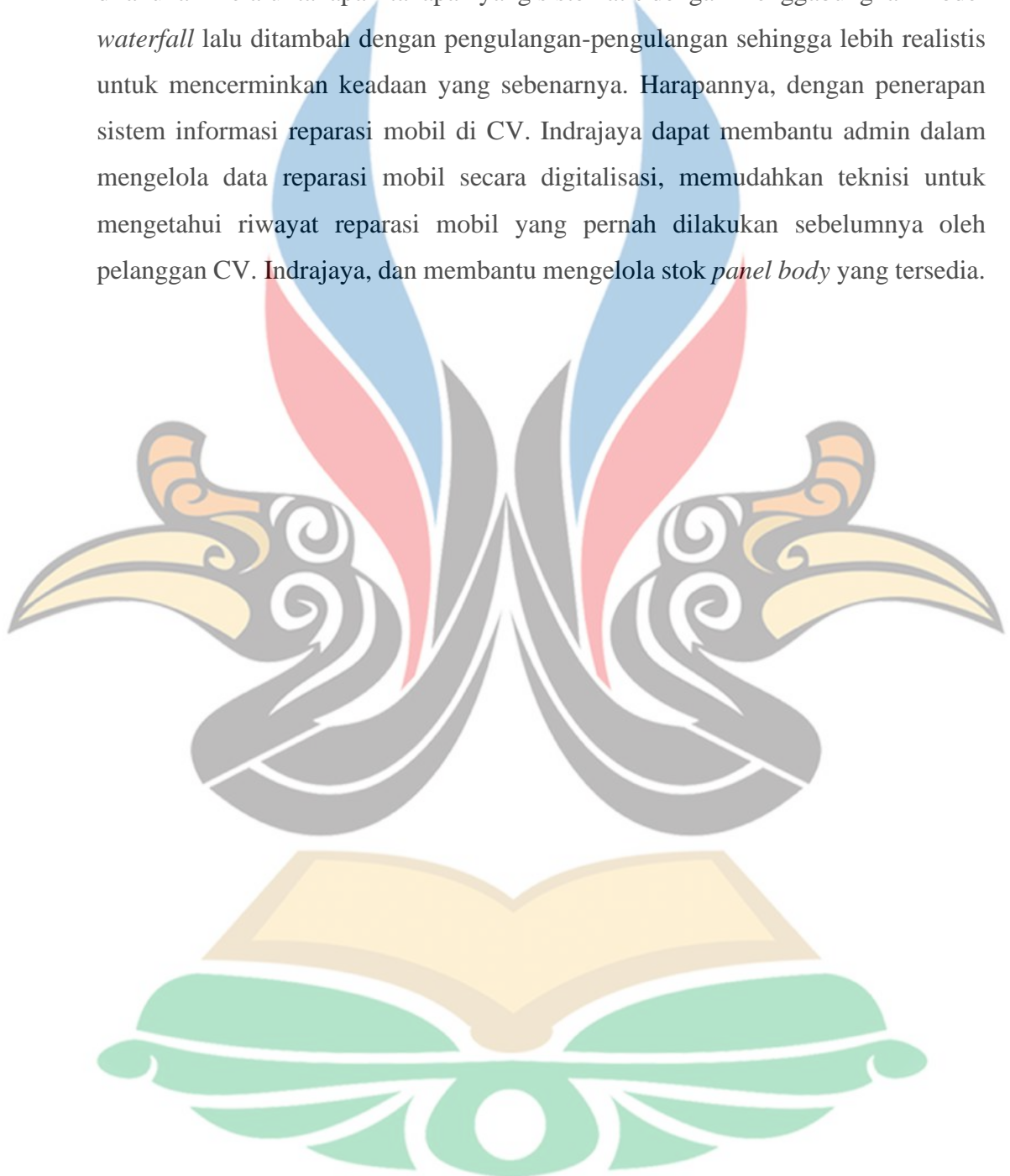
Penelitian kedelapan dilakukan oleh Yohannes Yahya Welim, T.W. Wisjhnuadji, dan Rasip Firmansyah yang mengembangkan sistem informasi servis kendaraan pada bengkel KFMP dengan metode *waterfall*. Penelitian ini menghasilkan sistem yang terkomputerisasi dapat membantu manajemen dalam mengambil keputusan dengan menyajikan informasi yang akurat, relevan, dan tepat waktu. Sistem komputerisasi juga dapat membantu dalam pengolahan data dan pembuatan laporan penjualan *spare part* dan *service*.

Penelitian kesembilan dilakukan oleh Aris Winardi dan Ida Farida yang membuat sistem informasi manajemen bengkel sederhana pada CV. Anugrah Bogor dengan metode *waterfall*. Hasil dari penelitian ini adalah sistem yang dapat memudahkan dalam proses transaksi penjualan, pelayanan *service*, proses pemesanan, pembelian dan penerimaan stok barang menjadi lebih terorganisir. Penelitian kesembilan dilakukan oleh Aera Santiana dan Herlawati yang membuat sistem informasi pelayanan jasa *service* pada bengkel Cipta Prima Motor Cibitung dengan metode *waterfall*. Hasil dari penelitian ini adalah sistem yang dapat meminimalisir kesalahan pada transaksi, memudahkan dalam mencari data barang yang tersedia, serta keakuratan dalam pembuatan laporan transaksi.

Penelitian kesepuluh dilakukan oleh Firzaldy Hanaf yang merancang bangun sistem informasi bengkel mobil berbasis web dengan metode *waterfall*. Penelitian ini menghasilkan aplikasi yang memberikan informasi mengenai riwayat mobil yang pernah melakukan reparasi yang dapat dilihat pada laporan *service* mobil dan pengelolaan stok *spare part*.

Berdasarkan hasil penelitian terdahulu, maka dalam penelitian ini dibuat sebuah sistem informasi reparasi mobil dimana aplikasi ini akan memudahkan pengelolaan data reparasi mobil di CV. Indrajaya dengan model *spiral*. Alasan menggunakan model *spiral* dalam merancang dan membangun sistem informasi reparasi mobil di CV. Indrajaya adalah karena model ini dapat menjadi pendekatan yang realistis untuk perkembangan perangkat lunak, dapat mengecek kemungkinan risiko, serta dapat mencerminkan keadaan sebenarnya. Perangkat lunak terus bekerja seiring dengan proses yang terus bergerak. Model *spiral* membutuhkan pertimbangan langsung terhadap risiko teknis dan manajemen pada semua keadaan proyek, yang jika dipakai dengan benar maka akan mengurangi risiko sebelum

risiko tersebut menjadi besar. Selain itu, antara pengembang dan pengguna sistem dapat memahami dan menanggapi lebih baik terhadap risiko dari setiap proses pengembangan perangkat lunak dengan model *spiral*. Pendekatan model ini dilakukan melalui tahapan-tahapan yang sistematis dengan menggabungkan model *waterfall* lalu ditambah dengan pengulangan-pengulangan sehingga lebih realistis untuk mencerminkan keadaan yang sebenarnya. Harapannya, dengan penerapan sistem informasi reparasi mobil di CV. Indrajaya dapat membantu admin dalam mengelola data reparasi mobil secara digitalisasi, memudahkan teknisi untuk mengetahui riwayat reparasi mobil yang pernah dilakukan sebelumnya oleh pelanggan CV. Indrajaya, dan membantu mengelola stok *panel body* yang tersedia.



www.itk.ac.id