# BAB II

# LANDASAN TEORI

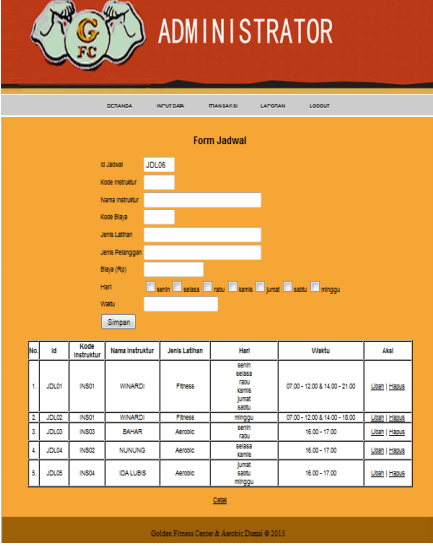
## Kajian Penelitian Terdahulu

Penelitian pertama yang dilakukan oleh Herry Erwanto dan Siti Monalisa (2016), dengan judul Rancang Bangun Sistem Informasi *Fitness* Berbasis *Web* dan Sms *Gateway* pada Rajawali *Fitness* Pekanbaru. Penelitian ini bertujuan untuk membantu mengumpulkan data proses manajemen yang dapat menjangkau seluruh keanggotaan pelanggan dan membantu menilai kinerja layanan untuk kenyamanan pelanggan. Sistem ini memudahkan bisnis untuk mempromosikan produk, mendapatkan *user* dan meningkatkan retensi *user*. Hasil dari sistem ini adalah sistem informasi fitness berbasis web dan SMS gateway. Sehingga, Rajawali *Fitness* dapat membantu dalam mengelola data pelanggan dan data fasilitas *fitness*. Hasil desain *interface* dari penelitian ini dapat dilihat pada Gambar 2.1



Gambar 2.1 Halaman Data Member

Penelitian kedua yang dilakukan oleh Nur Rubiati (2018), dengan judul Aplikasi Informasi pelayanan *Fitness* pada Golden *Fitness* Center Dumai dengan Bahasa Pemrograman Php. Aplikasi ini dirancang untuk memudahkan pengguna untuk mendapatkan informasi jadwal aerobik, instruktur aerobik, biaya, dan menghasilkan laporan terjadwal dan tidak terjadwal dalam waktu yang relatif singkat. Hasil desain *interface* dari penelitian ini dapat dilihat pada Gambar 2.2



Gambar 2.2 Form Transaksi Data Jadwal

Penelitian ketiga yang dilakukan oleh Marlince Nababan, Kevin, dan Andre Wijaya (2019), dengan judul Perancangan Aplikasi Member *Fitness* Berbasis Android Menggunakan App Inventory. Aplikasi ini dirancang untuk memudahkan manajemen member dan memudahkan anggota untuk mengetahui jadwal pelatihan dan status pembayaran mereka. Hasil desain *interface* dari penelitian ini dapat dilihat pada Gambar 2.3



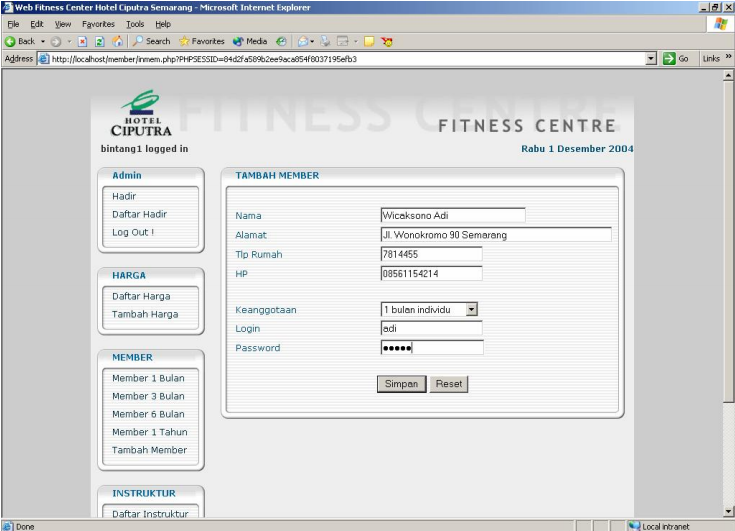
Gambar 2.3 Desain Menu Pilihan

Penelitian Keempat yang dilakukan oleh Kosidin dan Resha Nur Farizah (2016), dengan judul Pemodelan Aplikasi *Mobile Reminder* Berbasis Android. *Reminder* adalah fitur pesan yang membantu mengingat sesuatu, berbeda dengan alarm yang hanya berbunyi pada waktu tertentu. Dapat digunakan sebagai pengingat pada waktu tertentu saat melihat pesan yang dibuat sebelumnya. Hal ini diperlukan di zaman modern ini untuk melakukan aktivitas yang serba cepat dan padat untuk selalu mengingatkan setiap aktivitas. Aplikasi *Reminder* merupakan alternatif pengingat yang setia, mudah digunakan, dan portabel. Aplikasi ini diimplementasikan berbasis android, yang dapat diterapkan pada gadget .Hasil desain *interface* dari penelitian ini dapat dilihat pada Gambar 2.4



Gambar 2.4 Tampilan *Input* Agenda

Penelitian Kelima yang dilakukan oleh Graviardhi K.P (2017), dengan judul Sistem Informasi *Fitness* Center Hotel Ciputra Semarang. Sistem informasi ini dibuat untuk manajemen *fitness center* yang dapat melakukan otomatisasi dalam sistem registrasi anggota, instruktur jadwal dan fasilitas lainnya yang tergabung dalam sebuah *database* sehingga pengguna aplikasi bisa mendapatkan informasi sesuai dengan yang diinginkan secara cepat dan akurat. Hasil desain tambah memberdari terlihat pada Gambar 2.5.



Gambar 2.5 Menu tambah member

## Gym (Gymnastic)

Gym berasal dari Yunani kuno “Gymnastic” yang artinya sarana pendidikan yang sangat baik untuk melatih jasmani dan rohani. Di ruang *gymnasium*, anak-anak muda dilatih secara fisik dalam olahraga atletik untuk menjadi pribadi yang disiplin dan atletis. Bagi sebagian orang, kata "gym" selalu identik dengan ruangan yang penuh dengan orang berbadan kekar, dengan alat binaraga yang menekuk otot dan didampingi instruktur. Namun dalam arti luas, gym memiliki arti ruang olahraga atau gedung olahraga. Gym adalah tempat bagi mereka yang ingin menyegarkan diri melalui olahraga yang memungkinkan mereka untuk meregangkan, melatih otot dan membuat tubuh menjadi bugar.

## Fitness

Fitness adalah kegiatan yang bertujuan untuk membuat orang lebih sehat melalui angkat beban (*weight lifting*), aerobik (*aerobics*) dan pemenuhan nutrisi (*nutrition*). Pada dasarnya *fitness* adalah kombinasi antara aktivitas angkat beban dan kardiovaskuler (jantung).

## Sistem

Sistem berasal dari bahasa Latin (systēma) dan bahasa Yunani (sustēma) adalah suatu kesatuan yang terdiri dari komponen atau elemen yang dihubungkan bersama untuk memudahkan aliran informasi, materi atau energi untuk mencapai suatu tujuan. Istilah ini sering dipergunakan untuk menggambarkan suatu setenitas yang berinteraksi, dimana suatu model matematika sering kali bisa dibuat.

Sistem juga merupakan suatu kesatuan bagian-bagian yang saling berhubungan yang berada dalam suatu wilayah serta memiliki item-item penggerak, contoh umum seperti negara. Negara merupakan suatu kumpulan dari beberapa elemen kesatuan lain seperti provinsi yang saling berhubungan sehingga membentuk suatu negara dimana yang berperan sebagai penggeraknya yaitu rakyat yang berada di negara tersebut.

## Informasi

Informasi merupakan hasil pengolahan data, merupakan format yang penting bagi penerimanya dan menjadi dasar pengambilan keputusan yang dapat dirasakan akibatnya secara langsung saat itu juga atau secara tidak langsung pada saat mendatang (Sutanta, 2011). Informasi adalah data yang telah dirangkum atau dimanipulasi dalam bentuk lain untuk tujuan pengambilan keputusan (William, 2007).Sedangkan menurut (Fajri, 2014) informasi dapat diartikan suatu data yang telah diproses dan diubah menjadi konteks yang berarti sehingga memiliki makna dan nilai bagi penerimanya dan biasa digunakan untuk pengambilan keputusan. Secara Etimologi kata informasi berasal dari kata Perancis kuno informacion (tahun 1387) yang diambil dari bahasa latin *informationem* yang berarti “garis besar, konsep, ide”. Informasi merupakan kata benda dari informasi yang berarti aktivitas dalam “pengetahuan yang dikomunikasikan”.

Informasi adalah pengetahuan yang diperoleh melalui pembelajaran, pengalaman, atau pengajaran. Namun, istilah tersebut memiliki banyak makna kontekstual dan umumnya terkait erat dengan konsep-konsep seperti makna, pengetahuan, negentropy, komunikasi, kebenaran, representasi, dan stimulasi mental.

## Sistem Informasi

Sistem informasi adalah kombinasi dari teknologi informasi dan aktivitas orang-orang yang menggunakannya untuk mendukung operasi dan manajemen. Dalam arti yang sangat luas, istilah sistem informasi sering digunakan untuk menggambarkan interaksi antara manusia, proses algoritma, data, dan teknologi. Dalam pengertian ini, istilah tersebut tidak hanya mengacu pada penggunaan teknologi informasi dan komunikasi (TIK) oleh organisasi, tetapi juga cara orang berinteraksi dengan teknologi ini untuk mendukung proses bisnis.

## Aplikasi *Mobile*

Pengembangan aplikasi *mobile* sangat bergantung pada evolusi *smartphone*, baik dari segi teknologi maupun produksi. Tentu saja, *smartphone* adalah platform untuk menjalankan aplikasi seluler. Dalam waktu yang sangat singkat, pertumbuhan *smartphone* telah menunjukkan angka yang signifikan. Menurut Kementerian Informasi dan Komunikasi (Kominfo), Indonesia menempati urutan keenam dunia dalam hal penggunaan internet setelah China, Amerika Serikat, India, Brasil, dan Jepang. Seiring dengan itu, penggunaan *smartphone* semakin meningkat dari tahun ke tahun di Indonesia. Jumlah ini diproyeksikan akan terus meningkat menjadi 4.444, sejalan dengan antusiasme pasar. Seperti kedua sisi mata uang, pesatnya pertumbuhan smartphone membawa angin segar bagi pengembang aplikasi seluler. Secara garis besar, ada tiga pendekatan dalam pengembangan aplikasi *mobile*: aplikasi native, aplikasi web, dan aplikasi *hybrid*.

## Aplikasi *Web*

Aplikasi web adalah aplikasi situs web yang secara khusus dioptimalkan untuk pengguna di lingkungan smartphone. Aplikasi ini dibuat menggunakan teknologi web standar. Seperti HTML5, CSS3, JavaScript dan sebagainya. Pendekatan *write-once-anywhere* untuk aplikasi web mengarah ke aplikasi seluler lintas *platform* yang dapat berjalan di berbagai *platform* seluler.

## Web Service

Menurut *World Wide Web Consortium*, W3C, sebuah organisasi internasional yang mengkhususkan diri dalam pengembangan standarisasi layanan web, adalah sistem perangkat lunak yang dirancang untuk mengoperasikan mesin ke mesin melalui jaringan. Menurut Chen, arsitektur layanan web memodelkan interaksi antara tiga peran yaitu penyedia layanan, konsumen layanan, dan pendaftar layanan. Penggunaan layanan web juga digunakan oleh Pruter et al untuk terhubung secara dinamis dari perangkat yang dikenal dan tidak dikenal di jaringan komputer. Oleh karena itu, menggunakan layanan web adalah cara standar untuk bekerja di antara perangkat lunak pada platform yang berbeda.

## Representational State Transfer (Rest)

*Representational State Transfer* (*Rest*) menurut Sinha et al dan Zhou et al merupakan gaya arsitektur saat mendesain layanan web, dan desain lainnya memiliki sumber daya yang dapat diakses menggunakan alamat URL HTTP yang unik. Istirahat juga memungkinkan klien untuk dengan mudah membuat permintaan melalui protokol HTTP melalui URL. Setiap URL mengacu pada kumpulan program yang dijalankan dan mengembalikan pesan ke pengirim perintah.

## Application Programming Interface (API)

*Application Programming Interface* (API) adalah representasi terfokus dari fungsionalitas keseluruhan modul perangkat lunak yang dapat diakses oleh orang yang membutuhkannya dengan cara yang ditentukan layanan.

Representasi terfokus dari fungsi yang dideklarasikan dalam API dimaksudkan untuk menyediakan serangkaian layanan tertentu untuk tujuan tertentu. Hal ini biasa terjadi karena jika sebuah modul memiliki beberapa API, setiap API didedikasikan untuk penggunaan tertentu dari modul terkait (Rama dan Avinash, 2015).

## JavaScript Object Notation (JSON)

*JavaScript Object Notation* (JSON) adalah bentuk kecil dari umpan balik yang mudah dibaca dan ditulis oleh manusia, dan juga mudah diurai dan tulis oleh mesin. JSON adalah salah satu pesan kembali yang tersedia di REST API.

JSON terbagi dalam dua struktur, yaitu:

1. Gabungan *name/value*, atau yang dikenal dengan *object* atau *record* dalam bahasa pemrograman.
2. *List value*, atau yang dikenal dengan larik dalam bahasa pemrograman.

## Flowchart

Menurut Ladjamudin (2013) *flowchart* adalah bagan-bagan yang mempunyai arus yang menggambarkan langkah-langkah penyelesaian suatu masalah. *Flowchart* merupakan cara penyajian dari suatu algoritma. *Flowchart* menggambarkan setiap langkah-langkah yang diperlukan dalam setiap aktivitas atau kegiatan yang dilakukan dalam bentuk simbol. Adapun simbol-simbol dalam flowchart yang akan dijelaskan dalam Tabel 2.1 berikut ini.

Tabel 2.1 Simbol-Simbol *Flowchart*

|  |  |  |
| --- | --- | --- |
| Nama | Simbol | Keterangan |
| Simbol proses |  | Simbol proses digunakan untuk mewakili suatu proses |
| Simbol keputusan |  | Digunakan untuk memilih kondisi dalam program |
| Simbol *input* atau *output* |  | Digunakan untuk mewakili suatu data *input* atau *output* |
| Simbol proses terdefinisi |  | Digunakan untuk menunjukkan operasi yang ditampilkan di tempat lain |
| Simbol *display* |  | Digunakan untuk menampilkan *output* yang ditampilkan pada monitor |
| Simbol dokumen |  | Digunakan untuk menampilkan dokumen *input* dan *output* manual dan komputerisasi. |
| Simbol titik terminal |  | Digunakan untuk menunjukkan awal dan akhir dari suatu proses |
| Simbol penghubung |  | Digunakan untuk menunjukkan koneksi diagram alur pada halaman yang sama atau halaman lain. |
| Simbol diskette |  | Digunakan untuk menunjukkan sebuah *input* atau *output* menggunakan diskette |

## 

## 2.14 Unified Modeling Language (UML)

*Unified Modeling Language* (UML) adalah ‘bahasa’ sistem atau perangkat lunak dengan paradigma "berorientasi objek". Pemodelan (modeling) digunakan untuk menyederhanakan masalah yang kompleks sehingga lebih mudah dipelajari dan dipahami, Nugroho (2010: 6). Macam diagram UML antara lain *Use Case* Diagram*, Class* Diagram*, Activity* Diagram*, Sequense* Diagram*.*

*Use Case* Diagram (Tabel 2.2) memiliki simbol-simbol dan pengertian. Setiap simbol memiliki fungsi yang berbeda karena itu setiap programmer wajib mengetahui demi mempermudah perancangan perangkat lunak yang akan dibuat.

Tabel 2.2 Notasi *Use Case* Diagram

|  |  |  |
| --- | --- | --- |
| SIMBOL | NAMA | KETERANGAN |
|  | *Actor* | *Actor* adalah pengguna sistem*. Actor* tidak terbatas pada manusia. Sebuah aplikasi juga dapat dianggap sebagai *actor* jika sistem berkomunikasi dengan aplikasi lain dan membutuhkan *input* atau memberikan *output.* |
|  | *Use Case* | *Use case* digambarkan sebagai lingkaran *elips* dengan nama *use case* di dalam *elips.* |
|  | *Association* | *Association* digunakan untuk menghubungkan *actor* dengan *use case.* *Association* diwakili oleh garis yang menghubungkan *actor* dan *use case* |

Salah satu dari diagram-diagram yang ada pada UML, *Sequence* Diagram (Tabel 2.3) adalah diagram yang mewakili kolaborasi dinamis antara satu set objek. Tujuannya adalah untuk menunjukkan urutan pesan yang dikirim antara objek juga interaksi antara objek..

Tabel 2.3 Notasi *Sequence Diagram*

|  |  |  |
| --- | --- | --- |
| SIMBOL | NAMA | KETERANGAN |
|  | *Object* | *Object* adalah turunan dari *class* dan ditulis secara *horizontal*. Direpresentasikan sebagai *class* (kotak) yang berisi nama objek yang dimulai dengan titik koma. |
|  | *Actor* | *Actor* juga dapat berkomunikasi dengan *object,* sehingga Anda dapat mengurutkannya sebagai kolom. Ikon *actor* sama dengan ikon diagram kasus penggunaan *acto*r. |
|  | *Lifeline* | *Lifeline* menunjukkan keberadaan *object* berdasarkan waktu. Notasi *lifeline* adalah garis putus-putus vertikal yang ditarik dari *object.* |
|  | *Activation* | *Activation* direpresentasikan sebagai kotak persegi panjang di garis hidup. *Activation* menentukan objek untuk melakukan tindakan. |
|  | *Message* | *Message* diwakili oleh panah horizontal di antara aktivasi. Pesan menunjukkan komunikasi antara objek. |

*Activity* diagram (Tabel 2.4), sesuai dengan namanya diagram ini menggambarkan tentang aktivitas yang terjadi pada sistem. Dari pertama sampai akhir, diagram ini menunjukkan langkah–langkah dalam proses kerja sistem yang kita buat.

Tabel 2.4 Notasi *Activity Diagram*

|  |  |
| --- | --- |
| SIMBOL | KETERANGAN |
|  | Titik Awal |
|  | Titik Akhir |
|  | *Activity* |
|  | Pilihan Untuk mengambil Keputusan |

*Class* Diagram (Tabel 2.5) adalah model yang menunjukkan struktur *class* dan relasi dengan class yang lain.

Tabel 2.5 Notasi *Class Diagram*

|  |  |  |
| --- | --- | --- |
| SIMBOL | NAMA | KETERANGAN |
|  | *Assosiation* | *Assosiation* adalah hubungan yang paling umum antara dua kelas dan diwakili oleh garis yang menghubungkan dua kelas.  Baris ini dapat mewakili tipe *relationship* dan juga dapat mewakili beberapa hukum dari suatu hubungan (Contoh: *One-to-one, one-to-many, many-to-many*). |
|  | *Composition* | Jika suatu *class* tidak dapat berdiri sendiri dan perlu menjadi bagian dari *class* lain, *class* tersebut berada dalam hubungan konstitutif dengan *class* yang menjadi sandarannya. Hubungan komposisi diwakili oleh garis padat dengan ujung jajaran genjang. |
|  | *Dependency* | *Class* dapat menggunakan *class* yang berbeda. Ini disebut *dependency*. Penggunaan *dependency* biasanya digunakan untuk menunjukkan operasi satu *class* yang menggunakan *class* lain. *Dependency* diwakili oleh panah putus-putus. |
|  | *Aggregation* | *Aggregation* mewakili seluruh hubungan dan biasanya disebut sebagai hubungan “mempunyai sebuah” atau “bagian dari”. Sebuah *aggregation* digambarkan  sebagai sebuah garis dengan sebuah jajaran genjang yang tidak berisi/tidak solid. |
|  | *Generalization* | Hubungan umum setara dengan relasi *inheritance* dalam konsep berorientasi objek. Generalisasi diwakili oleh panah dengan panah padat yang menunjuk ke kelas "*parent*" . |

## MySQL

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL atau dikenal dengan *Database Management System* (DBMS) *Database* ini *multithread* dan *multi user*. MySQL pertama kali dikembangkan oleh seorang programmer database bernama Michael Widenieus.. MySQL *database* server adalah RDBMS (*Relational Database Management System*) yang dapat menangani data dalam jumlah besar. Meskipun demikian, itu tidak membutuhkan banyak sumber daya. MySQL adalah *database* yang paling populer dari semua *database* lainnya.

## Android

Menurut Irawan (2012), Android merupakan sebuah sistem operasi yang berbasis Linux untuk perangkat *portable* seperti *smartphone* dan komputer tablet. Sedangkan menurut Juhara (2016), Android adalah sistem operasi berbasis Linux yang dimodifikasi untuk perangkat bergerak (*mobile devices*) yang terdiri dari sistem operasi, *middleware*, dan aplikasi-aplikasi utama.

Android adalah sistem operasi yang dikembangkan oleh Google dan merupakan *platform* pemrograman untuk *smartphone* dan perangkat seluler lainnya (seperti tablet). Android bekerja pada perangkat yang berbeda dari produsen yang berbeda. Android menyertakan *kit development* perangkat lunak untuk menulis kode asli, merakit modul perangkat lunak, dan membuat aplikasi untuk pengguna Android. Android juga menyediakan pasar untuk mendistribusikan aplikasi. Secara keseluruhan, Android mewakili ekosistem aplikasi seluler (Google Developers, 2016).

## Android Studio

Android Studio merupakan sebuah *Integrated Development Environment* (IDE) yang didedikasikan untuk membuat aplikasi yang berjalan di platform Android. Android Studio didasarkan pada IntelliJ IDEA, sebuah IDE untuk bahasa pemrograman Java. Bahasa pemrograman utama yang digunakan adalah Java, tetapi XML digunakan untuk membuat tampilan atau tata letak. Android Studio juga terintegrasi dengan *Android Software Development Kit* (SDK) untuk diterapkan pada perangkat Android. Android Studio juga merupakan pengembangan dari Eclipse, dirancang agar lebih kompleks dan profesional, dan sudah tersedia di Android Studio IDE dan Android SDK *tool*s.

## Java

. Java adalah bahasa pemrograman tingkat tinggi yang berorientasi objek dan program java tersusun dari bagian yang disebut kelas. Kelas terdiri atas metode-metode yang melakukan pekerjaan dan mengembalikan informasi setelah melakukan tugasnya. Kumpulan kelas di pustaka kelas Java disebut dengan Java *Application Programming Interface* (API). Kelas-kelas ini diorganisasikan menjadi sekelompok yang disebut paket (*package*). Java API telah menyediakan fungsionalitas yang memadai untuk menciptakan applet dan aplikasi canggih.

*Interface* adalah seperangkat metode struktural dalam bahasa pemrograman Java yang berfungsi sebagaipenyimpanan *constant* atau metode dan atribut kosong yang tidak diimplementasikan. Hal ini dapat diimplementasikan sebagai *interface* dengan kelas lain dan dapat diwarisi dari *interface* lain. Di dalam sebuah class juga dapat mengimplementasikan atau mewarisi beberapa interface dalam sebuah class. Hal ini biasa disebut sebagai *Multiple inheritance*. *Interface* tidak dapat dipakai seperti class abstract Method di *interface* harus di set *modifier* ke *public*. *Class* yang mengimplementasikan *interface,* perlu mengimplementasikan metode yang tersedia di *interface.*

## Extensible Markup Language (XML)

*Extensible Markup Language* (XML) adalah Bahasa *markup* sekumpulan aturan-aturan yang mendefinisikan suatu sintaks yang digunakan untuk mendeskripsikan teks atau data dalam dokumen menggunakan tag. XML dikembangkan untuk menyimpan data dengan cara yang jelas dan mudah dikelola. Kata kunci utama dalam XML adalah data (jamak dari datum) yang dapat memberikan informasi saat diproses.

*Extensible Markup Language* (XML) adalah bahasa markup untuk memproses metadata (informasi tentang data) yang menjelaskan struktur dan tujuan data yang terdapat dalam dokumen XML, tetapi bukan format tampilan data tersebut. XML adalah standar sederhana yang digunakan untuk menggambarkan data tekstual dengan cara *self-describing* (deskripsi diri)..

## Android Software Development Kit (SDK)

*Android Software Development Kit* (SDK) adalah perangkat lunak yang dirancang untuk membangun aplikasi Android. Android SDK (*Software Development Kit*) kini tersedia sebagai alat dan API untuk memulai mengembangkan aplikasi pada *platform* Android menggunakan bahasa pemrograman Java. Selama proses instalasi, akan diberikan opsi untuk paket Android yang digunakan untuk pengembangan.

## Application Programming Interface Level (API Level)

*Application Programming Interface Level* (API level) adalah nilai integer yang secara unik mengidentifikasi revisi API *framework* yang disediakan oleh versi platform Android. *Platform* Android menyediakan kerangka kerja API yang dapat digunakan aplikasi untuk berinteraksi dengan sistem android yang mendasarinya. API *framework* terdiri dari:

* Paket dan set inti kelas
* Satu set elemen dan atribut XML untuk mendeklarasikan file manifes
* Satu set elemen dan atribut XML untuk mendeklarasikan dan mengakses sumber daya
* Satu set intent
* Satu set izin yang dapat diminta aplikasi dan penegakan izin yang dimasukkan pada sistem.

Versi *platform* Android yang lebih baru dapat menyertakan pembaruan ke API framework aplikasi Android yang disediakan.

*Update* API framework dirancang untuk menjaga kompatibilitas antara API baru dan versi API yang lebih lama. Ini berarti bahwa sebagian besar perubahan API dilakukan secara bertahap dan fitur baru atau pengganti. Beberapa API telah diperbarui untuk menggantikan vesri lama dan tidak digunakan lagi, tetapi belum dihapus dan masih dapat digunakan di aplikasi yang ada. Dalam kasus yang sangat jarang, beberapa bagian API dapat diubah atau dihapus, tetapi perubahan tersebut biasanya hanya diperlukan untuk memastikan ketahanan API dan keamanan aplikasi atau sistem. Semua bagian API lainnya dari revisi sebelumnya akan dibawa dan digunakan tanpa modifikasi..

API framework yang disediakan oleh platform Android didefinisikan menggunakan pengenal integer yang disebut "API *level*". Setiap versi platform Android mendukung tepat satu API *level*, tetapi dukungan untuk semua API *level* sebelumnya (hingga API *level* 1) bersifat implisit.

Pada penelitian ini penulis memakai API *level* 21 atau android 5.0 ke atas. Tabel API bisa dilihat pada Tabel 2.6.

Tabel 2.6 API *Level*

|  |  |  |  |
| --- | --- | --- | --- |
| Versi Platform | API Level | VERSION\_CODE | Catatan |
| Android 10.0 | 29 | Q | Sorotan Platform |
| Android 9 | 28 | P | Sorotan Platform |
| Android 8.1 | 27 | O\_MR1 | Sorotan Platform |
| Android 8.0 | 26 | O | Sorotan Platform |
| Android 7.1.1 | 25 | N\_MR1 | Sorotan Platform |
| Android 7.1 | 25 | N\_MR2 | Sorotan Platform |
| Android 7.0 | 24 | N | Sorotan Platform |
| Android 6.0 | 23 | M | Sorotan Platform |
| Android 5.1 | 22 | LOLLIPOP\_MR1 | Sorotan Platform |
| Android 5.0 | 21 | LOLLIPOP | Sorotan Platform |
| Android 4.4W | 20 | KITKAT\_WATCH | Sorotan Platform |
| Android 4.4 | 19 | KITKAT | Sorotan Platform |
| Android 4.3 | 18 | JELLY\_BEAN\_MR2 | Sorotan Platform |
| Android 4.2, 4.2.2 | 17 | JELLY\_BEAN\_MR1 | Sorotan Platform |
| Android 4.1, 4.1.1 | 16 | JELLY\_BEAN | Sorotan Platform |
| Android 4.0.3, 4.0.4 | 15 | ICE\_CREAM\_SANDWICH\_MR1 | Sorotan Platform |