# BAB I PENDAHULUAN

Pada bab pertama ini akan dibahas mengenai beberapa hal. Pembahasan pertama mengenai latar belakang masalah yang mendorong pembuatan Aplikasi Pemesanan Parkir Online “ParkirIn” Berbasis Android dan alasan dikembangkannya sistem ini. Pembahasan berikutnya ialah tujuan dari pembuatan aplikasi “ParkirIn”. Pada bab ini juga akan dibahas mengenai ruang lingkup dari pengembangan sistem yang akan dibuat beserta batasan masalah yang ada. Kemudian juga akan dijelaskan batasan dari system “ParkirIn”, *hardware* dan *software* yang diperlukan untuk membuat aplikasi “ParkirIn” dan sistematika pembahasan untuk buku aplikasi “ParkirIn”.

* 1. **Latar Belakang**

Kota Surabaya telah menjadi kota yang sangat padat penduduk. Jalanan di Surabaya selalu dipenuhi dengan kendaraan-kendaraan bermotor. Dengan keadaan penduduk yang semakin hari semakin bertambah, begitu pula pertumbuhan angka jumlah kendaraan di Surabaya. Dengan bertambahnya jumlah kendaraan tentunya akan berdampak pula dengan lahan parkir yang tersedia di Surabaya. Ketika akhir pekan lahan parkir yang tersedia di lokasi ramai pengunjung seperti *mall* menjadi suatu hal yang berharga yang diperebutkan banyak orang. Lahan parkir ini bersifat spontan, yang berarti lahan parkir dapat digunakan oleh orang yang terlebih dahulu berada di lahan parkir tersebut. Selain itu lahan parkir ini bersifat tidak dapat di pesan. Jumlah lahan parkir di mall juga sering tidak mencukupi kebutuhan masyarakat yang butuh tempat untuk memarkirkan kendaraannya. Hal ini menyebabkan banyaknya kendaraan yang diparkirkan secara paralel di depan kendaraan lain. Lahan parkir yang tersedia sekarang belum bisa mencukupi jumlah kendaraan yang semakin bertambah setiap harinya. Seringkali masyarakat tidak mendapatkan lahan parkir dan baru mengetahuinya saat tiba di lokasi.

Dengan adanya sebuah jasa pemesanan lahan parkir, masyarakat dapat melihat apakah masih ada lahan kosong di suatu lokasi parkir sehingga masyarakat dapat menghindari kejadian seperti tidak mendapat lahan parkir. Masyarakat juga dapat memastikan diri akan mendapatkan lahan parkir saat berpergian karena telah melakukan pemesanan suatu lahan parkir. Adanya jasa ini juga dapat memberikan pendapatan pasif bagi para pemilik lahan yang bekerja sama menyediakan lahannya untuk menjadi lahan parkir.

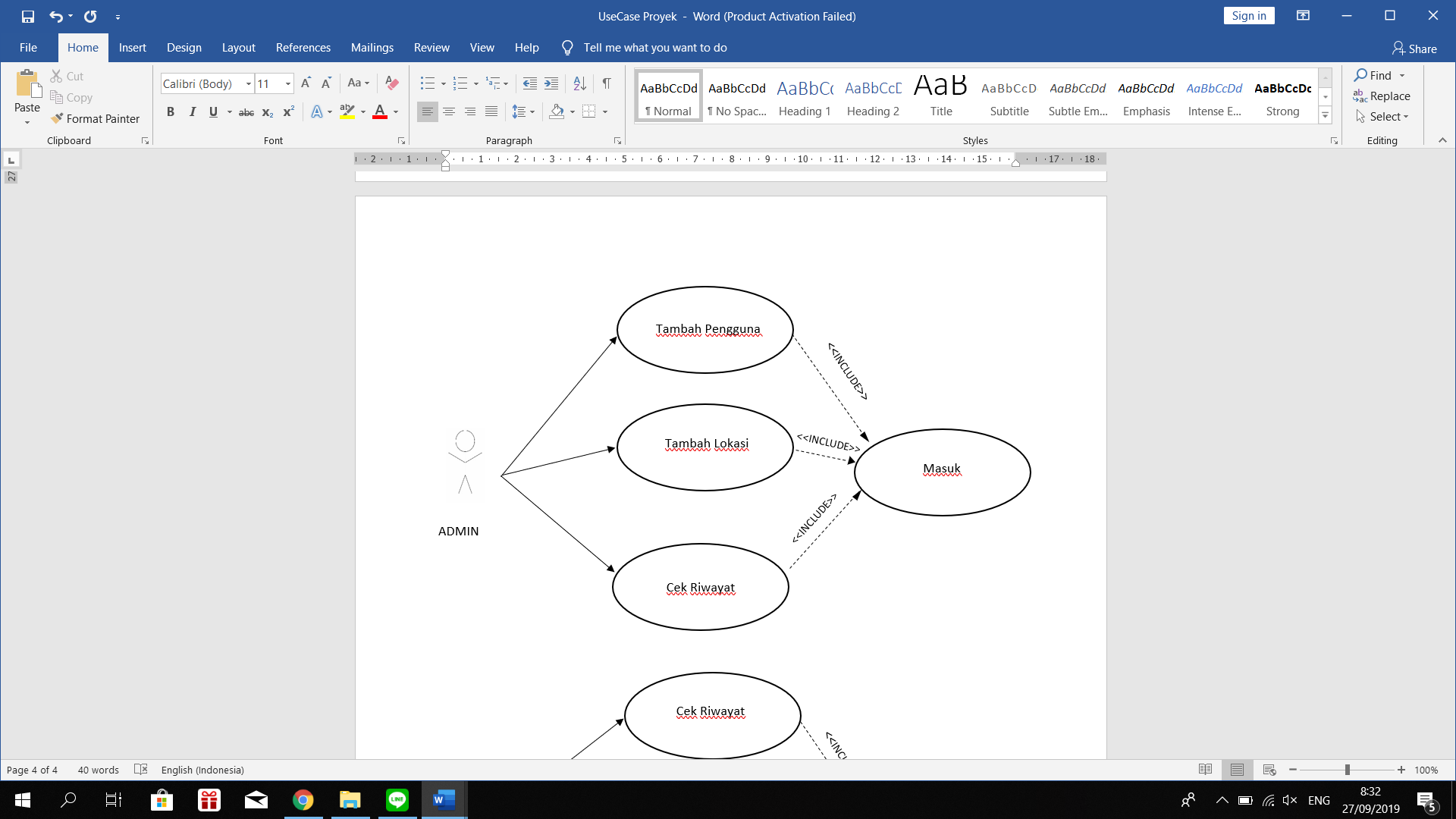
* 1. **Tujuan**

Aplikasi “ParkirIn” ini akan digunakan untuk menanggulangi situasi dan kondisi lahan parkir yang ada pada masa sekarang di Kota Surabaya. Aplikasi ini menguntungkan baik untuk masyarakat dan bagi pemilik lahan parkir di Kota Surabaya. Juga aplikasi “ParkirIn” ini memudahkan dan mengorganisir proses dari parkir kendaraan. Tujuan aplikasi “ParkirIn” secara lengkap adalah sebagai berikut:

* Memenuhi kebutuhan akan lahan parkir dengan menyediakan jasa pemesanan lahan parkir dimana masyarakat dapat memesan lahan parkir di *smartphone* sebelum sampai pada tujuan.
* Memberi informasi tentang keadaan lokasi lahan-lahan parkir yang ada di kota Surabaya dengan menampilkan jumlah kendaraan yang sedang di parkirkan di suatu lokasi pada waktu nyata.
* Memberi pemilik lahan parkir sebuah cara untuk mempermudah dan mengorganisir pemarkiran kendaraan dengan memperbolehkan pemilik lahan parkir untuk bergabung dengan aplikasi dan sistem “ParkirIn”.
* Menyediakan solusi bagi pemilik kendaraan yang tidak dapat merawat kendaraannya karena kendaraan dalam kondisi tertinggal dengan fitur TitipIn, dimana kendaraan dapat dititipkan dan akan dijaga dalam hitungan harian hingga maksimal satu bulan.
  1. **Ruang Lingkup**

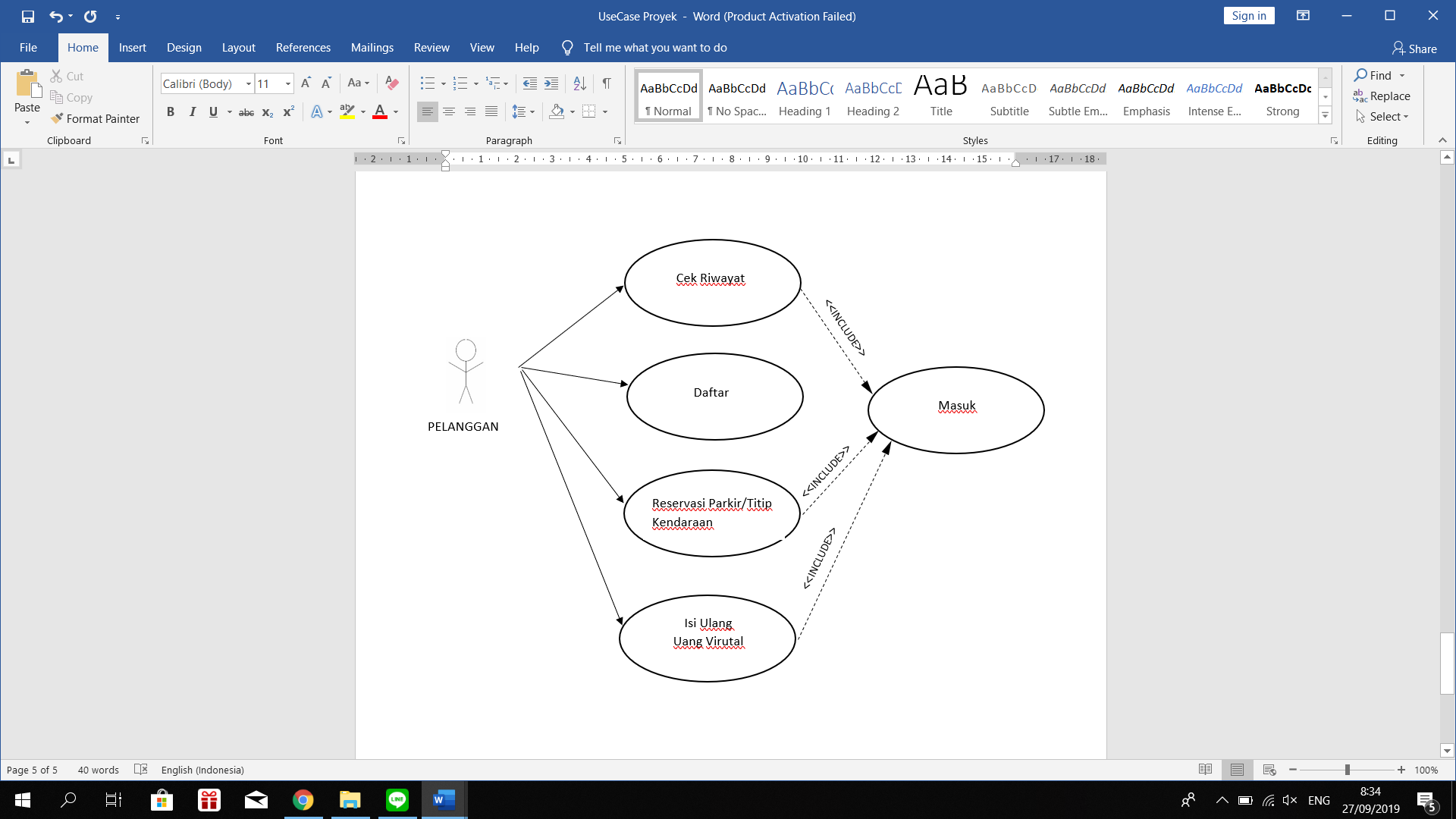
Aplikasi “ParkirIn” berfokus pada area lingkup lahan-lahan parkir berbentuk bangunan yang ada di Kota Surabaya seperti mall. Menyediakan cara untuk pesan lahan parkir sebelum sampai pada tujuan lokasi. Juga memberikan informasi detail tentang lokasi lahan parkir yang terdaftar.

“ParkirIn” juga berfokus untuk mengorganisir dan mengembangkan system parkir yang sudah ada seperti mall, dengan cara bekerja sama dengan pemilik lahan parkir yang sudah ada, seperti mall, dan mengimplementasikan system kami pada sebagian hingga semua lahan parkir yang tersedia. Berikut adalah detail dari bentuk kerja sama yang disediakan oleh “ParkirIn” :

* Pemilik dari Lahan Parkir akan mendapatkan 40% (empat puluh persen) dari pendapatan yang didapatkan dari lokasi tersebut. 

Gambar 1.1 Use Case Admin

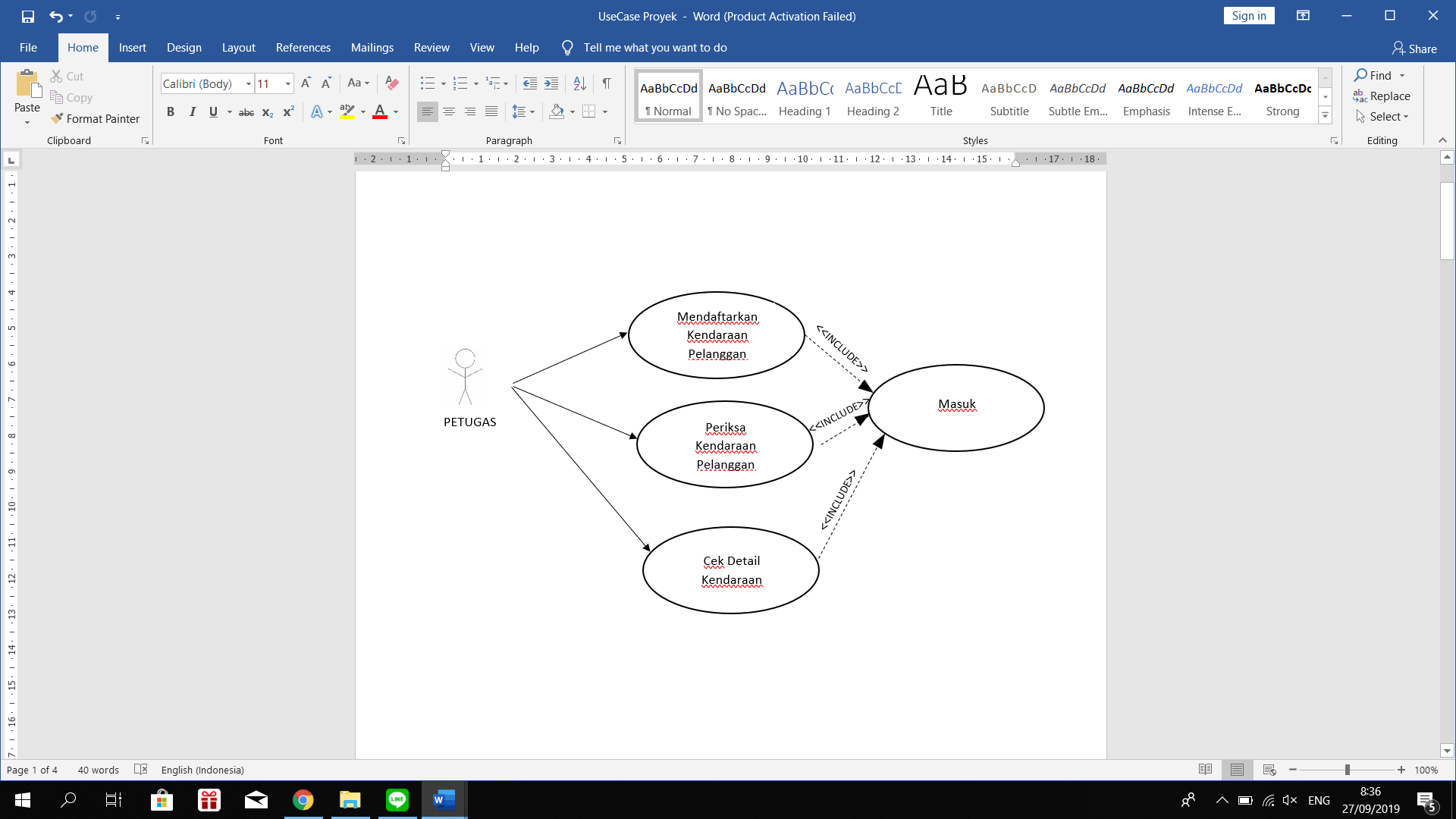
* Aplikasi “ParkirIn” merupakan aplikasi pemesanan parkir, dimana *user* dapat memesan sebuah tempat parkir dengan membuka aplikasi “ParkirIn” di dalam *smartphone*. User dapat memesan duasm jenis parkir, yakni fitur “ParkirIn” untuk parkir jangka waktu jam dan fitur “TitipIn” untuk parkir jangka waktu harian. User dapat menggunakan aplikasi “ParkirIn” dengan mendaftarkan diri melalui aplikasi di smartphone dan dengan memilih lokasi mana yang ingin di tuju. Ketika sampai di tujuan, user hanya perlu menuju parkir milik “ParkirIn” untuk memparkirkan kendaraannya dilanjutkan dengan konfirmasi oleh petugas “ParkirIn”. Kendaraan dapat diambil kapanpun selama tidak melanggar peraturan yang telah dibuat.



Gambar 1.2 Use Case User

“ParkirIn” memiliki harga jasa yang akan dibebankan bagi para pengguna aplikasi “ParkirIn”. Berikut adalah detail dari harga jasa tersebut :

* Untuk pemesanan tempat parkir (Dibawah 24 Jam, biasanya tempat mall), dikenakan Rp.20.000 (dua puluh ribu rupiah) untuk Mobil dan Rp.10.000 (sepuluh ribu rupiah ) untuk motor untuk sekali parkir, termasuk jasa valet.
* Untuk penyimpanan kendaraan (Diatas dua puluh empat Jam, tempat penyimpanan jangka lama), dikenakan Rp.25.000 (dua puluh lima ribu rupiah) untuk mobil dan Rp.12.000 (dua belas ribu rupiah) untuk motor per hari, termasuk jasa perawatan berkala kendaraan
* Denda sebanyak Rp. 30.000 (tiga puluh ribu rupiah) per hari hingga kendaraan diambil untuk fitur “TitipIn”.
* Denda sebanyak Rp. 5.000 (lima ribu rupiah) per jam hingga kendaraan diambil untuk fitur “ParkirIn” (parkir di bawah dua puluh empat jam seperti peraturan di dalam mall)



Gambar 1.3 Use Case Petugas

* 1. **Batasan Sistem**

Batasan sistem adalah batasan-batasan atau hal-hal yang tidak dapat dilakukan dalam pembuatan ataupun pemakaian. Batasan ini terjadi karena berbagai aspek seperti lokasi dan kondisi lingkungan. Aplikasi “ParkirIn” memiliki batasan sistem yang cukup luas dikarenakan lokasi . Batasan sistem meliputi banyak hal, berikut adalah batasan-batasan tersebut :

* Aplikasi “ParkirIn” terbatas di dalam kota Surabaya
* Mata Uang yang digunakan hanya mata uang virtual, yakni CashIn
* Lahan parkir yang dapat menggunakan sistem “ParkirIn” harus merupakan Lahan parkir yang di dalam bangunan
* Pemarkiran kendaraan di dalam bangunan “ParkirIn” harus melalui aplikasi
* Kendaraan yang dapat di parkirkan hanya terbatas ke Kendaraan Roda Empat dan Roda Dua (Mobil dan Motor)
  1. **Hardware**

Hardware adalah segala piranti atau komponen dari sebuah komputer yang sifatnya bisa dilihat secara kasat mata dan bisa diraba secara langsung. Hardware atau perangkat keras berfungsi untuk mendukung segala proses dalam komputerisasi dan bekerja sesuai perintah yang sudah ditentukan penggunanya. Dengan adanya komponen hardware, maka bentuk output dari setiap proses bisa direalisasikan. Dalam aplikasi “ParkirIn” dibutuhkan beberapa peranti keras atau hardware untuk membuat dan sekaligus menggunakan. Hardware yang digunakan adalah :

* Komputer dengan minimal spesifikasi *RAM* sebanyak 4 *gigabyte*.
* Sebuah *monitor* untuk menampilkan output

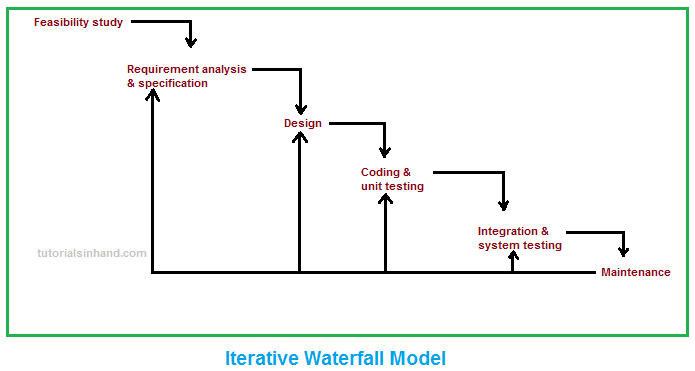
**1.6 Software**

Software adalah suatu bagian dari sistem komputer yang tidak memiliki wujud fisik dan tidak terlihat karena merupakan sekumpulan data elektronik yang disimpan dan diatur oleh komputer berupa program yang dapat menjalankan suatu perintah. Software juga dapat didefinisikan sebagai suatu data yang diprogram, diformat, dan disimpan secara digital, tidak berbentuk fisik tapi dapat dioperasikan oleh penggunanya melalui perangkat komputer. Software atau perangkat lunak merupakan jembatan penghubung yang menghubungkan antara pengguna dengan hardware sehingga dapat melakukan suatu perintah tertentu. Dalam pembuatan aplikasi “ParkirIn”, selain hardware dibutuhkan juga peranti lunak atau software , berikut adalah software-software tersebut :

* Android Studio (Version 3.5)
* Firebase Real-Time Database SDK (Version 19.2)
* Google Maps API (Version 17.0.0)
* Visual Studio Code (Version 13.9)
* Windows 10 (Version 1903)
* Git (2.23.0 Release)

**1.7 Metodologi**

Metode yang digunakan dalam pembuatan aplikasi “ParkirIn” adalah metode *Iterative Waterfall*, yakni sebuah metode yang terbagi menjadi fase dimana suatu fase dapat dijalankan ketika fase sebelumnya telah diselesaikan. Sehingga proses pengembangan dari aplikasi “ParkirIn” dapat dikatakan sebagai *sequential*. Akan tetapi, walaupun berbentuk sequential, pada setiap fase yang ada terdapat *feedback*, yaitu setiap fase dapat diulang kembali jika ada suatu masalah atau *bug* yang terdapat di aplikasi “ParkirIn”. Berikut adalah tahap-tahap dari metode Iterative Waterfall :



Gambar 1.4 Metodologi Iterative Waterfall

* **Studi Kelayakan**

Pada fase ini, kita menentukan apakah suatu project dapat dikembangkan secara finansial dan teknikal. Juga mencari masalah dan strategi untuk menyelesaikan masalah tersebut

* **Analisa Kebutuhan dan Spesifikasi**

Fase ini menganalisa dan menentukan kebutuhan dari aplikasi “ParkirIn” serta spesifikasi yang harus dibuat untuk aplikasi “ParkirIn”

* **Design**

Pada fase ini, dibentuk sistem dari aplikasi “ParkirIn”

* **Implementasi**

Di fase ini adalah dimana aplikasi “ParkirIn” mulai dibuat secara *code*. Juga termasuk *testing* secara unit

* **Integrasi dan Testing Sistem**

Dalam fase ini, aplikasi “ParkirIn” dicoba dan di implementasikan secara system. Dilakukan sebuah simulasi untuk aplikasi “ParkirIn”

* **Pemeliharaan**

Fase ini adalah fase pemeliharaan dari system yang telah di implementasikan jika adanya bug ataupun suatu pemeliharaan konstan\

**1.8 Sistematika Pembahasan**

Dalam penulisan suatu laporan diperlukan sistematika pembahasan yang berisi ringkasan pembahasan dalam setiap bab yang ada dalam buku Pemesananan Parkir Online “ParkirIn”, bertujuan membantu membahas dan melakukan penelitian terhadap buku software development project ini. Sistematika pembahasan buku software development project dengan marketplace ini adalah sebagai berikut :

* Bab 1 : Pendahuluan

Pada bab ini dijelaskan latar belakang, tujuan, ruang lingkup, batasan system, kebutuhan sistem secara hardware dan software, dan juga sistematika pembahasan dari buku.

* Bab 2 : Teori Penunjang

Pada bab ini dijelaskan tentang teori-teori yang menunjang dalam pembuatan Aplikasi “ParkirIn”

* Bab 3 : Analisa Sistem

Bab ini menjelaskan secara detail tentang bagaimana system dari “ParkirIn” dan juga system yang mirip dengan system “ParkirIn”, perbandingannya, dan juga kelebihan dan kekurangannya.

* Bab 4 : Desain Sistem

Pada bab ini dijelaskan bentuk dari system “ParkirIn”, bentuk database dan cara kerja dari system “ParkirIn”

* Bab 5 : Implementasi

Bab ini menjelaskan bagaimana isi dari aplikasi “ParkirIn” dalam bentuk code dan pembuatannya.

* Bab 6 : Uji Coba

Dijelaskan mengenai pencobaan dari aplikasi “ParkirIn” dan implementasinya di lingkungan nyata.

* Bab 7 : Penutup

Berisi kesimpulan dan garis besar mengenai aplikasi “ParkirIn”

# BAB II TEORI PENUNJANG

Pada bab ini akan dijelaskan teori dasar dari setiap hal yang digunakan untuk proyek ini. Penjelasan akan ditekankan pada bagian penting yang digunakan dalam proyek ini. Terdapat 2 sub bab yang akan menjelaskan tentang *front end* dan *back end* aplikasi. Sub bab Front end akan menjelaskan tentang layout pada android dan sub bab back end akan menjelaskan tentang Mobile device programming, MVC, API, database, dan android.

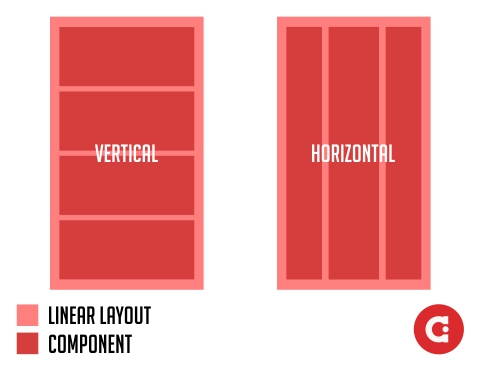
* 1. **Front End**

*User interface* (UI) adalah bagian visual dari website, aplikasi software atau device hardware yang memastikan bagaimana seorang user berinteraksi dengan aplikasi atau website tersebut serta bagaimana informasi ditampilkan di lay arnya. User interface sendiri menggabungkan konsep desain visual, desain interasi, dan infrastruktur informasi. Tujuan dari user interface adalah untuk meningkatkan usability dan tentunya user experience. Media yang dapat digunakan pengguna untuk berinteraksi dengan program (aplikasi atau website) dapat berupa tampilan layar (layout), keyboard, dan mouse. Berikut hal-hal yang harus diperhatikan dalam merancang User Interface yang baik konsistensi, kesederhanaan, responsibilitas, efefktivitas, efisiensi, dan layout.

Program yang baik tentunya adalah program yang memiliki layouting yang nyaman bagi penggunanya. Layout merupakan sebuah struktur visual yang digunakan untuk user interface dari suatu aplikasi . Layout pada Android Studio sendiri adalah suatu tampilan tata letak untuk mengatur penempatan teks, gambar maupun atribut lainnya yang sudah terkonsep. Jadi layout di sini adalah bagian terpenting untuk memperindah tampilan pada aplikasi yang dibuat agar pengguna dapat menggunakan aplikasi dengan nyaman. Terdapat beberapa jenis layout pada Android Studio, antara lain :

* Linear Layout

Layout yang menyejajarkan semua child view-nya dalam satu arah, secara vertikal atau horizontal.

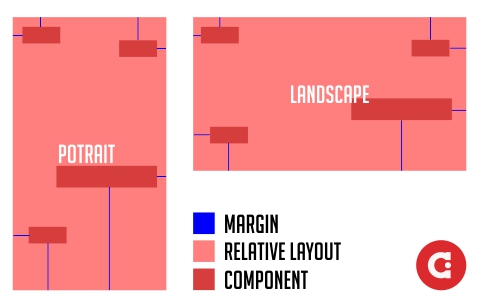


Gambar 2.1 Linear Layout

Semua anak LinearLayout akan ditumpuk satu sama lain, jadi daftar vertikal hanya akan memiliki satu anak per baris, seberapa pun lebarnya, dan daftar horizontal hanya akan setinggi satu baris (tinggi anak yang tertinggi, ditambah pengisi). LinearLayout akan mengikuti margin antara anak dan gravity (sejajar kanan, tengah, atau kiri) setiap anak.

* Relative Layout

Layout yang penataan nya ini adalah penataan yang menempatkan widget-widget didalamnya seperti layer, sehingga sebuah widget dapat berada di atas/di bawah widget lainnya atau dengan kata lain Relative merupakan layout yang penataannya lebih bebas (Relative) sehingga bisa di tata di mana saja.

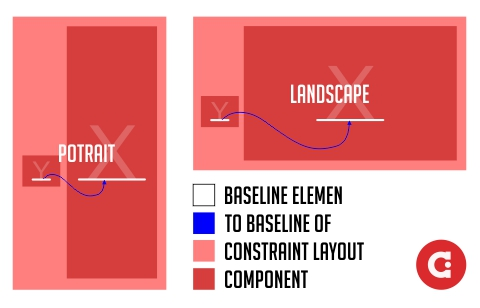


Gambar 2.2 Relative Layout

Jadi dapat diartikan bahwa Relative Layout adalah desain tampilan pada aplikasi dengan tata letak objek atau komponen secara bebas tanpa aturan sesuai orientasi (horizontal atau vertical) seperti yang diterapkan pada Linear Layout.

* Constraint Layout

Constraint Layout mirip dengan Relative Layout, karena letak View bergantung pada View lain dalam satu Layout ataupun dengan Parent Layoutnya. Contohnya di Relative Layout sebuah View bisa diletakkan di atas, bawah, atau samping View lain.



Gambar 2.3 Constraint Layout

* Frame Layout

Frame Layout adalah Layout yang biasanya digunakan untuk membuat objek yang saling bertindihan contohnya yaitu kita membuat button di atas image

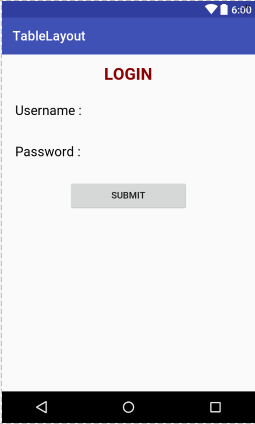


Gambar 2.4 Frame Layout

* Table Layout

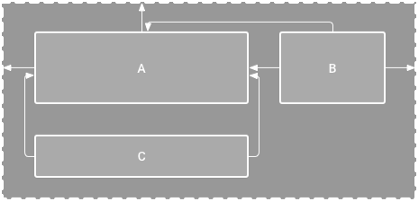
Table Layout adalah Layout yang digunakan untuk membangun user interface (tampilan antar muka ) aplikasi android dengan berdasarkan Baris dan Kolom. Table Layout terdiri dari:

1. Row/ baris pada dasarnya digunakan untuk menyimpan satu jenis record, hanya satu informasi yang dapat disimpan
2. Kolom adalah sub bagian terbagi dari setiap baris dan satu baris dapat menampung beberapa jenis kolom. Setiap kolom terdiri dari jenis informasi yang berbeda mengenai baris tersebut.



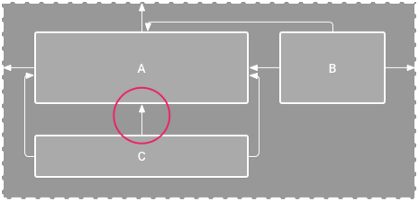
Gambar 2.5 Table Layout

Aplikasi “Parkirin” ini dikembangkan dengan menggunakan Constraint Layout karena penggunaannya yang sangat mudah dan fleksibel dalam pengaturan komponen-komponennya. Pada Constraint Layout, setiap View memiliki tali (Constraint) yang menarik tiap sisinya. Tali tersebut wajib diikatkan kepada anchor point atau suatu titik yang dapat berupa sisi dari Parent Layout, View lain, ataupun suatu titik bantu (helper). Tanpa constraint tersebut, sebuah Layout secara default akan terletak di posisi [0,0] (sebelah kiri-atas), sehingga sebuah View paling tidak membutuhkan 2 Constraint untuk sisi X dan sisi Y.



Gambar 2.6 Constraint Layout yang salah

Pada Gambar 2.6, Layout terlihat normal pada Layout Editor. Namun jika diperhatikan, View C tidak memiliki Vertical Constraint. Jika program dijalankan maka View C akan sesuai Horizontal Constraintnya (sisi kiri dan kanan View A), tapi terletak di bagian atas Parent Layout (titik 0).



Gambar 2.7 Constraint Layout yang benar

Pada Gambar 2.7, tampilan aplikasi yang dijalankan akan sesuai dengan tampilan pada Layout Editor. Selain layouting ada banyak komponen lain yang digunakan dalam pengembangan aplikasi “Parkirin”. Di antara lain ada TextView, EditText, Button, Radio, Spinner, ListView,

* 1. **Back End**

*Mobile Device Programming* merupakan pemrograman yang digunakan perangkat mobile(*mobile device*). Perangkat mobile adalah istilah umum untuk semua jenis komputer yang dapat digenggam. Perangkat ini dirancang untuk mudah dibawa kemanapun (portabel) dan dapat digenggam dengan tangan. Beberapa jenis perangkat mobile seperti tablet, e-reader (alat baca buku elektronik), dan smartphone sudah sangat mampu mengerjakan tugas-tugas yang umumnya dikerjakan di laptop atau komputer desktop.

Pemrograman adalah proses menulis, menguji dan memperbaiki *(debug*), dan memelihara kode yang membangun suatu program komputer. Kode ini ditulis dalam berbagai bahasa pemrograman. Tujuan dari pemrograman adalah untuk memuat suatu program yang dapat melakukan suatu perhitungan atau 'pekerjaan' sesuai dengan keinginan si pemrogram. Untuk melakukan pemrograman, diperlukan keterampilan dalam algoritme, logika, bahasa pemrograman, dan pada banyak kasus, pengetahuan-pengetahuan lain seperti matematika. Berikut ini akan dijelaskan tentang berbagai hal yang berhubungan dengan mobile device programming.

**2.2.1 Model-View-Controller**

*Model-View-Controller* atau MVC adalah sebuah metode untuk membuat sebuah aplikasi dengan memisahkan data (*Model*) dari tampilan (*View*) dan cara bagaimana memprosesnya (*Controller*). Dalam implementasinya kebanyakan framework dalam aplikasi website adalah berbasis arsitektur MVC. MVC memisahkan pengembangan aplikasi berdasarkan komponen utama yang membangun sebuah aplikasi seperti manipulasi data, antarmuka pengguna, dan bagian yang menjadi kontrol dalam sebuah aplikasi web.

* Model : Model mewakili struktur data. Biasanya model berisi fungsi-fungsi yang membantu seseorang dalam pengelolaan basis data seperti memasukkan data ke basis data, pembaruan data dan lain-lain.
* Viewers : Viewers adalah bagian yang mengatur tampilan ke pengguna. Bisa di katakan berupa halaman web.
* Controller : Controller merupakan bagian yang menjembatani model dan view. Controller berisi perintah-perintah yang berfungsi untuk memproses suatu data dan mengirimkannya ke halaman web.

Dengan menggunakan metode MVC maka aplikasi akan lebih mudah untuk dirawat dan dikembangkan. Untuk memahami metode pengembangan aplikasi menggunakan MVC diperlukan pengetahuan tentang pemrograman berorientasi objek (*Object Oriented Programming*). Terdapat beberapa jenis MVC, yaitu :

* *Server Side MVC* : Server Side MVC biasa terjadi pada aplikasi web tradisional, yang tidak melibatkan client side seperti Javascript, Java applet, Flash, dan lain-lain. Server Side MVC menyerahkan keseluruhan proses bisnis pada server, aplikasi pada sisi pengguna hanya dapat menerima. MVC jenis ini kadang-kadang disebut juga dengan nama *Thin Client*.
* *Mixed Client Side and Server Side MVC* : Pada Mixed Client Side and Server Side MVC 1 client tidak menggunakan model sebagai jembatan untuk melakukan komunikasi pada server, dibandingkan dengan Server Side MVC, arsitektur ini memiliki tingkat kompleksitas yang lebih tinggi karena lebih banyak komponen yang terlibat. Untuk selanjutnya arsitektur ini disebut, dengan Mixed MVC 1. Pada Mixed Client Side and Server Side MVC 2, client menggunakan model sebagai jembatan untuk melakukan komunikasi pada server, dibandingkan dengan arsitektur MVC yang lain, arsitektur ini memiliki tingkat kompleksitas yang paling tinggi karena lebih banyak komponen yang terlibat, sehingga membutuhkan sumber daya yang lebih besar pula. Untuk selanjutnya arsitektur ini disebut dengan Mixed MVC 2.
* *Rich Internet Application MVC* : Application MVC Rich Internet Application (RIA) disebut juga dengan nama Fat Client, merupakan aplikasi web yang memiliki kemampuan dan fungsi hampir seperti aplikasi desktop. RIA pada sisi client, memiliki mesin untuk mengambil data yang berada pada server, sehingga pada client terdapat bagian MVC sendiri dan hanya membutuhkan bagian model pada sisi server.

**2.2.2 Integrated Development Environment**

Hal yang dibutuhkan dalam programming salah satunya ialah Integrated Development Environment (IDE). IDE adalah program komputer yang memiliki beberapa fasilitas yang diperlukan dalam pembangunan perangkat lunak. Tujuan dari IDE adalah untuk menyediakan semua utilitas yang diperlukan dalam membangun perangkat lunak. Sebuah IDE, atau secara bebas dapat diterjemahkan sebagai Lingkungan Pengembangan Terpadu, setidaknya memiliki fasilitas:

* Editor, yaitu fasilitas untuk menuliskan kode sumber dari perangkat lunak.
* Compiler, yaitu fasilitas untuk mengecek sintaks dari kode sumber kemudian mengubah dalam bentuk binari yang sesuai dengan bahasa mesin.
* Linker, yaitu fasilitas untuk menyatukan data binari yang beberapa kode sumber yang dihasilkan compiler sehingga data-data binari tersebut menjadi satu kesatuan dan menjadi suatu program komputer yang siap dieksekusi.
* Debuger, yaitu fasilitas untuk mengetes jalannya program, untuk mencari bug/kesalahan yang terdapat dalam program.­­­

Sampai tahap tertentu IDE modern dapat membantu memberikan saran yang mempercepat penulisan. Pada saat penulisan kode, IDE juga dapat menunjukan bagian-bagian yang jelas mengandung kesalahan atau keraguan. Terdapat banyak ide yang populer saat ini dan digunakan oleh banyak programmer. Salah satu contoh IDE yang digunakan untuk mengembangkan aplikasi android, yaitu Android Studio.

Android Studio adalah Integrated Development Enviroment (IDE) untuk sistem operasi Android, yang dibangung diatas perangkat lunak JetBrains IntelliJ IDEA dan didesain khusus untuk pengembangan Android. IDE ini merupakan pengganti dari Eclipse Android Development Tools (ADT) yang sebelumnya merupakan IDE utama untuk pengembangan aplikasi android. Android studio sendiri pertama kali diumumkan di Google I/O conference pada tanggal 16 Mei 2013. Ini merupakan tahap preview dari versi 0.1 pada Mei 2013, dan memasuki tahap beta sejak versi 0.8 dan mulai diliris pada Juni 2014.

**2.2.3 Database**

**Database** atau basis data adalah kumpulan berbagai data dan informasi yang tersimpan dan tersusun di dalam komputer secara sistematik yang dapat diperiksa, diolah atau dimanipulasi dengan menggunakan program komputer untuk mendapatkan informasi dari basis data tersebut.

Perangkat lunak yang digunakan untuk mengelolan dan memanggil database disebut dengan sistem database management system. Istilah database sendiri mengacu pada koleksi data-data yang saling terkait satu sama lain dimana tujuan database adalah dapat digunakan untuk mengelola data dengan lebih efektif dan efisien.

1. Kecepatan dan kemudahan (*Speed*) : Untuk dapat menyimpan data atau melakukan perubahan ( manipulasi ) dan menampilkan kembali data tersebut dengan cepat dan mudah, dari pada disimpan secara manual.
2. Efisien ruang penyimpanan (*Space*) : Untuk menekanan jumlah pengulangan data dengan menerapkan sejumlah pengkodean.
3. Keakuratan (*Accuracy*) : Membentuk relasi antar data dengan penerapan aturan atau batasan tipe data dapat diterapkan dalam Database yang berguna untuk menentukan ketidakakuratan pemasukan atau penyimpanan.
4. Keamanan (*Security*) : Dalam sejumlah sistem ( aplikasi ) pengelolah database tidak menerapkan aspek keamanan dalam penggunaan database. Tetapi untuk sistem yang besar dan serius, aspek keamanan juga dapat diterapkan. Dengan begitu kita dapat menentukan siapa yang boleh menggunakan database dan menentukan jenis operasi-operasi apa saja yang boleh dilakukan.
5. Terpeliharanya keselarasan data (*Consistant*) : Apabila ada perubahan data pada aplikasi yang berbeda maka secara otomatis perubahan itu berlaku untuk keseluruhan.
6. Data dapat dipakai secara bersama (*Shared*) : Data dapat dipakai secara bersama-sama oleh beberapa program aplikasi (secara batch maupun on-line) pada saat bersamaan.
7. Dapat diterapkan standarisasi (*Standardization*) : Dengan adanya pengontrolan yang terpusat maka DBA dapat menerapkan standarisasi data yang disimpan sehingga memudahkan pemakaian, pengiriman maupun pertukaran data.

Firebase adalah suatu layanan dari Google yang digunakan untuk mempermudah para pengembang aplikasi dalam mengembangkan aplikasi. Dengan adanya Firebase, pengembang aplikasi bisa fokus mengembangkan aplikasi tanpa harus memberikan usaha yang besar. Dua fitur yang menarik dari Firebase yaitu Firebase Remote Config dan Firebase Realtime Database. Selain itu terdapat fitur pendukung untuk aplikasi yang membutuhkan pemberitahuan yaitu Firebase Notification.

Firebase berevolusi dari Envolve, startup sebelumnya yang didirikan oleh James Tamplin dan Andrew Lee pada tahun 2011. Envolve menyediakan pengembang Antarmuka pemrograman aplikasi yang memungkinkan integrasi fungsi obrolan daring ke situs web mereka. Setelah merilis layanan obrolan, Tamplin dan Lee menemukan bahwa Envolve digunakan untuk mengirimkan data aplikasi yang bukan pesan obrolan. Pengembang menggunakan Envolve untuk menyinkronkan data aplikasi seperti status permainan secara waktu nyata di seluruh penggunanya. Tamplin dan Lee memutuskan untuk memisahkan sistem obrolan dan arsitektur waktu nyata yang menjalankannya. Tamplin dan Lee mendirikan Firebase sebagai perusahaan terpisah pada bulan April 2012.

Firebase Inc. mengumpulkan dana awal pada bulan Mei 2012. Perusahaan ini kemudian meningkatkan pendanaan Seri A pada bulan Juni 2013. Pada bulan Oktober 2014, Firebase diakuisisi oleh Google. Pada bulan Oktober 2015, Google mengakuisisi Divshot untuk menggabungkannya dengan Tim Firebase. Sejak akuisisi, Firebase telah berkembang di Google dan memperluas layanan untuk menjadi platform terpadu bagi pengembang aplikasi. Firebase sekarang terintegrasi dengan berbagai layanan Google lainnya, untuk menawarkan produk dan skala yang lebih luas bagi pengembang aplikasi. Pada bulan Januari 2017, Google mengakuisisi Fabric dan Crashlytics dari Twitter untuk bergabung dengan layanan tersebut ke Tim Firebase. Firebase meluncurkan Cloud Firestore, sebuah database dokumen, pada bulan Oktober 2017.

* + 1. **Application Programming Interface**

Application Programming Interface (API) adalah sekumpulan perintah, fungsi, dan protokol yang dapat digunakan oleh programmer saat membangun perangkat lunak untuk sistem operasi tertentu. API memungkinkan programmer untuk menggunakan fungsi standar untuk berinteraksi dengan sistem operasi lain.

Dalam contoh sederhana, dibutuhkan setidaknya ribuan system calls per detik. Oleh karena itu Kebanyakan programmer membuat aplikasi dengan menggunakan API. Dalam API itu terdapat fungsi-fungsi/perintah-perintah untuk menggantikan bahasa yang digunakan dalam system calls dengan bahasa yang lebih terstruktur dan mudah dimengerti oleh programmer. Fungsi yang dibuat dengan menggunakan API tersebut kemudian akan memanggil system calls sesuai dengan sistem operasinya. Tidak tertutup kemungkinan nama dari system calls sama dengan nama di API. Keuntungan memprogram dengan menggunakan API adalah:

* Portabilitas. Programmer yang menggunakan API dapat menjalankan programnya dalam sistem operasi mana saja asalkan sudah ter- install API tersebut. Sedangkan system callberbeda antar sistem operasi, dengan catatan dalam implementasinya mungkin saja berbeda.
* Lebih Mudah Dimengerti. API menggunakan bahasa yang lebih terstruktur dan mudah dimengerti daripada bahasa system call. Hal ini sangat penting dalam hal editing dan pengembangan.

System call interface ini berfungsi sebagai penghubung antara API dan system call yang dimengerti oleh sistem operasi. System call interface ini akan menerjemahkan perintah dalam API dan kemudian akan memanggil system calls yang diperlukan.

# BAB III ANALISA SISTEM

Pada bab ini akan dijelaskan tentang sistem aplikasi pembanding dari aplikasi “ParkirIn” dan juga tentang sistem aplikasi “ParkirIn” itu sendiri. Pembahasan selanjutnya adalah mengenai masalah apa saja yang ada pada sistem aplikasi yang sebelumnya sudah ada dan juga fitur-fitur baru yang ada pada aplikasi “ParkirIn”. Aplikasi “ParkirIn” ini tentunya dibangun dengan harapan dapat memenuhi kekurangan-kekurangan dari aplikasi serupa yang sudah ada sebelumnya. Sehingga kualitas dari aplikasi “ParkirIn” dapat dipertaruhkan di pasar digital.

* 1. **Analisa Sistem Lama**

Aplikasi “Cari Parkir” sebagai pembanding aplikasi “Parkirin” sudah memiliki fitur-fitur yang membantu user antara lain untuk mencari tempat-tempat parkir dan lahan yang tersedia. Terdapat pula fitur locate current location untuk mencari lahan parkir terdekat. Selain itu juga ada fitur navigasi yang membantu user untuk menuju lokasi parkir yang diinginkan dengan menggunakan aplikasi pihak ketiga “Google Maps”. Tetapi masih banyak kekurangan dan masalah yang muncul dari keterbatasan aplikasi ini, antara lain :

1. Tidak adanya fitur pemesanan / booking di setiap tempat parkir.
2. Tidak adanya fitur pembayaran non-tunai / cashless
3. Kurangnya informasi mengenai denda parkir jika user memarkirkan kendaraannya melebihi batas waktu.
4. Tidak ada fitur kerjasama bagi pemilik lahan parkir yang ingin menyewakan lahan parkirnya.
   1. **Analisa Masalah**

Kekurangan dan keterbatasan sistem menjadi masalah tersendiri bagi user selaku pengguna aplikasi. Beberapa hal dapat menghambat kenyamanan user untuk menjelajahi aplikasi. Bahkan tidak memungkinkan juga apabila ada user yang berhenti memakai layanan dari suatu aplikasi karena keterbatasannya. Berikut adalah beberapa analisa masalah yang bisa terjadi karena kekurangan dari sistem :

1. Keterbatasan sistem yang tidak memungkinkan user melakukan pemesanan/booking lahan parkir akan menimbulkan masalah karena user bisa saja kehabisan lahan parkir saat tiba di tujuan.
2. Pembayaran non–tunai adalah trend yang sedang menjamur di khalayak masyarakat, sudah banyak masyarakat yang beralih dari pembayaran tunai ke non-tunai. Bahkan ada banyak orang yang tidak membawa uang tunai saat bepergian.
3. Pada hakikatnya sistem parkir di suatu tempat memiliki batas maksimal, biasanya pada tengah malam atau pada subuh hari selanjutnya. Sistem akan mengingatkan melalui notifikasi jika waktu pemesanan akan habis dan dapat memberikan pilihan untuk menambah waktu pesanan.
4. Pemilik lahan tidak dapat mengajukan kerja sama dengan pihak developer/owner dari project ini. Hal tersebut tentunya juga akan mengurangi kesempatan dari pengembangan aplikasi ini karena keterbatasan lahan parkir yang hanya disediakan dari pihak developer.

* 1. **Analisa Sistem Baru**

Pada aplikasi “Parkirin” terdapat beberapa penambahan fitur – fitur untuk menambah kepraktisan dan kemudahan dalam menggunakan aplikasi. Fitur yang ditambahkan salah satunya ialah penitipan kendaraan jangka panjang yang akan berguna jika kendaraan ditinggal pergi lama oleh pemiliknya. Fitur lainnya yang menambah kepraktisan yaitu pembayaran cashless dimana user tidak perlu lagi menyiapkan uang tunai dan tidak perlu pusing memikirkan jika tidak ada kembalian. Kurangnya lahan pada aplikasi pembanding juga dapat diatasi dengan kerjasama dengan pihak – pihak yang ingin lahannya digunakan sehingga lahan parkir dapat tersedia secara merata dan user memiliki lebih banyak pilihan lokasi. Beberapa fitur yang ditambahkan, yaitu :

1. Booking/pemesanan parkir : user dapat memesan terlebih dahulu lahan parkir untuk parkir atau penitipan sehingga user tidak akan kehabisan lahan parkir ketika datang ke lokasi.
2. Pembayaran non-tunai : pembayaran dilakukan terlebih dahulu sebelum user datang ke lokasi sehingga ketika sampai di lokasi user tidak perlu membayar lagi.
3. Sistem pengingat : Sistem akan mengingatkan user jika waktu pemesanan akan habis dan dapat menambah waktu sehingga kemungkinan terkena denda akan berkurang.
4. Sistem kerjasama : pemilik lahan yang ingin lahannya dimanfaatkan untuk parkir dapat menghubungi developer untuk kerjasama sehingga lokasi parkir akan lebih bervariasi dan merata di kota surabaya. User juga dapat lebih fleksibel untuk menentukan lokasi.
   1. **Fitur Aplikasi**

Untuk mengatasi kekurangan dan keterbatasan pada sistem yang sudah ada sebelumnya, tentunya dibutuhkan fitur-fitur pendukung agar user dapat mengalami pengalaman menggunakan layanan yang lebih baik. Fitur-fitur yang disediakan pada aplikasi “ParkirIn” diperoleh dari analisa kebutuhan user sehingga dapat meminimalisir keterbatasan dan masalah yang terjadi. Fitur-fitur yang disediakan aplikasi “ParkirIn” dibagi menjadi 3 (tiga) pihak yaitu dari pihak User, Admin dan Petugas. Fitur-fitur tersebut antara lain :

1. User :
2. Cek Riwayat

User dapat melihat riwayat penitipan kendaraan di aplikasi ini.

1. Pemberitahuan lahan parkir penuh

Jika lokasi parkir penuh maka lokasi parkir tidak dapat dipilih oleh user.

1. Rincian pesanan

User akan mendapatkan rincian pesanan yang dapat digunakan untuk penitipan dan pengambilan.

1. Isi ulang uang virtual

User dapat mengisi ulang uang virtual untuk pembayaran reservasi parkir atau penitipan kendaraan.

1. Pilih lokasi

User dapat memilih lokasi sesuai dengan keinginannya.

1. Cuci mobil

Dalam fitur ini user dapat memesan agar kendaraannya dicuci terlebih dahulu sebelum diambil. Fitur ini hanya berlaku pada penitipan kendaraan.

1. Pemberitahuan waktu parkir atau penitipan

User akan mendapatkan pemberitahuan ketika masa pemesanan parkir sudah akan habis dan dapat menawarkan opsi perpanjangan sewa melalui email.

1. Admin :
2. Master User

Admin dapat mengatur user yang terdaftar dalam aplikasi ini, seperti mengatur data pribadi dari user, mengaktifkan dan menon-aktifkan status user.

1. Master Lokasi

Admin dapat menambah dan mengatur data dari suatu lokasi parkir atau lokasi penitipan kendaraan, dimana lokasi tersebut bisa digunakan oleh user.

1. Laporan

Admin dapat melihat riwayat seluruh user. Admin juga bisa memberikan laporan per-bulan atau per-tahun.

1. Petugas :
2. Mendaftarkan kendaraan user

Disini petugas memasukan data-data kendaraan user untuk parkir di suatu tempat.

1. Periksa kendaraan user

Petugas dapat memeriksa kendaraan user yang diparkirkan atau dititipkan untuk mengecek kondisi kendaraan.

1. Cek detail kendaraan

Petugas dapat melihat detail kendaraan yang akan diparkirkan atau sudah diparkirkan untuk mendapatkan kondisi kendaraan. Di dalam fitur ini juga petugas dapat melihat waktu yang tersisa bagi kendaraan user dan dapat mengeluarkan denda bagi user jika melebihi batas waktunya.

# BAB IV DESAIN

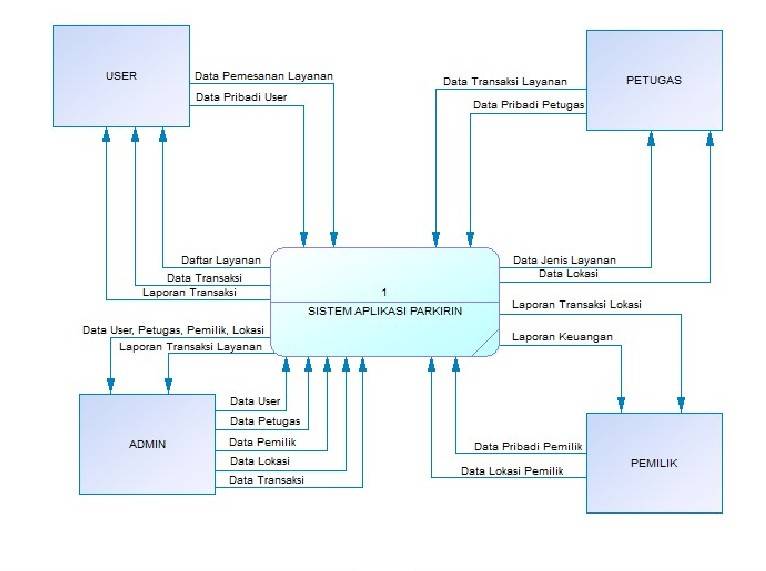
Pada bab ini akan dijelaskan tentang desain dari sistem aplikasi “ParkirIn”. Pembahasan utama adalah tentang desain struktur dari alur sistem yang membentuk aplikasi ini. Selanjutnya akan dibahas juga mengenai desain dari setiap tabel yang ada pada database. Selain itu akan dijelaskan tentang desain-desain tampilan dari keseluruhan aplikasi ini.

**4.1 Data Flow Diagram**

Data Flow Diagram (DFD) adalah alat pembuatan model yang memungkinkan profesional sistem untuk menggambarkan sistem sebagai suatu jaringan proses fungsional yang dihubungkan satu sama lain dengan alur data, baik secara manual maupun komputerisasi. DFD ini adalah diagram yang digunakan untuk memodelkan sistem secara logik. DFD sering juga disebut dengan nama bubble chart, bubble diagram, model proses, diagram alur kerja, atau model fungsi. Data Flow Diagram (DFD) sendiri dibagi menjadi beberapa tingkatan level yaitu:

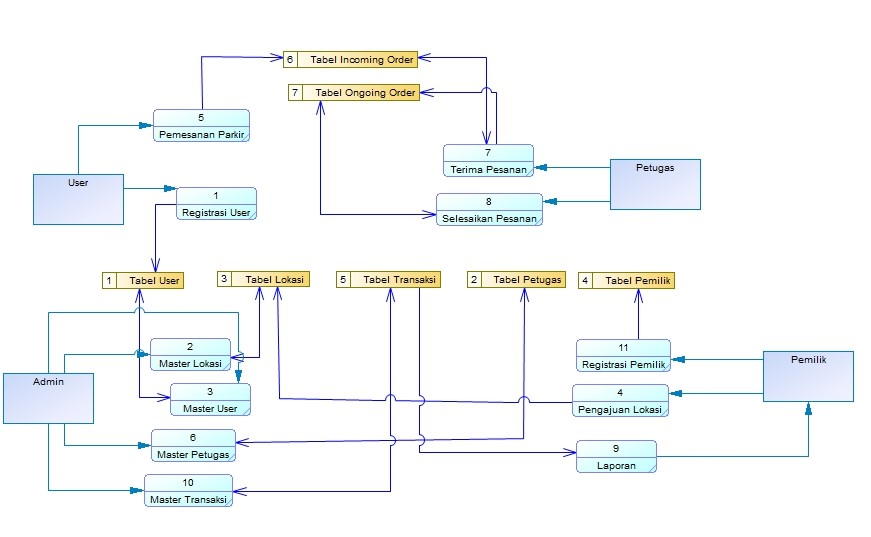
* Data Flow Diagram (DFD) Level 0 atau sering disebut Context Diagram (CD) merupakan level tertinggi dari suatu Data Flow Diagram (DFD).
* Data Flow Diagram (DFD) Level 1 merupakan hasil dekomposisi dari Data Flow Diagram (DFD) Level 0.
* Data Flow Diagram (DFD) Level 2 merupakan hasil dekomposisi dari proses-proses yang ada di Data Flow Diagram (DFD) Level 1.

Tingkatan level yang digunakan pada sistem aplikasi “ParkirIn” adalah DFD Level 0 (Context Diagram) dan DFD Level 1. Berikut adalah data flow diagram level 0 dari aplikasi “ParkirIn” yang telah dikembangkan.



Gambar 4.1 Data Flow Diagram Level 0 (Context Diagram)

Pada Gambar 4.1, telah ditunjukkan alur data dari sistem aplikasi “ParkirIn” yang sudah dikembangkan. Keseluruhan aktivitas dari setiap entity mengacu hanya pada 1 (satu) sistem. Tidak ada detail dari aktivitas yang dilakukan oleh setiap entity. Detail dari proses-proses yang dilakukan oleh user, petugas dan pemilik lahan serta admin akan tertera pada diagram tingkat selanjutnya yaitu data flow diagram level 1. Berikut adalah data flow diagram level 1 dari aplikasi “ParkirIn” yang telah dikembangkan.



Gambar 4.2 Data Flow Diagram Level 1

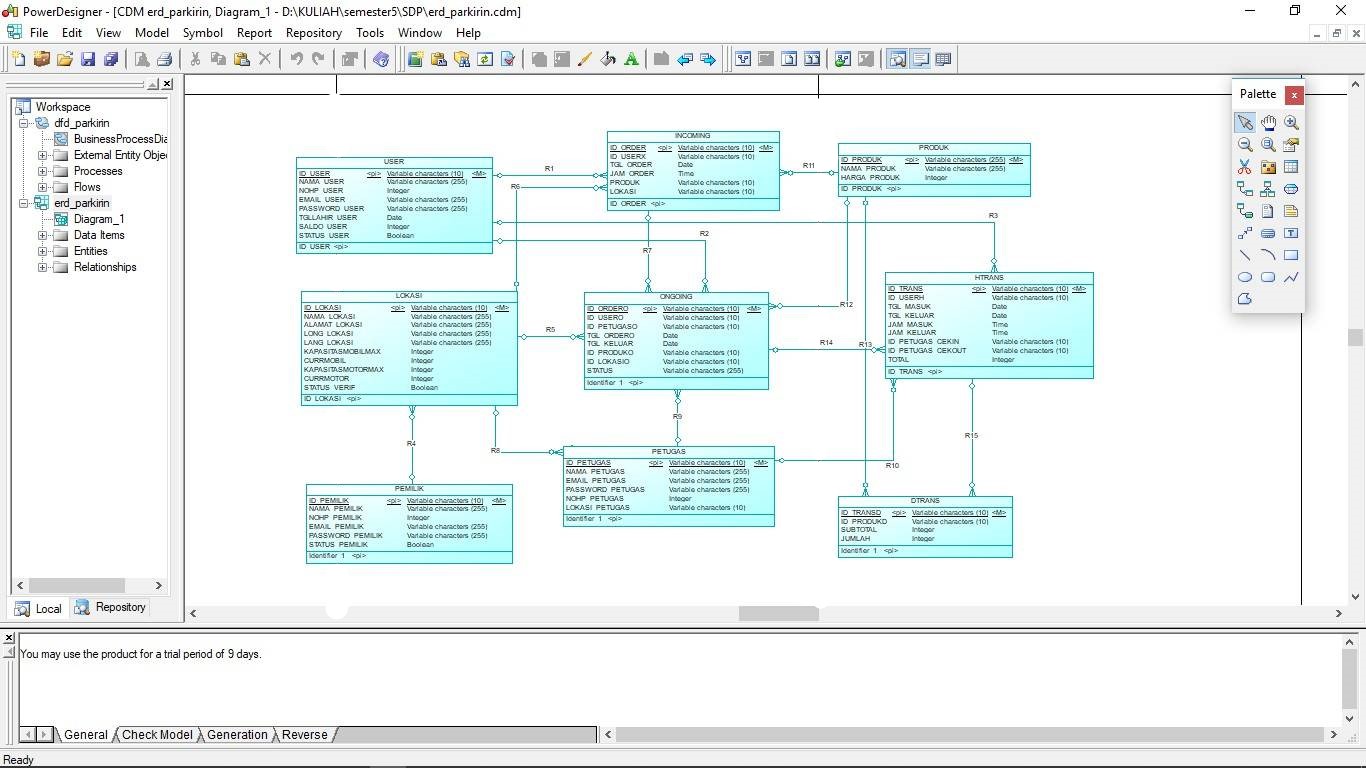
**4.2 Desain Arsitektur Sistem**

Desain arsitektur sistem merupakan proses mendeskripsikan dan mendefinisikan bagaimana suatu sistem aplikasi dibentuk dan diorganisasikan ke dalam komponen-komponen yang akan menbentuk aplikasi tersebut. Beberapa kriteria yang dipilih dalam pengembangan aplikasi “ParkirIn” antara lain :

1. Usable
2. Correct
3. Reliable
4. Flexible
5. Efficient
6. Maintainable

**4.3 Class Diagram**

Class Diagram adalah diagram yang menunjukan class-class yang ada dari sebuah sistem dan hubungannya secara logika. Class diagram menggambarkan struktur statis dari jalannya sebuah sistem. Karena itu class diagram merupakan tulang punggung atau kekuatan dasar dari hampir setiap metode berorientasi objek termasuk Unified Modeling Language (UML). Berikut adalah class diagram dari aplikasi “ParkirIn”



Gambar 4.1 Class Diagram Aplikasi "ParkirIn"

**4.4 Desain Tabel Database**

Berikut ini merupakan desain dari table-tabel yang akan digunakan dalam aplikasi “ParkirIn” ini. Tabel–tabel ini merupakan realisasi dari alur sistem yang telah dirancang. Tabel akan digunakan untuk menyimpan data–data dan informasi terkait dengan atribut-atribut dan transaksi operasional yang terjadi di dalam klinik. Seluruh tabel akan dijelaskan dalam pembahasan berikut.

* + 1. **Tabel User**

Tabel User menyimpan data-data user dari aplikasi “ParkirIn” ini. Tabel ini memiliki beberapa atribut antara lain ID\_USER, NAMA\_USER, EMAIL\_USER, PASSWORD\_USER, NOHP\_USER, SALDO\_USER, TGLLAHIR\_USER dan JK\_USER. Sebagian dari field tersebut tidak boleh dibiarkan kosong saat pertama kali mendaftar sebagai user.

Field ID\_USER berfungsi sebagai *identifier* untuk setiap user yang mendaftar di aplikasi ini. Nilai dari ID\_USER ini akan digenerate secara otomatis oleh sistem dengan hasil random. Field NAMA\_USER berfungsi untuk menyimpan nama dari user. Field EMAIL\_USER berfungsi untuk menyimpan alamat email user. Email digunakan untuk melakukan aktivitas login. Field PASSWORD\_USER berfungsi untuk menyimpan kata sandi dari akun yang dibuat user. Field NOHP\_USER berfungsi untuk menyimpan kontak dari user. Field SALDO\_USER berfungsi untuk menyimpan jumlah saldo yang dimiliki user dalam akun yang terdaftar. Field TGLLAHIR\_USER berfungsi untuk mengetahui tanggal lahir dari user. Field JK\_USER berfungsi untuk mengetahui jenis kelamin dari user. Berikut adalah rincian dan struktur dari tabel user.

**Tabel 4.1**

**Tabel User**

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD** | **TYPE** | **SIZE** | **DETAIL** |
| ID\_USER | VARCHAR | 50 | Primary Key (auto generate) |
| NAMA\_USER | VARCHAR | 100 | Not Null |
| EMAIL\_USER | VARCHAR | 100 | Not Null |
| PASSWORD\_USER | VARCHAR | 100 | Not Null |
| NOHP\_USER | NUMBER | 100 | Not Null |
| SALDO\_USER | NUMBER | 100 |  |
| TGLLAHIR\_USER | DATE |  | Not Null |
| JK\_USER | VARCHAR | 10 |  |

* + 1. **Tabel Pemilik**

Tabel Pemilik menyimpan data-data pemilik lahan dari aplikasi “ParkirIn” ini. Tabel ini memiliki beberapa atribut antara lain ID\_PEMILIK, NAMA\_ PEMILIK, EMAIL\_PEMILIK, PASSWORD\_PEMILIK, NOHP\_PEMILIK, dan STATUS\_KOMISI. Field-field ini punya fungsi yang sama seperti field pada user. Terkecuali untuk STATUS\_KOMISI berfungsi untuk menentukan durasi pembagian komisi yang dipilih oleh pemilik lahan. Berikut adalah rincian dan struktur dari tabel pemilik.

**Tabel 4.2**

**Tabel Pemilik**

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD** | **TYPE** | **SIZE** | **DETAIL** |
| ID\_PEMILIK | VARCHAR | 50 | Primary Key (auto generate) |
| NAMA\_PEMILIK | VARCHAR | 100 | Not Null |
| EMAIL\_PEMILIK | VARCHAR | 100 | Not Null |
| PASSWORD\_PEMILIK | VARCHAR | 100 | Not Null |
| NOHP\_PEMILIK | NUMBER | 50 | Not Null |
| STATUS\_KOMISI | VARCHAR | 50 |  |

* + 1. **Tabel Lokasi**

Tabel Lokasi menyimpan data-data dari lahan lokasi parkir yang ada pada aplikasi “ParkirIn” ini. Tabel ini memiliki beberapa atribut antara lain ID\_LOKASI, NAMA\_LOKASI, ALAMAT\_LOKASI, LONGTITUDE, LATITUDE, ID\_PEMILIK, KAPASITASMOBIL\_LOKASI, KAPASITASMOTOR\_LOKASI, CURRAVAILMOBIL, CURRAVAILMOTOR, JENIS\_LOKASI, FOTO\_LOKASI dan STATUS\_VERIFIKASI.

Field LONGTITUDE menyimpan titik lokasi pada garis bujur dan LANGTITUDE menyimpan titik lokasi pada garis lintang. Field ID\_PEMILIK berfungsi untuk menunjukkan kepemilikan suatu lahan. KAPASITASMOBIL\_LOKASI dan KAPASITASMOTOR\_LOKASI berfungsi untuk mengetahui jumlah maksimal lahan tersebut dapat menampung kendaraan. Field CURRAVAILMOBIL dan CURRAVAILMOTOR berguna untuk mengetahui jumlah lahan parkir yang tersedia pada suatu waktu. JENIS\_LOKASI berfungsi untuk mengetahui apakah suatu lokasi dapat menjadi tempat titip kendaraan ataupun hanya tempat parkir kendaraan. STATUS\_VERIFIKASI berfungsi untuk mengetahui apakah suatu lokasi sudah disetujui oleh pihak “ParkirIn” untuk menjadi lahan parkir. Berikut adalah rincian dan struktur dari tabel lokasi

**Tabel 4.3**

**Tabel Lokasi**

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD** | **TYPE** | **SIZE** | **DETAIL** |
| ID\_LOKASI | VARCHAR | 50 | Primary Key (auto generate) |
| NAMA\_ LOKASI | VARCHAR | 100 | Not Null |
| ALAMAT\_ LOKASI | VARCHAR | 100 | Not Null |
| LONGTITUDE | VARCHAR | 100 | Not Null |
| LATITUDE | VARCHAR | 100 | Not Null |
| ID\_PEMILIK | VARCHAR | 50 | Foreign key (to tabel pemilik) |

**Tabel 4.3 (lanjutan)**

**Tabel Lokasi**

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD** | **TYPE** | **SIZE** | **DETAIL** |
| KAPASITASMOBIL\_LOKASI | NUMBER | 50 | NOT NULL |
| KAPASITASMOTOR\_LOKASI | NUMBER | 50 | NOT NULL |
| CURRAVAILMOBIL | NUMBER | 50 | NOT NULL |
| CURRAVAILMOTOR | NUMBER | 50 | NOT NULL |
| JENIS\_LOKASI | VARCHAR | 50 |  |
| FOTO\_LOKASI | IMAGE |  |  |
| STATUS\_VERIFIKASI | NUMBER | 1 | 1 = aktif  0 = tidak aktif |

* + 1. **Tabel Petugas**

Tabel petugas menyimpan data-data dari petugas yang bekerja pada aplikasi “ParkirIn”. Tabel ini memiliki beberapa atribut antara lain ID\_PETUGAS, NAMA\_PETUGAS, EMAIL\_PETUGAS, PASSWORD\_PETUGAS, NOHP\_PETUGAS dan LOKASI\_PETUGAS. Field LOKASI\_PETUGAS berfungsi untuk mengetahui di mana petugas tersebut ditempatkan. Berikut adalah rincian dan struktur dari tabel petugas.

**Tabel 4.4**

**Tabel Petugas**

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD** | **TYPE** | **SIZE** | **DETAIL** |
| ID\_PETUGAS | VARCHAR | 50 | Primary Key (auto generate) |
| NAMA\_ PETUGAS | VARCHAR | 100 | Not Null |
| EMAIL\_ PETUGAS | VARCHAR | 100 | Not Null |
| PASSWORD\_ PETUGAS | VARCHAR | 100 | Not Null |
| NOHP\_ PETUGAS | NUMBER | 50 | Not Null |
| LOKASI\_ PETUGAS | VARCHAR | 50 | Foreign Key (to tabel lokasi) |

* + 1. **Tabel Produk**

Tabel Produk menyimpan data-data dari petugas yang bekerja pada aplikasi “ParkirIn”. Tabel ini memiliki beberapa atribut antara lain ID\_PRODUK, NAMA\_PRODUK dan HARGA\_PRODUK. Field HARGA\_PRODUK berfungsi untuk menetapkan harga dari produk yang ditawarkan kepada user. Berikut adalah rincian dan struktur dari tabel produk.

**Tabel 4.5**

**Tabel Petugas**

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD** | **TYPE** | **SIZE** | **DETAIL** |
| ID\_PRODUK | VARCHAR | 50 | Primary Key (auto generate) |
| NAMA\_PRODUK | VARCHAR | 100 | Not Null |
| HARGA\_PRODUK | NUMBER | 50 | Not Null |

* + 1. **Tabel Incoming**

Tabel Incoming menyimpan data-data dari incoming order pada aplikasi “ParkirIn”. Tabel ini memiliki beberapa atribut antara lain ID\_ORDER, ID\_USER, TGL\_PESAN, JAM\_PESAN, PRODUK, dan LOKASI. Berikut adalah rincian dan struktur dari tabel incoming.

**Tabel 4.6**

**Tabel Incoming**

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD** | **TYPE** | **SIZE** | **DETAIL** |
| ID\_ORDER | VARCHAR | 50 | Primary Key (auto generate) |
| ID\_USER | VARCHAR | 100 | Foreign key (to tabel user) |
| TGL\_PESAN | DATE |  | Not Null |
| JAM\_PESAN | TIME |  | Not Null |
| PRODUK | VARCHAR | 50 | Foreign key (to tabel produk) |
| LOKASI | VARCHAR | 50 | Foreign key (to tabel lokasi) |

* + 1. **Tabel Ongoing**

Tabel Ongoing menyimpan data-data dari order yang telah diterima oleh petugas pada aplikasi “ParkirIn”. Tabel ini memiliki beberapa atribut antara lain ID\_ORDER, ID\_USER, ID\_PETUGAS TGL\_PESAN, JAM\_PESAN, PRODUK, dan LOKASI. Berikut adalah rincian dan struktur dari tabel produk.

**Tabel 4.7**

**Tabel Ongoing**

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD** | **TYPE** | **SIZE** | **DETAIL** |
| ID\_ORDER | VARCHAR | 50 | Primary Key (FK to tabel incoming) |
| ID\_USER | VARCHAR | 50 | Foreign key (to tabel user) |
| ID\_PETUGAS | VARCHAR | 50 | Foreign key (to tabel petugas) |
| TGL\_PESAN | DATE |  | Not Null |
| JAM\_MASUK | TIME |  | Not Null |
| PRODUK | VARCHAR | 50 | Foreign key (to tabel produk) |
| LOKASI | VARCHAR | 50 | Foreign key (to tabel lokasi) |

* + 1. **Tabel Htrans**

Tabel Htrans menyimpan data-data transaksi yang telah selesai pada sistem aplikasi “ParkirIn”. Tabel ini memiliki beberapa atribut antara lain ID\_TRANSAKSI, ID\_USER, ID\_PETUGAS\_CEKIN, ID\_PETUGAS\_CEKOUT, TGL\_MASUK, TGL\_KELUAR, JAM\_MASUK, JAM\_KELUAR dan TOTAL. Setiap transaksi akan diselesaikan oleh petugas dan petugas yang melakukan check in ataupun check out bisa saja berbeda. Berikut adalah rincian dan struktur dari tabel htrans.

**Tabel 4.8**

**Tabel HTRANS**

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD** | **TYPE** | **SIZE** | **DETAIL** |
| ID\_TRANSAKSI | VARCHAR | 50 | Primary Key (FK to tabel ongoing) |
| ID\_USER | VARCHAR | 50 | Foreign key (to tabel user) |
| TGL\_MASUK | DATE |  | Not Null |
| TGL\_KELUAR | DATE |  | Not Null |
| JAM\_MASUK | TIME |  | Not Null |
| JAM\_KELUAR | TIME |  | Not Null |
| ID\_PETUGAS\_CEKIN | VARCHAR | 50 | Foreign key(tabel petugas) |
| ID\_PETUGAS\_CEKOUT | VARCHAR | 50 | Foreign key(tabel petugas) |
| TOTAL | NUMBER | 50 |  |

* + 1. **Tabel Dtrans**

Tabel Dtrans menyimpan detail dari transaksi yang telah selesai pada sistem aplikasi “ParkirIn”. Tabel ini memiliki beberapa atribut antara lain ID\_TRANSAKSI, ID\_PRODUK, SUBTOTAL dan JUMLAH. Berikut adalah rincian dan struktur dari tabel dtrans

**Tabel 4.9**

**Tabel DTRANS**

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD** | **TYPE** | **SIZE** | **DETAIL** |
| ID\_TRANSAKSI | VARCHAR | 50 | Primary Key (FK to tabel htrans) |
| ID\_PRODUK | VARCHAR | 50 | Foreign key (to tabel produk) |
| SUBTOTAL | NUMBER | 50 | Not Null |
| JUMLAH | NUMBER | 50 | Not Null |

**4.5 Desain Interface**

Pada bagian ini akan dibahas mengenai desain-desain interface pada aplikasi “ParkirIn”.

**4.5.1 Desain Aplikasi User**

Aplikasi “Cari Parkir

**4.5.2 Desain Aplikasi Owner**

Aplikasi “Cari Parkir”

**4.5.3 Desain Aplikasi Operator**

Aplikasi “Cari Parkir”

**4.5.4 Desain Aplikasi Admin**

Aplikasi “Cari Parkir”

# BAB V IMPLEMENTASI

Pada bab ini akan dijelaskan tentang layouting dari sistem aplikasi “ParkirIn”. Pembahasan pada bab ini adalah seputar tampilan-tampilan yang dipilih pada setiap page pada aplikasi “ParkirIn”. Aplikasi ini dikembangkan dengan tampilan yang seminimalis mungkin sehingga dapat meminimalisir user yang kebingungan saat menggunakan aplikasi ini. Aplikasi “ParkirIn” ini tentunya dibangun dengan harapan dapat memenuhi kekurangan-kekurangan dari aplikasi serupa yang sudah ada sebelumnya.

**5.1 Layouting**

Aplikasi “Cari Parkir” sebagai pembanding aplikasi “Parkirin” sudah memiliki fitur-fitur yang membantu user antara lain untuk mencari tempat-tempat parkir dan lahan yang tersedia.