# BAB II LANDASAN TEORI

## Sistem

Menurut O’Brien (2011), sistem merupakan sebuah komponen yang saling berkaitan dan bekerja sama dalam mencapai seperangkat tujuan dengan menerima input dan menghasilkan output secara terorganisasi melalui proses transformasi.

## Parkir

Menurut Departemen Perhubungan (1996), parkir adalah tempat pemberhentian kendaraan dalam jangka waktu pendek atau lama, sesuai dengan kebutuhan pengendara. Parkir merupakan salah satu unsur prasarana transportasi yang tidak terpisahkan dari sistem jaringan transportasi, sehingga pengaturan parkir akan mempengaruhi kinerja suatu jaringan, terutama jaringan jalan raya. Daerah perkotaan dengan kepadatan penduduk dan tingkat ekonomi yang tinggi mengakibatkan tingkat kepemilikan kendaraan pribadi yang tinggi pula. Apabila kondisi ini didukung dengan kebijakan pemerintah dalam manajemen lalu lintas yang tidak membatasi penggunaan mobil pribadi, maka akan mendukung pelaku pergerakan untuk selalu menggunakan kendaraan pribadi. Hal ini akan menimbulkan kebutuhan lahan parkir yang besar pada zona tarikan sebagai contoh pada daerah pusat bisnis (Abbas, 2013).

Tidak semua pengembang pusat bisnis mampu menyediakan lahan parkir yang mencukupi, sehingga badan jalan yang berada di sekitarnya digunakan untuk lahan parkir. Apabila badan jalan tersebut dilalui lalu lintas dalam jumlah yang cukup besar maka bisa dipastikan bahwa parkir di badan jalan akan menimbulkan permasalahan lalu lintas (kecepatan menurun dan waktu tempuh meningkat). Timbulnya permasalahan parkir di kota-kota besar menuntut para ahli transportasi untuk betul-betul memahami parkir. Konsep dan karakteristik parkir, analisis kebutuhan parkir, perencanaan geometrik lahan parkir, dan kebijakan parkir merupakan materi bisa diimplementasikan untuk menangani permasalahan parkir. Berikut beberapa klasifikasi parkir (Pramudita, 2012):

### Cara dan jenis parkir

* + - 1. Menurut Penempatannya

Parkir di tepi jalan *(on-street parking)*.

Yakni parkir dengan menggunakan badan jalan sebagai tempat parkir.

Kerugian:

1. Mengganggu lalu lintas
2. Mengurangi kapasitas jalan karena adanya pengurangan lebar lajur lalu lintas
3. Meningkatkan kemungkinan terjadinya kecelakaan

Keuntungan:

1. Murah tanpa investasi tambahan
2. Bagi pengguna tempat parkir bisa lebih dekat dan mudah

Posisi parkir:

1. Sejajar dengan sumbu jalan
2. Tegak lurus sumbu jalan
3. Membuat sudut dengan sumbu jalan

Parkir di luar badan jalan *(off-street parking*).

Yakni parkir kendaraan diluar badan jalan bisa di halaman gedung perkantoran, supermarket, atau pada taman parkir. Perparkiran yang ideal adalah parkir di luar badan jalan berupa fasilitas peralatan (taman) parkir atau bangunan (gedung) parkir. Di pusat kegiatan kota yang sangat sulit memperoleh lahan yang cukup luas, fasilitas yang sesuai dengan gedung parkir yang dapat dibangun bertingkat sesuai dengan kebutuhan. Taman parkir maupun gedung parkir memerlukan biaya investasiyang cukup besar, namun pengembaliannya dapat diharapkan tidak terlalu lamadan bisa menjadi lahan usaha.

Keuntungan:

1. Tidak mengganggu lalu lintas
2. Faktor keamanan lebih tinggi

Kerugian:

1. Perlu biaya investasi awal yang besar.
2. Bagi pengguna dirasakan kurang praktis, apalagi jika kepentingannya hanya sebentar saja.
3. Menurut Jenis Kendaraan
4. Kendaraan tidak bermesin (sepeda)
5. Sepeda motor
6. Mobil

### Kebijakan Parkir

Beberapa kebijakan parkir yang diterapkan diberbagai negara antara lain:

Kebijakan tarif parkir yang ditetapkan berdasarkan lokasi dan waktu semakin dekat dengan pusat kegiatan/ kota tarif lebih tinggi, demikian juga semakin lama semakin tinggi. Kebijakan ini diarahkan untuk mengendalikan jumlah pemarkir di pusat kota/ pusat kegiatan dan mendorong penggunaan angkutan umum.

Kebijakan pembatasan ruang parkir, terutama di daerah pusat kota ataupun pusat kegiatan. Kebijakan ini biasanya dilakukan pada parkir di pinggir jalan yang tujuan utamanya untuk melancarkan arus lalu lintas, serta pembatasan ruang parkir di luar jalan yang dilakukan melalui IMB (Ijin Mendirikan Bangunan).

Kebijakan penegakan hukum yang tegas terhadap pelanggar ketentuandilarang parkir dan dilarang berhenti serta pemarkir di luar tempat yang ditentukan untuk itu. Bentuk penegakan hukum dapat dilakukan melaluipenilangan ataupun dengan gembok roda

### Hal yang harus diperhatikan dalam pengelolaan parkir

Parkir di badan jalan merupakan masalah utama yang menyebabkan kemacetan, karena sudah pasti mengurangi kapasitas ruas jalan dan mengganggu laju kendaraan yang lewat. Selain parkir di badan jalan, masih ada parkir di luar badan jalan yang juga harus diperhatikan, baik pengaturan parkir maupun penentuan bentuk SRP (Satuan Ruang Parkir) yang tepat, sehingga dengan demikian perencana tempat parkir hendaknya memahami dan mampu mendesaintempat parkir dengan nyaman untuk kendaraan.

* + - 1. Standar Fasilitas Parkir

Standar fasilitas parkir adalah suatu pedoman untuk menentukan suatuukuran petak parkir menurut berbagai bentuk penyediaannya. Penggunaan standar fasilitas parkir diperlukan karena tidak terdapatnya marka dari petak parkir pada daerah studi. Sehingga untuk menentukan jumlah petak parkir yang terdapat pada daerah studi didasarkan pada standar.

* + - 1. Standar Kebutuhan Parkir

Standar kebutuhan parkir adalah suatu ukuran yang dapat dipergunakanuntuk jumlah kebutuhan parkir kendaraan berdasarkan fasilitas dan fungsi dari tata guna lahan. Kebutuhan parkir untuk setiap tata guna lahan berbeda- beda, begitu pula untuk setiap negara bahkan daerah mempunyai standar yang berbeda- beda

## *Artificial Intelligence*

*Artificial Intelligence (*Kecerdasan Buatan)merupakan hal inti yang ingin diterapkan dalam penelitian kali ini. Menurut Jaya et al (2018), *Artificial Intelligence* adalah salah satu bagian ilmu komputer yang membuat agar mesin (komputer) dapat melakukan pekerjaan seperti dan sebaik yang dilakukan oleh manusia. Penggunaan komputer terus berkembang, dari yang hanya sebagai alat hitung menjadi suatu alat yang mendominasi hidup manusia. Komputer pun terus diharapkan untuk bisa mengerjakan segala sesuatu yang bisa dikerjakan oleh manusia.

Selain itu, Jaya juga membandingkan Kecerdasan Buatan dengan Kecerdasan Alami. Menurutnya, Kecerdasan Buatan memiliki beberapa keuntungan sebagai berikut:

1. Kecerdasan buatan lebih bersifat permanen. Kecerdasan alami akan cepat mengalami perubahan. Hal ini dimungkinkan karena sifat manusia yang pelupa. Kecerdasan buatan tidak akan berubah sepanjang sistem komputer & komputer tidak mengubahnya.
2. Kecerdasan buatan lebih mudah diduplikasi & disebarkan. Mentransfer pengetahuan manusia dari orang ke orang lain membutuhkan proses yang sangat lama; dan juga suatu keahlian itu tidak akan pernah dapat diduplikasi dengan lengkap. Oleh karena itu, jika pengetahuan terletak pada pada suatu sistem komputer, pengetahuan tersebut dapat disalin dari komputer tersebut dan dapat dipindahkan dengan mudah ke komputer yang lain.
3. Kecerdasan buatan lebih murah dibanding dengan kecerdasan alami. Menyediakan layanan komputer akan lebih mudah dan lebi murah dibandingkan harus mendatangkan seseeorang untuk mengerjakan sejumlah pekerjaan dalam jangka waktu yang sangat lama.
4. Kecerdasan buatan bersifat konsisten. Hal ini disebabkan kaarena kecerdasan buatan adalah bagian dari teknologi komputer. Sedangkan kecerdasan alami akan senantiasa berubah-ubah.
5. Kecerdasan buatan dapat didokumentasi. Keputusan yang dibuat oleh komputer dapat didokumentasi dengan mudah dengan cara melacak setiap aktifitas dari sistem tersebut. Kecerdasan alami sangat sulit untk direproduksi.
6. Kecerdasan buatan dapat mengerjakan pekerjaan lebih cepat dibanding dengan kecerdasan alami.

Sedangkan, menurut Poole (2017), *Artificial Intelligence* adalah bidang yang mempelajari tahap sintesis dan analisis dari sebuah agen komputasional yang bertindak secara cerdas. “Agen” di sini dapat diartikan sebagai sesuatu yang bertindak di lingkungannya, agen bisa termasuk anjing, termostat, pesawat, robot, manusia, perusahaan, dan negara.

Sebuah “Agen” dapat dikatakan cerdas bila memenuhi beberapa kriteria yang dapat kita lihat dari perlakuannya, yaitu:

1. Apabila yang dilakukan cocok untuk keadaan dan tujuannya, dengan mempertimbangkan konsekuensi jangka pendek dan Panjang dari tindakannya
2. Apabila agen tersebut fleksibel dengan perubahan lingkungan dan tujuan
3. Apabila agen tersebut belajar dari pengalaman
4. Apabila agen tersebut dapat mengambil pilihan yang tepat dari batasan yang dimiliki

## *Face Recognition*

*Face Recognition* merupakan suatu bentuk penerapan dari *Artificial Intelligence.* Menurut *Electronic Frontier Foundation, Face Recognition* merupakan sebuah metode yang digunakan dalam mengidentifikasi dan memverifikasi identitas dari seorang individu menggunakan wajah mereka. *Face Recognition* dapat digunakan untuk mengidentifikasi orang dari foto, video, atau secara *real-time.* (Tutoials Point, 2015)

*Face Recognition* juga merupakan suatu pengenalan biometrik dan sistem autentikasi yang banyak digunakan (Priyanka & Yashpal, 2015). Hal yang menjadi tantangan adalah mengenai akurasi dari gambar yang diambil secara *real-time*. Tantangan yang ada pun semakin bertambah, seperti gambar yang terpotong, *noise,* distorsi, dan lain- lain. Ada juga gambar yang mengalami proses *editing* yang membuat proses *face recognition* menjadi perlu dikembangkan. Fitur *SIFT (Scale Invariant Feature Transform)* menjadi fitur yang dibutuhkan dalam meningkatkan kemungkinan keberhasilan dalam mengenali gambar yang sudah disunting (*morphed image).* Proses ini akan membagi gambar yang utuh menjadi beberapa bagian berdasarkan fitur unik yang dimiliki, seperti bagian mata, hidung, mulut, dll. Sehingga, pada tahap pencocokan gambar nantinya, tiap bagian akan dicocokkan satu persatu untuk meningkatkan kemungkinan keberhasilan dalam proses pengenalan wajah.

Ada 4 tahap yang umumnya dilakukan dalam proses *face recognition,* yaitu:

1. *Filtration*

Tahap yang mengatasi masalah-masalah umum pada gambar yang diambil secara langsung. Tahap ini akan melakukan penyaringan/ filtrasi terhadap tiap *noise* yang ada, serta masalah-masalah lainnya. Hal ini dilakukan sehingga hasil gambar dari filtrasi ini dapat diproses secara lebih maksimal.

1. *Segmentation*

Tahap kedua adalah segmentasi. Tahap ini digunakan untuk mengidentifikasi dan membedakan tiap fitur yang ada pada gambar wajah, seperti mata, hidung, bibir, dll. Pendekatan yang dilakukan untuk membedakan tiap fitur di wajah adalah dengan menganalisis posisi tiap fitur di wajah.

1. *SIFT Feature Extraction*

Tahap ini dilakukan untuk mengekstrak tiap fitur dari wajah yang telah ditemukan sebelumnya.

1. *Distance based Recognition*

Pada tahap ini, proses pengenalan akan dilakukan pada tiap fitur secara satu persatu dengan menggunakan analisis berbasis jarak. Pengenalan akan dilakukan pada tiap bagian, yang nantinya akan digabungkan untuk memperoleh hasil kecocokan total dari pengenalan wajah yang dilakukan.

## *Computer Vision*

Kita sebagai manusia, dapat dengan mudah melihat dan memandang suatu objek atau benda. Kita mampu melihat objek secara 3D dan mengetahui bentuk serta karakteristik dari hal yang kita lihat. Sama halnya saat melihat orang lain, kita mampu mengetahui dan membedakan orang yang satu dengan yang lain. Namun, tidak sama halnya dengan cara komputer melihat dan memroses suatu objek.

Para ilmuwan mulai mengembangkan suatu teknik untuk memperoleh data dan penampilan dari objek 3 dimensi secara imajiner, sehingga dapat dimengerti oleh komputer. Teknik ini pun dikenal dengan sebutan *computer vision.* Dengan *computer vision,* kita telah mempunyai teknik dan cara yang terjamin dalam memroses suatu objek ke komputer. Kita pun juga mampu membedakan tiap orang dalam suatu gambar dengan menggunakan kombinasi dari wajah, baju, gaya rambut tentunya dengan menggunakan metode *recognition* terkait. Namun, *computer vision* tetap tidak semaksimal manusia. Hal ini dikarenakan bahwa ‘penglihatan’ merupakan sebuah masalah yang kontradiktif, karena kita harus memperoleh informasi yang tidak kita ketahui dengan hanya bermodal data yang (terkadang) tidak mencukupi dalam memecahkan masalah.

Dikaitkan dengan *face recognition, computer vision* dalam kategori tersebut dapat dikatakan sebagai yang paling berhasil. Komputer yang awalnya tidak dapat membedakan orang yang satu dengan yang lain, menjadi mampu membedakan tiap individu. Komputer dapat memroses foto tiap orang (*image processing)* dan membuat pola tersendiri dari tiap foto yang diproses. Hal ini pun membuat tiap wajah memiliki ‘identitas’ tersendiri di ‘mata komputer’ (Szeliski, 2010).

## *Kairos Face Library*

*Kairos* adalah perusahaan yang bergerak dalam bidang *artificial intelligence* (kecerdasan buatan) termasuk juga seperti *image processing, computer vision, machine learning* (pembelajaran mesin), *face recognition* (sistem pengenalan wajah) dan lain sebagainya (Kairos, 2012).

Kairos merupakan satu dari sekian banyak perusahaan yang bergerak dalam bidang Teknologi Informasi yang menyediakan fasilitas *API (Application Programming Interface)* dan *SDK (Software Development Kit*) ataupun dapat juga disebut sebagai  untuk bidang deteksi wajah (*face detection*) dan pengenalan wajah (*face recognition)* yang dapat diintegrasikan oleh pengembang dalam membuat suatu proyek aplikasi ataupun proyek penelitian dengan mudah. Pada prinsipnya, fitur dari kairos ini adalah berbayar jika fitur yang diinginkan pengembang lebih dari sekedar face detection dan face recognition. Tetapi disamping itu, kairos juga menyediakan layanan integrasi oleh pengembang secara gratis dan open source. Pada penerapannya, library Kairos memiliki fungsi yang sama dengan library-library untuk face recognition lainnya seperti openCV, emguCV atau library lainnya yang dapat mengenali wajah pada suatu objek gambar sehingga memudahkan untuk penggunaannya dalam integrasi aplikasi ataupun penelitian yang berkaitan dengan face recognition.

Hal yang menjadi perbedaan mendasar dari library Kairos dibandingkan dengan library lainnya yaitu library ini berfungsi secara online (koneksi jaringan internet), sedangkan library lainnya seperti openCV masih beroperasi dengan offline (tanpa ada jaringan internet).

Dari segi hasil dari fitur yang dimiliki, penerapan library kairos pada face recognition juga sangat lengkap jika dibandingkan dengan library face recognition lainnya seperti Amazon, Goggle API, Microsoft Face API, OpenCV, IBM Watson, Face.com API, Luxand API, Face++ API, Lambda Labs Face API dan lain sebagainya. Hal tersebut dikarenakan didalam library ini juga memuat fitur-fitur yang lengkap didalam sistem kerjanya seperti:

* *Face detection.*
* *Face identification.*
* *Face verification.*
* *Emotion analysis (joy, surprise, sadness, fear, anger, and disgust).*
* *Demographics (age, gender, ethnicity, attention, dwell, glances, blinks, facial features, glasses).*
* *Multi-face detection and tracking.*

Dan semua fitur-fitur tersebut tergabung dalam satu kesatuan sistem library kairos. Sehingga para pengembang (developer) dapat merasakan manfaat dari integrasi proyek pada bidang face recognition, image processing dan computer vision. Selain itu, jika ditinjau dari segi penerapan keamanan data wajah ataupun image yang menjadi objek dapat dikatakan sangat aman. Perbandingan *Kairos Face Library* dengan *Library* serupa lainnya, dapat dilihat pada Tabel 2.1 berikut:

**Tabel 2.1**

**Perbandingan Performa Kairos dengan Library Lain**



Sumber: <http://www.kairos.com>

## Application Programming Interface (API)

*Application Programming Interface (API)* adalah antar muka program dari sistem yang dapat diakses melalui *method* dan *header* pada protokol HTTP yang standar. *API* dapat diakses dari berbagai macam HTTP *client* seperti *browser* dan perangkat *mobile*. *API* juga memiliki keuntungan karena menggunakan infrastruktur yang juga digunakan oleh *web* terutama untuk penggunaan *caching* dan *concurrency. API* akan menerima *request* dari *user* dan mengirimkan *response* dalam bentuk *JavaScript Object Notation (JSON)* dari *request* yang telah diproses tersebut.(Block et al., 2014).

## Android

Pengertian Android menurut Dimarzio (2017) adalah

“Android adalah sebuah Sistem Operasi *mobile* yang didasarkan atas sebuah versi termodifikasi dari Linux. Awalnya dibangun oleh sebuah *startup* dengan nama yang sama, Android, Inc. Pada 2005, sebagai bagian dari strategi Google untuk masuk ke dalam ruang *mobile*, Google membeli Android, Inc. dan mengambil alih kerja pengembangan mereka (dan juga tim pengembangan mereka). Google menginginkan Android OS untuk terbuka dan gratis, oleh karena itu sebagian besar kode Android dirilis dalam lisensi sumber terbuka Apache. Dengan begitu setiap orang yang ingin menggunakan Android dapat melakukannya dengan mengunduh *source code* penuh Android.”

Keuntungan utama dari menggunakan Android adalah hal tersebut menawarkan pendekatan yang sudah merupakan kesatuan dalam pengembangan aplikasi. Pengembang aplikasi hanya perlu untuk mengembangkan untuk Android pada umumnya, lalu kemudian aplikasi mereka seharusnya dapat berjalan pada berbagai jenis perangkat yang berbeda, selama perangkat tersebut menggunakan sistem operasi Android.

Android sendiri telah berkembang lama dengan menghasilkan berbagai versi atau edisi yang berbeda-beda. Versi Android yang mulai digunakan banyak orang, adalah sejak versi *Gingerbread (2.3).* Berikut pemaparan lebih jelasnya:

**Tabel 2.2**

**Perkembangan Versi Android**

|  |  |  |
| --- | --- | --- |
| **Versi** | **Nama Versi** | **Tanggal Rilis** |
| 2.3 | *Gingerbread* | 9 Februari 2011 |
| 4.0 | *Ice Cream Sandwich* | 19 Oktober 2011 |
| 4.1 | *Jelly Bean* | 9 Juli 2012 |
| 4.2 | *Jelly Bean* | 13 November 2012 |
| 4.3 | *Jelly Bean* | 24 Juli 2013 |
| 4.4 | *KitKat* | 31 Oktober 2013 |
| 5.0 | *Lollipop* | 3 November 2014 |
| 5.1 | *Lollipop* | 9 Maret 2015 |
| 6.0 | *Marshmallow* | 5 Oktober 2015 |
| 7.0 | *Nougat* | 22 Agustus 2016 |
| 7.1 | *Nougat* | 4 Oktober 2016 |
| 8.0 | *Oreo* | 21 Agustus 2017 |
| 8.1 | *Oreo* | 5 Desember 2017 |
| 9 | *Pie* | 6 Agustus 2018 |

Sumber: <https://en.wikipedia.org/wiki/Android_(operating_system)>

## *JavaScript*

*JavaScript* adalah bahasa pemrograman komputer yang dinamis. Javascript bersifat ringan dan kebanyakan digunakan sebagai bagian dari halaman web. Javascript adalah bahasa pemrograman yang telah diinterpretasi dengan kemampuan untuk mendukung pemgrograman berorientasi objek (TutorialsPoint, 2015).

*JavaScript* pertama kali dikenal sebagai LiveScript, tetapi Netscape mengubah namanya menjadi JavaScript, mungkin karena dihasilkan oleh bahasa Java.

Client – side JavaScript adalah bentuk yang paling umum dari bahasa pemrograman ini. Script tersebut harus dimasukan atau direfrensikan oleh dokumen HTML supaya kodenya bisa diinterpretasi oleh browser yang bersangkutan.

## React Native

Pengertian *React Native* menurut Eisenman (2017) yaitu sebuah *framework* JavaScript untuk menulis secara langsung, membangun *mobile application* untuk iOS dan Android. React Native berdasar pada React, sebuah perpustakaan JavaScript milik Facebook untuk membangun antarmuka pengguna, tetapi selain ditujukan untuk

penjelajah web, React Native ditujukan pada platform *mobile*. Dengan kata lain: pengembang web kini dapat menulis *mobile application* yang terlihat dan terasa sangat “asli,” segala kenyamanan dari sebuah perpustakaan yang telah kita ketahui dan cintai. Dan, karena sebagian besar dari kode yang akan ditulis dapat dibagikan antar platform, React Native membuat pengembangan untuk kedua Android dan iOS menjadi mudah.

React Native memiliki kesamaan dengan React untuk web, yaitu aplikasinya tertulis menggunakan gabungan dari JavaScript dan *XML-esque* markup, yang umumnya diketahui sebagai JSX. Lalu, di balik layar React Native menjembatani API pembangun asli dalam Objective-C (untuk iOS) atau Java (untuk Android). Lalu kemudian aplikasi akan dibangun menggunakan komponen User Interface (UI) asli *mobile,* bukan tampilan web, dan akan terlihat dan terasa seperti aplikasi *mobile* lainnnya. React Native juga menggunakan antarmuka JavaScript untuk platform API, sehingga aplikasi yang dibangun dengan React Native dapat menggunakan fitur platform seperti kamera telepon, atau lokasi pengguna. React Native saat ini mendukung iOS dan Android, dan memiliki potensi untuk berkembang di platform yang akan ada nantinya.

## *JavaScript Object Notation (JSON)*

*JSON (JavaScript Object Notation)* adalah format pertukaran data ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat oleh komputer. Format ini dibuat berdasarkan bagian dari bahasa pemograman JavaScript, Standar ECMA-262 Edisi ke-3-Desember 1999. JSON merupakan format teks yang tidak bergantung pada bahasa pemograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh programmer keluarga C termasuk C, C++, C#, Java, JavaScript, Perl, Phyton. Oleh karena sifat-sifat tersebut, menjadikan JSON ideal sebagai bahasa pertukaran data (Anonim, 2002).

JSON terbuat dari dua struktur:

* Kumpulan pasangan nama/nilai. Pada beberapa bahasa, hal ini dinyatakan sebagai objek (*object*), rekaman (*record*), struktur (*struct*), kamus (*dictionary*), tabel hash (*hash table*), daftar berkunci (*keyed list*), atau [*associative array*](http://en.wikipedia.org/wiki/Associative_array).
* Daftar nilai terurutkan (*an ordered list of values*). Pada kebanyakan bahasa, hal ini dinyatakan sebagai larik (*array*), vektor (*vector*), daftar (*list*), atau urutan (*sequence*).

## *Unified Modeling Language (UML)*

*Unified Modeling Language (UML)* adalah sebuah hasil penyatuan dari berbagai cara terbaik dalam penggunaan *modeling language*. *UML* memungkinkan kita untuk dapat merepresentasikan aspek dari sebuah sistem *software,* seperti *requirements,* struktur data, alur data, alur informasi, dan lain-lain dalam sebuah ruang lingkup yang menggunakan konsep berbasis objek (*object oriented).*

*UML* dapat digunakan secara konsisten terhadap seluruh proses pengembangan *software.* Pada tiap tahapannya, tiap objek dan konsep bahasa yang digunakan dapat saling berhubungan satu sama lain. Setiap model dari *UML* dapat saling merepresentasikan suatu siklus, tanpa harus mengubahnya ke model/ bahasa lain. Ada beberapa jenis *UML* Diagram yang sering digunakan:

1. *Class Diagram*

Diagram jenis ini merupakan diagram yang sering digunakan dalam sebuah sistem berbasis objek. Diagram ini digunakan untuk menggambarkan hubungan antara tiap elemen dalam sistem dalam tahap pengembangan *software.* Elemen-elemen tersebut dapat berupa sebuah kelas, objek, ataupun antarmuka*.*

1. *Use Case Diagram*

Diagram ini digunakan untuk lebih menggambarkan setiap fungsi yang harus dilakukan oleh sebuah sistem. Diagram ini juga menggambarkan tiap fungsi yang bisa dilakukan oleh tiap user berbeda serta hubungan di antaranya. Hal ini kemudian disebut dengan *actor (user)* dan *use case (*fungsi).

1. *Activity Diagram*

Diagram ini digunakan untuk menggambarkan tiap tahapan (aktivitas) yang ada dalam sebuah pengembangan sistem, dengan sudut pandang dari *user/ actor* yang bersangkutan. Diagram ini menggambarkan urutan tiap tahapan yang terjadi dalam sistem secara dinamis.

## Penelitian Terdahulu

Terdapat beberapa penelitian terdahulu yang berkaitan dengan penerapan *face recognition* pada sebuah sistem perpakiran. Beberapa penelitian terdahulu yang telah dilakukan tersebut, akan menjadi acuan dalam memperoleh informasi mengenai tema pemabahasan penelitian yang akan dilakukan.

Pada tahun 2012, Deananta Pramudia Putra, Bambang Hidayat, dan Ari Novianty melakukan penelitian mengenai penerapan *face recognition* untuk keamanan sistem parkir dengan judul *Pengenalan Wajah untuk Sistem Keamanan Parkir Menggunakan Metode Principal Component Analysis.* Dalam penelitian tersebut, Deananta beserta para rekan menggunakan Metode *Principal Component Analysis* serta *Learning Vector Quantization* dalam mendeteksi wajah si pemilik kendaraan, beserta plat kendaraan yang dimiliki. Dalam hal pengenalan wajah, akurasi keberhasilan yang dicapai sebesar 91.875%.

Pada tahun 2015, Semuil Tjiharjadi dan William Setiadarma melakukan penelitian mengenai penerapan *face recognition* sebagai fitur keamanan tambahan di tempat parkir dengan judul *Face Recognition for Additional Security at Parking Place*. Dalam penelitian tersebut, mereka menggunakan metode *Eigenface* dalam mengenali citra wajah. Hal tersebut dilakukan pada saat si pemilik kendaraan masuk dan kemudian dikonfirmasi pada saat si pemilik kendaraan keluar. Hal ini mereka lakukan untuk mengurangi jumlah kasus kehilangan kendaraan yang sering terjadi.