



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB II LANDASAN TEORI

2.1 Petugas Masyarakat

2.1.1 RT

RT (Rukun Tetangga) adalah organisasi yang ada di lingkungan warga dengan proses pembentukannya dilakukan berdasarkan kedekatan alam tempat tinggal yang bisa disebut saling bertetangga. Sehingga dalam hal ini setiap anggota-anggota RT terdiri dari para kepala keluarga yang saling bertetangga satu sama lainnya.

Namun bisa di artikan juga sebagai gambaran dari pada sistem pemerintahan presidensial terkecil yang ada dalam kehidupan warga Indonesia. Setiap wilayah Indonesia dalam pengertian daerah otonom memiliki RT yang tujuannya sebenarnya ialah menjadi tangan panjang dari tugas-tugas yang diberikan kepada desa, RW, yang kemudian disampaikan dalam warga. (Maya Sari, 2017)

2.1.2 RW

Rukun Warga dalam sistem pemerintahan Indonesia yang bersifat sebagai sistem pemerintahan presidensial merupakan bagian pembagian di bawah keluarahan/desa/pekon. Dalam setiap penentuan

RW dilakukan secara musyawarah mufakat antar warga, bisa juga dilakukan dalam pengertian demokrasi, utamanya untuk pemilihan demokratis dilakukan pada wilayah Indonesia yang berada di perkotaan. (Maya Sari, 2017)

2.2 Android

Android adalah sebuah sistem operasi untuk perangkat mobile berbasis linux yang mencakup sistem operasi, *middleware*, dan aplikasi. Di dunia ini terdapat dua jenis distributor sistem operasi Android. Pertama yang mendapat dukungan penuh dari Google atau *Google Mobile Services* (GMS) dan kedua adalah yang benar-benar bebas distribusinya tanpa dukungan langsung Google atau dikenal sebagai *Open Handset Distribution* (OHD). (Efmi Maiyana, 2018)



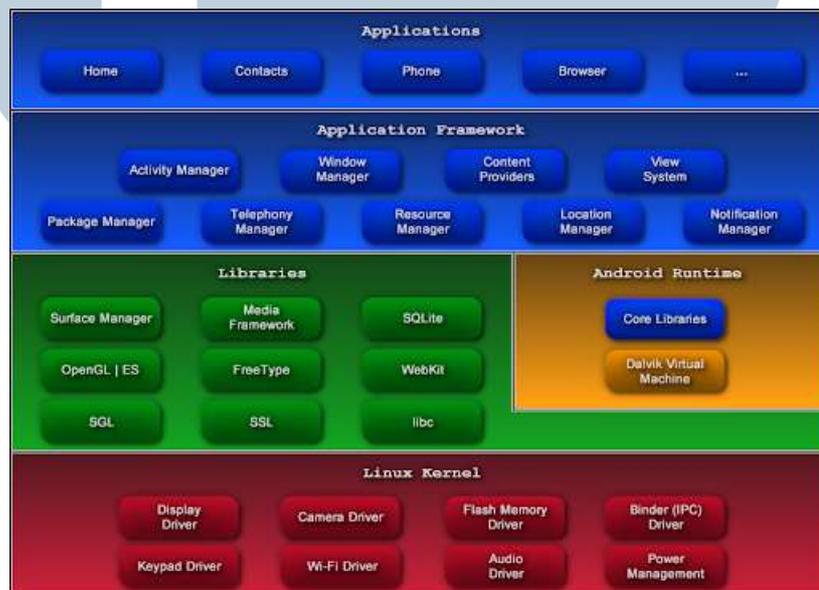
Gambar 2.1. Logo Android

Sumber: (irinablok, 2018)

Pada dasarnya saat ini kebanyakan vendor-vendor smartphone sudah memproduksi smartphone berbasis *Android*. Hal ini karena android itu adalah sistem operasi yang open source sehingga bebas di distribusikan dan dipakai oleh vendor manapun. Pesatnya pertumbuhan android selain faktor yang disebutkan sebelumnya adalah karena android itu sendiri adalah platform yang sangat lengkap baik sistem operasinya, aplikasi dan *Tool* Pengembangan, Market aplikasi android serta dukungan yang sangat tinggi dari komunitas open source dunia. (Chandra Kirana, 2018)

2.3 Arsitektur *Android*

Android memungkinkan penggunaannya untuk memasang aplikasi pihak ketiga, baik yang diperoleh dari toko aplikasi seperti Google Play, Amazon App store, ataupun dengan mengunduh dan memasang berkas APK dari situs pihak ketiga. Di Google Play, pengguna bisa menjelajah, mengunduh, dan memperbarui aplikasi yang diterbitkan oleh Google dan pengembang pihak ketiga, sesuai dengan persyaratan kompatibilitas Google. Google Play akan menyaring daftar aplikasi yang tersedia berdasarkan kompatibilitasnya dengan perangkat pengguna.



Gambar 2.2. Skema Arsitektur *Android*

Sumber: (Hafizh Herdi, 2012)

Android dibangun dengan menggunakan asal object oriented, dimana elemen-elemen penyusun sistem operasinya berupa objek yang dapat kita gunakan kembali/*reusable*. Agar bisa membuat aplikasi dengan baik, tentunya kita harus mengetahui arsitektur OS *Android* beserta elemennya. (Novri Hadinata, 2017)

2.3.1 Layer Applications dan Widget

Inilah layer pertama pada OS Android, biasa dinamakan layer *Applications* dan *Widget*. Layer ini merupakan layer yang berhubungan dengan aplikasi-aplikasi inti yang berjalan pada Android OS. Seperti klien email, program SMS, kalender, browser, peta, kontak, dan lain-lain. Semua aplikasi ini dibuat dengan menggunakan bahasa *Java*. Apabila kalian membuat aplikasi, maka aplikasi itu ada di layer ini. (Novri Hadinata, 2017)

2.3.2 Layer Applications Framework

Applications Framework merupakan layer dimana para pembuat aplikasi menggunakan komponen-komponen yang ada di sini untuk membuat aplikasi mereka. Beberapa contoh komponen yang termasuk di dalam *Applications Framework* adalah sebagai berikut:

1. *Views*
2. *Content Provider*
3. *Resource Manager*
4. *Notification Manager*
5. *Activity Manager*

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A

2.3.3 Layer Libraries

Libraries merupakan layer tempat fitur-fitur android berada. Pada umumnya *libraries* diakses untuk menjalankan aplikasi. Beberapa *libraries* yang terdapat pada *android* diantaranya adalah *libraries* Media untuk memutar media video atau audio, *libraries* untuk menjalankan tampilan, *libraries* *Graphic*, *libraries* SQLite untuk dukungan *database*, dan masih banyak *libraries* lainnya. (Novri Hadinata, 2017)

2.3.4 Android RunTime

Android RunTime merupakan layer yang membuat aplikasi android bisa dijalankan. Android RunTime dibagi menjadi dua bagian yaitu:

- *Core Libraries* : berfungsi untuk menerjemahkan bahasa Java/C
- *Dalvik Virtual Machine* : sebuah mesin virtual berbasis register yang dioptimalkan untuk menjalankan fungsi-fungsi pada *Android* secara efisien. (Novri Hadinata, 2017)

2.3.5 Linux Kernel

Linux Kernel merupakan layer tempat keberadaan inti dari operating system android. Layer ini berisi file-file system yang mengatur *system processing*, *memory*, *resource*, *drivers*, dan sistem *android* lainnya. Inilah yang membuat file sistem pada Android mirip dengan file sistem pada sistem operasi berbasis Linux. *Kernel* yang digunakan adalah kernel Linux versi 2.6, dan versi 3.x pada *Android* versi 4.0 ke atas. Kernel ini berbasis *monolithic*. (Novri Hadinata, 2017)

2.4 MySQL

MySQL (*My Structure Query Language*) adalah sebuah perangkat lunak sistem manajemen basis data SQL (*Database Management System*) atau DBMS yang multithread, multi-user. MySQL memungkinkan secara efisien menyimpan, mencari, mengurutkan dan mendapatkan data. MySQL tersedia sebagai perangkat lunak gratis dibawah lisensi GNU *General Public License* (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL. (IDreg, 2013) Tidak sama dengan proyek-proyek seperti Apache, dimana perangkat lunak dikembangkan oleh komunitas umum, dan hak cipta untuk kode sumber dimiliki oleh penulisnya masing-masing, MySQL dimiliki dan disponsori oleh sebuah perusahaan komersial Swedia MySQL AB, dimana memegang hak cipta hampir atas semua kode sumbernya. Kedua orang Swedia dan satu orang Finlandia yang mendirikan MySQL AB adalah: David Axmark, Allan Larsson, dan Michael “Monty” Widenius. (Astria Firman, 2016)

2.5 PHP

PHP adalah *script* untuk pemrograman *script web server-side*, script yang membuat dokumen HTML secara *on the fly*, dokumen HTML yang dihasilkan dari suatu aplikasi bukan dokumen HTML yang dibuat dengan menggunakan editor teks atau editor HTML. HTML adalah bahasa pemrograman dasar yang harus dikuasai jika ingin membuat suatu website. Untuk mengetahui lebih lanjut tentang HTML, simak artikel Belajar HTML Lengkap untuk Pemula.

Dengan menggunakan PHP maka maintenance suatu situs web menjadi lebih mudah. Proses update data dapat dilakukan dengan menggunakan aplikasi yang dibuat dengan menggunakan script PHP. PHP/FI merupakan nama awal dari PHP. PHP merupakan singkatan dari *Personal Home Page* dan FI adalah singkatan dari *Form Interface*. Dibuat pertama kali oleh Rasmus Lerdoff. PHP, awalnya merupakan program CGI yang dikhususkan untuk menerima input melalui form yang ditampilkan dalam *browser web*. (Tantri Wahyuni, 2018)

2.6 JSON

JavaScript Object Notation (JSON) adalah sebuah format untuk berbagi data. Seperti dapat kita lihat dari namanya, JSON diturunkan dari bahasa pemrograman *JavaScript*, akan tetapi format ini tersedia bagi banyak bahasa lain termasuk *Python*, *Ruby*, *PHP*, dan *Java*. JSON biasanya dilafalkan seperti nama "*Jason*." (Meilvin Wijaya, 2018)

JSON menggunakan ekstensi *.json* saat ia berdiri sendiri. Saat didefinisikan di dalam format file lain (seperti di dalam *.html*), ia dapat tampil didalam tanda petik sebagai *JSON string*, atau ia dapat dimasukkan kedalam sebuah variabel. Format ini sangat mudah untuk ditransfer antar *server web* dengan klien atau browser. Karena sangat mudah dibaca dan ringan, JSON memberikan alternatif lebih baik dari *XML* dan membutuhkan *formatting* yang tidak banyak. Panduan ini akan membantu pembaca untuk memahami apa itu JSON, bagaimana menggunakan data di file JSON, serta struktur dan sintaks dari format ini. (Ulfa Mariathul Qibtiyah, 2017)

2.7 SDLC (Systems Development Life Cycle)

SDLC mengacu pada model dan proses yang digunakan untuk mengembangkan sistem perangkat lunak dan menguraikan prosesnya dengan menerima perpindahan dari permasalahan ke solusi. Pengembangan rekayasa sistem informasi (*system development*) dan perangkat lunak (*software engineering*) dapat diartikan untuk menyusun sistem atau perangkat lunak yang baru tercipta atau menggunakannya untuk menyempurnakan yang sudah ada. SDLC sendiri terbagi menjadi beberapa model dan salah satunya adalah RAD (*Rapid Application Development*) yang nantinya akan membantu dalam perancangan aplikasi mobile agar dapat di mengerti oleh peneliti lain untuk dapat di kembangkan lebih lanjut. (Rahmat Inggi, 2018)

2.7.1 *Rapid application development (RAD)*

Rapid Application Development (RAD) adalah istilah awalnya digunakan untuk menggambarkan proses pengembangan perangkat lunak pertama kali dikembangkan dan berhasil digunakan selama pertengahan 1970-an oleh Sistem Pusat Pengembangan New York Telephone Co di bawah arahan Dan Gielan. Setelah serangkaian implementasi sangat berhasil dari proses ini, Gielan kuliah

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A

Secara ekstensif di berbagai forum pada metodologi , praktek, dan manfaat dari proses ini. (Okky Irnawati, 2018)



Gambar 2.3. Rapid Application Development

Sumber: (piyaneo, 2014)

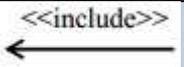
RAD melibatkan pengembangan dan pembangunan prototipe iteratif . Pada tahun 1990 , dalam buku RAD, *Rapid Application Development*, James Martin di dokumentasikan penafsirannya tentang metodologi Baru-baru ini, istilah dan singkatan yang telah datang untuk digunakan dalam lebih luas, pengertian umum yang mencakup berbagai metode yang bertujuan untuk mempercepat pengembangan aplikasi ada beberapa tahapan yang harus di lalui untuk bisa membuat sebuah aplikasi dengan tahapan *Bussiness Modeling, Data Modeling, Process Modelling, Application generation, Testing and turnover* seperti yang nanti akan di terapkan dalam pembuatan aplikasi mobile ataupun jenis lainnya. (Aeni Zuhana Saidin, 2015)

UNIVERSITAS
MULTIMEDIA
NUSANTARA

2.7.2 Use Case Diagram

Use case adalah kegiatan atau urutan interaksi yang saling berkaitan antara sistem dan aktor. *Use case* bekerja dengan cara mendeskripsikan tipe interaksi antara user sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem bisa digunakan *user*. *Use case* juga digunakan untuk membentuk aktifitas sistem yang akan dibuat. Sebuah *use case* menggambarkan sebuah interaksi antara pengguna (aktor) dengan sistem yang ada. (Tri A. Kurniawan, 2018)

Tabel 2.1 Komponen pada *Use Case Diagram*

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

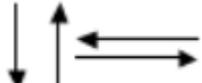
U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A

2.7.3 Activity Diagram

Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal dari keputusan yang mungkin terjadi, dan bagaimana mereka berakhir dalam sebuah aktifitas.

Activity diagram juga dapat menggambarkan proses *paralel* yang mungkin terjadi pada beberapa eksekusi. *Activity diagram* merupakan *state diagram* khusus, di mana sebagian besar *state* adalah *action* dan sebagian besar transisi di-trigger oleh selesainya *state* sebelumnya (*internal processing*). Oleh karena itu *activity diagram* tidak menggambarkan *behaviour internal* sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum. (Ade Hendini, 2016)

Tabel 2.2 Komponen pada Activity Diagram

Gambar	Nama	Keterangan
	<i>Action</i>	State dari sistem yang mencerminkan eksekusi dari suatu aksi.
	<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.
	<i>Activity Final Node</i>	Bagaimana objek dibentuk atau diakhiri.
	<i>Decision</i>	Digunakan untuk menggambarkan suatu keputusan/tindakan yang harus diambil pada kondisi tertentu.
	<i>Line Connector</i>	Digunakan untuk menghubungkan satu simbol dengan simbol lainnya.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

2.7.4 Class Diagram

Class diagram adalah model statis yang menggambarkan struktur dan deskripsi *class* serta hubungannya antara *class*. *Class diagram* mirip ERD pada perancangan database, bedanya pada *class diagram* tidak terdapat operasi/*methode* tapi hanya atribut. *Class* terdiri dari nama kelas, atribut dan operasi/*methode*.

- Elemen Class Diagram
 - *Class* (Nama)
 - *Attributes*
 - *Operations*
 - *Relationships*

Dalam *class diagram* terdapat penghubung antara satu class dengan class lainnya, berikut adalah tabel dari simbol hubungan antara *class* yang digunakan pada *class diagram* (Ade Hendini, 2016)

Tabel 2.3 simbol hubungan *class diagram*

Nama	Simbol	Keterangan
Association		Hubungan antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan multiplicity
Directed Association		Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan multiplicity
Generalisasi		Relasi antar kelas dengan makna generalisasi-spesialisasi
Dependency		Relasi antar kelas dengan makna kebergantungan antar kelas
Aggregation		Relasi antar kelas dengan makna semu (whole-part)

2.8 Penelitian Terdahulu

Berikut adalah tabel 2.3 yaitu beberapa penelitian terdahulu yang akan dijadikan landasan dan tolak ukur pada penelitian ini, yaitu:

Tabel 2.3 Penelitian Terdahulu

No.	Judul	Masalah	Solusi	Kesimpulan
1	Implementasi metode RAD pada Rancang Aplikasi BAN-SOS Ter Distribusi Berbasis <i>Mobile</i> Nama Penulis: Dini Silvi Purnia Tahun Penulisan: 2018 Penerbit: IJCIT (<i>Indonesian Journal on Computer and Information Technology</i>) (Dini Silvi Purnia, 2018)	Banyaknya Organisasi sosial yang membantu pemerintah dalam menyalurkan dana bansos tapi banyak pula yang beberapa organisasi yang menggunakan dana tersebut untuk diri sendiri	Merancang aplikasi berbasis mobile untuk mengintegrasikan organisasi serta LSM yang berbadan hukum agar pemerintah dan masyarakat dapat mengontrol dan mengetahui aktifitas dari organisasi sosial tersebut baik dalam menerima Mendistribusian Ban-Sos dan penyaluran Dana	aplikasi ini digunakan untuk mengabungkan semua organisasi sosial sehingga memudahkan masyarakat untuk mengetahui organisasi sosial mana yang berbadan hukum, sehingga masyarakat lebih mudah menyalurkan bantuannya.
2.	Metode <i>Rapid Application Development</i> (RAD) pada Perancangan <i>Website</i> Inventory PT. Sarana Abadi Makmur Bersama (S.A.M.B) Jakarta	Karyawan sering mengeluhkan kinerja dari aplikasi desktop yang dipakai. Dalam aplikasi tersebut tidak bisa menyimpan data dan informasi	Pengembangan sistem inventory berbasis website dengan pengolahan data yang terpusat guna untuk meminimalisi terjadinya kesalahan pada memasukan data serta terintegrasi dengan baik dari	Program inventory ini merupakan aplikasi sistem komputerisasi yang dibuat berbasis web dan memuat database pengolahan data secara terpusat sehingga dapat

	<p>Nama Penulis: Oky Irnawati</p> <p>Tahun Penulisan: 2018</p> <p>Penerbit: Manajemen Informatika AMIK BSI Bekasi</p> <p>(Oky Irnawati, 2018)</p>	<p>secara terpusat pada suatu basis data, sehingga sering terjadi ketidaksesuaian informasi</p>	<p>satu divisi ke divisi yang lain</p>	<p>mengolah database tersebut menjadi informasi yang dibutuhkan oleh para pengguna.</p>
3	<p><i>Eyeglasses Frame Selection Based On Face Shape with Convolutional Neural Network Method</i></p> <p>Nama Penulis: Stephen Young</p> <p>Tahun Penulisan: 2018</p> <p>Penerbit: Mutlimedia Nusantara University</p> <p>(Stephen Young, 2018)</p>	<p>Bagaimana cara untuk bisa mengklasifikasi gambar untuk bisa membedakan antar wajah seseorang apakah memiliki bentuk oval atau tidak agar dapat bisa menentukan kacamata yang cocok untuk bentuk wajahnya</p>	<p>Penggunaan metode <i>Convolutional Neural Network</i> sebagai penentu untuk menentukan bentuk wajah apakah memiliki bentuk oval atau tidak serta penentuan kacamata yang cocok</p>	<p>Untuk mengetahui cara membuat aplikasi <i>android</i> dengan menggunakan <i>Android Studio</i> serta mengetahui tahap-tahap yang diperlukan untuk pembuatan aplikasi.</p>

Tabel 2.3 Menjelaskan bahwa penelitian terdahulu menjadi bahan acuan untuk dapat melakukan pembuatan aplikasi RT/RW sehingga aplikasi yang di buat tidak salah dalam perancangan maupun pengimplementasiannya serta menjadi landasar untuk memilih metode perancangan yang tepat bagi aplikasi RT/RW yang akan dibuat.