BAB II

TINJAUAN PUSTAKA

II.1 Sekilas Tempat Penelitian

Satuan Polisi Pamong Praja kota Bandung atau biasa disingkat Satpol PP kota Bandung beralamat di jalan R.E Martanegara No.4 Bandung. Satpol PP kota Bandung sendiri berdiri pada tahun 1950. Satpol PP merupakan perangkat pemerintah daerah dengan tugas pokok menegakkan perda, menyelenggarakan ketertiban umum dan ketenteraman masyarakat. Menegakan perda disini termasuk perda No.4 tahun 2011 tentang pedagang kaki lima.

II.1.1 Visi dan Misi Satuan Polisi Pamong Praja kota Bandung

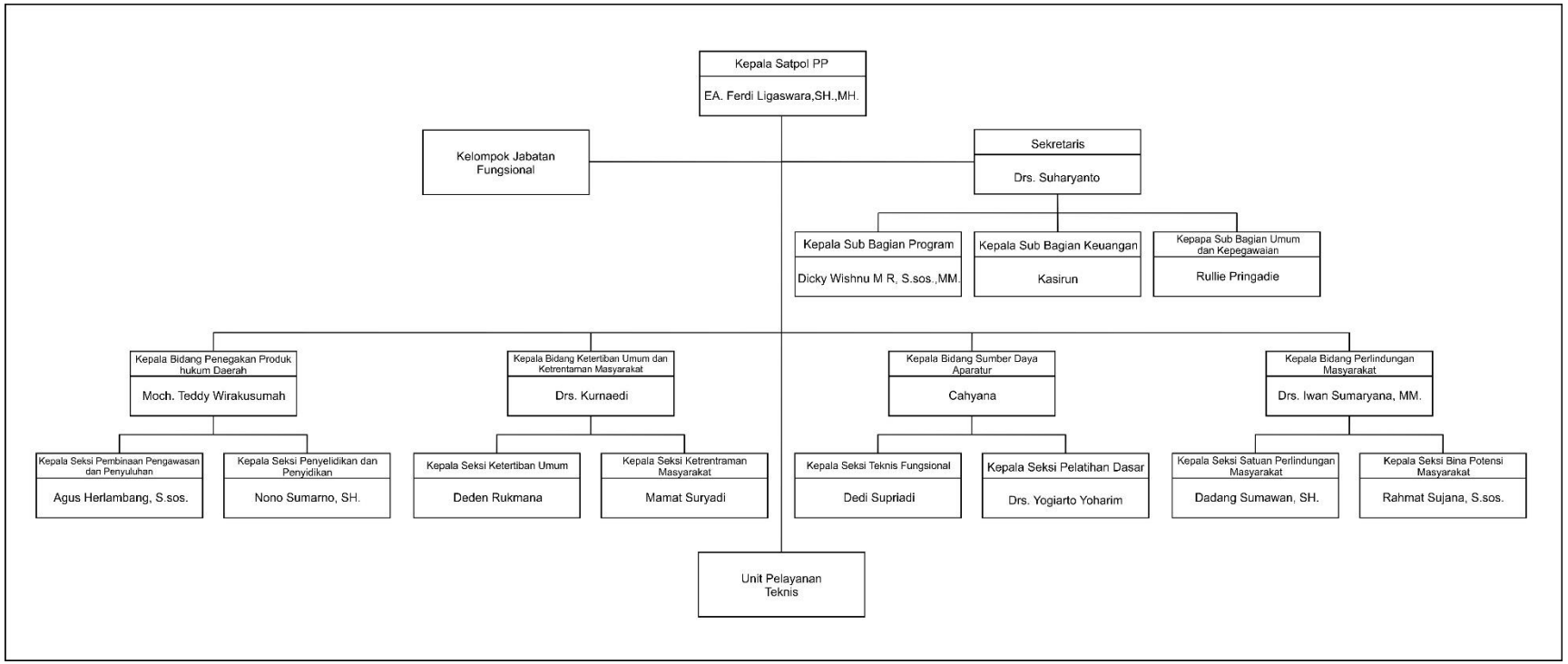
Visi Satuan Polisi Pamong Praja kota Bandung adalah Terwujudnya Masyarakat kota Bandung yang Tenteram, Tertib dan Taat Hukum. Dari pernyataan visi tersebut terdapat kata-kata kunci yang mengandung makna:

1. Tentram adalah suatu tatanan yang sesuai dengan kaidah hukum, norma agama, norma sosial dan peraturan perundang-undangan sehingga terselenggara sendi-sendi kehidupan yang menjamin rasa aman dan tenang di daerah.
2. Tertib adalah suatu keadaan kehidupan yang serba teratur dan tertata dengan baik sesuai ketentuan perundang-undangan yang berlaku guna mewujudkan kehidupan masyarakat yang dinamis, aman, tenteram lahir dan bathin.
3. Taat Hukum adalah suatu bentuk kesadaran individu ataupun kolektif yang memahami bahwa hidup bermasyarakat, berbangsa dan bernegara terdapat ketentuan-ketentuan adanya hak, kewajiban serta larangan yang harus dipatuhi bersama agar kehidupan menjadi teratur.

Sedangkan misi dari Satuan Polisi Pamong Praja kota Bandung adalah meningkatkan penyelenggaraan dan pemeliharaan ketenteraman dan ketertiban umum serta menumbuhkan kepatuhan hukum masyarakat.

II.1.2 Struktur Organisasi

Struktur organisasi di Satuan Polisi Pamong Praja kota Bandung dapat dilihat pada Gambar II.1 berikut:



Gambar II. 1 Struktur Organisasi Satuan Polisi Pamong Praja kota Bandung

Pada struktur organisasi tersebut bagian yang akan terlibat didalam sistem adalah bagian Bidang Penegakan Produk Hukum Daerah dan Bidang Ketertiban Umum dan Ketrentaman Masyarakat, dimana kedua bagian tersebut membawahi masing-masing 2 subbagian. Pada Bidang Penegakan Produk Hukum Daerah terdapat subbagian Pembinaan Pengawasan dan Penyuluhan serta Penyelidikan dan Penyidikan. Pada bidang Ketertiban Umum dan Ketrentaman Masyarakat terdapat subbagian Ketertiban Umum dan Ketrentaman Masyarakat.

II.1.3 Deskripsi Kerja

Berikut ini adalah deskripsi kerja pada struktur organisasi yang nantinya akan terlibat didalam sistem. Terdapat dua bagian yang terlibat yaitu bagian Penegakan Produk Hukum Daerah Serta Ketertiban Umum dan Ketentraman Masyarakat karena hanya dua bagian ini yang mengatur tentang permasalahan di masyarakat termasuk di dalamnya adalah masalah pedagang kaki lima. Berikut adalah tugas dari bagian Penegakan Produk Hukum Daerah ini adalah :

1. Menyelenggarakan pengkajian program kerja Bidang Penegakan Produk Hukum Daerah
2. Menyelenggarakan pengkajian bahan fasilitasi penyusunan pedoman dan supervisi Penegakan Produk Hukum Daerah
3. Menyelenggarakan pengkajian bahan fasilitasi penyidikan pelanggaran Peraturan Daerah
4. Menyelenggarakan pengkajian bahan fasilitasi penindakan pelanggaran Peraturan Daerah
5. Menyelenggarakan pembinaan, pengawasan dan penyuluhan Penegakan Produk Hukum Daerah
6. Menyelenggarakan fasilitasi dan Penegakan Produk Hukum Daerah
7. Menyelenggarakan pengkajian bahan koordinasi penyelenggaraan Penegakan Produk Hukum Daerah
8. Menyelenggarakan koordinasi dan fasilitasi Penyidik Pegawai Negeri Sipil di Jawa Barat

9. Menyelenggarakan telaahan staf sebagai bahan pertimbangan pengambilan kebijakan
10. Menyelenggarakan pelaporan dan evaluasi kegiatan Bidang Penegakan Produk Hukum Daerah
11. Menyelenggarakan koordinasi dengan Badan Koordinasi Pemerintahan dan Pembangunan Wilayah dalam pelaksanaan kegiatan di Kota
12. Menyelenggarakan koordinasi dengan Organisasi Perangkat Daerah Provinsi lainnya dan Kota
13. Menyelenggarakan koordinasi dengan unit kerja terkait
14. Menyelenggarakan tugas lain sesuai tugas pokok dan fungsinya.

Adapun tugas dari bagian Bidang Ketertiban Umum dan Ketrentaman Masyarakat adalah :

1. Menyelenggarakan pengkajian program kerja Bidang Ketertiban Umum dan Ketenteraman Masyarakat
2. Menyelenggarakan pengkajian bahan fasilitasi penyusunan pedoman dan supervisi ketertiban umum dan ketenteraman masyarakat
3. Menyelenggarakan pengkajian bahan fasilitasi ketertiban umum dan ketenteraman masyarakat
4. Menyelenggarakan pengkajian bahan koordinasi penyelenggaraan ketertiban umum dan ketenteraman masyarakat serta kerjasama
5. Menyelenggarakan fasilitasi dan pemeliharaan ketertiban umum
6. Menyelenggarakan pengkajian bahan koordinasi penyelenggaraan ketertiban umum dan ketenteraman masyarakat
7. Menyelenggarakan kerjasama lintas provinsi
8. Menyelenggarakan telaahan staf sebagai bahan pertimbangan pengambilan kebijakan
9. Menyelenggarakan koordinasi dengan Badan Koordinasi Pemerintahan dan Pembangunan Wilayah dalam pelaksanaan kegiatan di Kota
10. Menyelenggarakan koordinasi dengan OPD Provinsi lainnya dan instansi terkait

11. Menyelenggarakan pelaporan dan evaluasi kegiatan Bidang Ketertiban Umum dan Ketenteraman Masyarakat
12. Menyelenggarakan koordinasi dengan unit kerja terkait
13. Menyelenggarakan tugas lain sesuai tugas pokok dan fungsinya

II.2 Landasan Teori

Bagian ini berisi tentang beberapa teori-teori yang mendukung dalam proses analisis dan implementasi berdasarkan masalah yang diangkat dalam pembangunan aplikasi *mobile* Pelaporan pedagang Kaki Lima Liar.

II.2.1 Sistem Informasi

Sistem informasi adalah sekumpulan elemen yang saling bekerja sama baik secara manual atau berbasis komputer yang didalamnya ada pengumpulan, pengolahan, pemprosesan dan penyimpanan data untuk mengasilkan informasi yang berguna untuk proses pengambilan keputusan [4].

II.2.1.1 Komponen sistem informasi

Komponen sistem informasi terdiri dari beberapa hal sebagai berikut [5]:

1. Blok Masukan

Masukan disini mewakili data yang masuk ke dalam sistem informasi. Masukan disini termasuk metode-metode dan media untuk menangkap data yang akan dimasukan, yang dapat berupa dokumen-dokumen dasar.

2. Blok Model

Blok ini terdiri dari prosedure, logika dan model matematik yang akan memanipulasi data input dan data yang tersimpan di basis data dengan cara yang sudah ditentukan untuk menghasilkan keluaran yang diinginkan.

3. Blok Keluaran

Blok keluaran menghasilkan informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen serta semua pemakai sistem.

4. Blok Teknologi

Teknologi merupakan “kotak alat” dalam sistem informasi. Teknologi digunakan untuk menerima masukan, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran dan membantu pengendalian dari sistem secara keseluruhan.

5. Blok Basis Data

Basis data merupakan kumpulan dari data yang saling berhubungan satu dengan lainnya, tersimpan di perangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya.

6. Blok Kendali

Banyak hal yang dapat merusak sistem informasi, seperti misalnya bencana alam, api, temperature, air, debu, kecurangan, kegagalan sistem, kesalahan-kesalahan, ketidak efisiensian, sabotase dan lain sebagainya.

II.2.2 Pedagang Kaki Lima

Pedagang kaki lima pertama kali muncul pada masa penjajahan belanda, dahulu Belanda membuat peraturan bahwa setiap jalan raya yang dibangun menyediakan sarana untuk pejalan kaki yang biasa disebut trotoar. Lebar trotoar untuk pejalan kaki yaitu lima kaki (kaki: satuan ukuran panjang yang digunakan mayoritas oleh bangsa Eropa) atau sekitar satu setengah meter. Kemudian saat Indonesia merdeka, trotoar untuk pejalan kaki itu dimanfaatkan oleh para pedagang untuk berjualan.

Sebenarnya asal usul istilah pedagang kaki lima masih simpang siur dan banyak versi. Menurut seorang tokoh Indonesianis bernama William Liddle, aturan trotoar lima kaki justru berasal dari bahasa inggris, five foot (lima kaki). Liddle mempercayai bahwa yang membuat aturan tentang pembangunan trotoar di Indonesia bukanlah Belanda, tetapi Inggris karena Inggris pernah mengambil kekuasaan atas Indonesia dari Belanda. Merunut Liddle gubernur jendral asal Inggris lah yang membuat trotoar di Indonesia yaitu Sir Stamford Raffles.

Menurut kamus umum bahasa Indonesia yang disusun oleh WJS Poerwadarminta istilah kaki lima itu mempunyai arti lantai (tangga) di muka pintu

atau di tepi jalan dan lantai diberi atap sebagai penghubung rumah dengan rumah. Walaupun banyak istilah dan versi tengtang asal usul pedagang kaki lima namun bisa disimpulkan bahwa pedagang kaki lima berarti pedagang yang berjualan, tapi tidak mempunyai kios atau toko. Kebanyakan pedagang kaki lima memilih berjualan di tempat keramaian, seperti di pasar, stasiun bis dan kereta, atau halte-halte dan tempat wisata.

Banyaknya orang yang mau menjadi pedagang kaki lima menurut para ahli ekonomi karena banyaknya tenaga kerja dan terbatasnya lapangan pekerjaan, hal itu yang menyebabkan banyaknya masyarakat yang menjadi pedagang kaki lima. Contohnya adalah peningkatan jumlah pedagang kaki lima di kota Bandung. Pada tahun 2000, jumlah pedagang kaki lima yang ada di kota Bandung telah mencapai 16.880 unit. Padahal, di tahun 1997 hanya 3000 unit pedagang kaki lima saja. Itu berarti, dalam waktu tiga tahun terjadi peningkatan lebih dari 500 kali lipat. Penyebab utamanya adalah banyaknya pemutusan hubungan kerja (PHK).

Jika disimpulkan yang menjadi penyebab banyak munculnya pedagang kaki lima adalah:

1. Kesulitan Ekonomi

Krisis keuangan yang terjadi tahun 1997-1999 itu menyebabkan harga-harga barang naik dengan begitu cepatnya (drastis). Orang juga banyak kehilangan pekerjaan atau menganggur lalu banyak diantara mereka memilih menjadi pedagang kaki lima.

2. Sempitnya Lapangan Pekerjaan

Masyarakat semakin banyak yang menganggur karena tidak adanya lapangan pekerjaan. Masyarakat lalu memilih menjadi pedagang kaki lima karena modalnya kecil dan tidak perlu punya kios atau toko, yang terpenting bagi mereka yaitu bisa mencari nafkah untuk keluarganya.

3. Urbanisasi

Masyarakat di desa berdatangan ke kota karena di desanya tidak ada pekerjaan dan kehidupannya miskin. Mereka berangkat ke kota tanpa modal pendidikan maupun keahlian dan akhirnya menjadi pedagang kaki lima [6].

II.2.3 Citizen Journalism

Citizen journalism atau jurnalisme warga merupakan kegiatan dimana peran dari wartawan atau kegiatan jurnalistik bisa dilakukan oleh masyarakat yang secara formal bukan wartawan. Kegiatan yang dilakukannya sama dengan wartawan pada umumnya, yakni mengumpulkan informasi, menulis berita, mengedit dan menyiarkannya. Dalam menyiarkan informasinya, citizen journalism bisa dengan mengirim tulisannya kepada media massa seperti koran atau media online, kemudian redaksi memutuskan apakah tulisan tersebut layak atau tidak untuk dipublikasikan melalui media massanya [7].

II.2.3.1 Prinsip-Prinsip Journalism

Ada beberapa prinsip dalam jurnalisme, berikut adalah prinsip-prinsip dalam jurnalisme [8]:

1. Akurat

Keakuratan dari sebuah berita adalah titik awal dari semua kegiatan jurnalis atau singkatnya adalah kegiatan memeriksa kembali fakta yang sudah didapatkan sebelumnya. Akurat disini berarti juga dapat memisahkan apakah sebuah informasi itu berupa fakta atau fiktif belaka.

2. Ketelitian

Kecermatan atau ketelitian dalam penyampaian sebuah berita berarti teliti dalam menerima informasi, teliti dalam mempercayai sumber informasi. Dapat disimpulkan disini bahwa untuk satu berita atau topik yang akan dipublikasikan harus dipelajari dari berbagai sumber informasi, artinya tidak hanya merujuk kepada satu informasi saja.

3. Kejujuran

Bagi semua jurnalis yang baik kejujuran adalah suatu keharusan. Di mana satu berita yang dipublikasikan harus seimbang dan pembaca tidak merasa dibohongi karena fakta sebenarnya tidak sesuai dengan berita yang ditampilkan. Kejujuran di sini juga berarti mau mendengarkan orang lain dalam berpendapat tentang informasi yang disampaikan, sehingga dapat diketahui salah atau benarnya suatu informasi dari sudut pandang orang lain.

4. Transparan

Penyajian sebuah berita seharusnya berasal dari berbagai informasi guna mendukung keakuratan berita tersebut. Oleh karena itu transparan disini berarti berbagai sumber informasi tersebut harus ditampilkan jika memang sebuah keharusan, agar tidak ada permasalahan dikemudian hari.

5. Kebebasan

Penyajian sebuah berita seharusnya tidak terikat oleh kelompok mana pun karena tekanan dari pihak lain salah satunya dapat digunakan untuk menghakimi pihak lainnya.

II.2.4 Android

Android adalah sebuah sistem operasi untuk perangkat *mobile* berbasis linux yang mencakup sistem operasi, middleware dan aplikasi. Android menyediakan sistem operasi yang terbuka bagi para pengembang untuk menciptakan aplikasi Android diluncurkan untuk umum pada musim gugur di tahun 2008. Android sangat berkembang pesat di industri karena dua aspek utama yaitu bersifat opensource dan model arsitekturya. Sebagai sebuah proyek yang bersifat open source, memungkinkan android untuk sepenuhnya dipahami dan dianalisis mengenai fitur, penyelesaian pada bug program hingga hardware [9].

II.2.5.1 Komponen Aplikasi Android

Aplikasi android terdiri dari komponen yang bebas digabungkan, diikat menggunakan suatu *project manifest* yang menggambarkan masing-masing komponen dan bagaimana komponen-komponen itu saling berhubungan. Ada enam komponen dalam android untuk membangun suatu aplikasi yaitu [10]:

1. *Activities*

Activity merupakan bagian yang paling penting dalam sebuah aplikasi, karena *activity* menyajikan tampilan *visual* program yang sedang dijalankan oleh *user.* Setiap *activity* dideklarasikan dalam sebuah *class* yang bertugas untuk menampilkan antarmuka yang terdiri dari *views* dan repon terhadap *event.* Setiap aplikasi mempunyai satu buah *activity* atau lebih. Biasanya pasti akan ada *activity*

yang pertama kali tampil ketika aplikasi sedang dijalankan. Perpindahan antara *activity* dengan *activity* lainnya diatur melalui sistem, dengan memanfaatkan *activity stack.* Keadaan suatu *activity* ditentukan oleh posisinya pada *activity stack.* Bila suatu *activity* baru dimulai, *activity* yang sebelumnya digunakan maka akan dipindahkan ke tumpukan paling atas. Ketika *activity* dipanggil dan disimpan dalam *activity stack* terdapat 4 kemungkinan kondisi transisi yang akan terjadi [10]:

1. *Active*

   Setiap *activity* yang berada di tumpukan paling atas, maka *activity* tersebut akan terlihat dan menerima masukan dari *user*. Android akan membuat *activity* ini tetap hidup dengan menghentikan *activity* yang berada dibawah tumpukan jika diperlukan. Ketika *activity* sedang aktif, maka yang lainnya akan dihentikan sementara.

2. *Paused*

   Beberapa *activity* akan terlihat tapi tidak terfokus pada kondisi ini disebut *paused*. Keadaan ini terjadi jika *activity* transparan dan tidak fullscreen pada layar. Ketika *activity* dalam keadaan *paused*, dia terlihat aktif namun tidak bisa diakses oleh *user*.

3. *Stopped*

   Ketika sebuah *activity* tidak terlihat, maka itulah yang disebut *stopped*. *Activity* akan tetap berada dalam memori dengan semua keadaan dan informasi yang ada. Namum akan di eksekusi oleh sistem jika sistem membutuhkan sumberdaya lebih.

4. *Inactive*

   Kondisi ini ketika *activity* telah dihentikan atau sebelum dijalankan. Inactive *activity* teh ditiadakan dari *activity stack* sehinga perlu restart ulang gar dapat tampil dan digunakan kembali.

2. *Services*

Suatu *service*s tidak memiliki tampilan antarmuka, melainkan berjalan di background untuk waktu yang tidak terbatas. Komponen *service* diproses tidak terlihat, memperbaharui sumber data dan menampilkan notifikasi.

3. *Content Providers*

Content providers digunakan untuk mengelola dan berbagi *database*. Data dapat disimpan dalam *file* sistem, dalam *database SQLite*, atau dengan cara lain yang pada prinsipnya sama. Dengan adanya *content provider* memungkinkan antar aplikasi untuk saling berbagi data.

4. *Intens*

Intens merupakan sebuah mekanisme untuk menggambarkan tindakan tertentu, seperti memilih foto, menampilkan halaman *web,* dan lain sebagainya. *Intens* tidak selalu dimulai dengan menjalankan aplikasi, namun juga digunakan oleh sistem untuk memberitahukan ke aplikasi bila terjadi sesuatu, semisal pesan masuk.

5. *Broadcast Receivers*

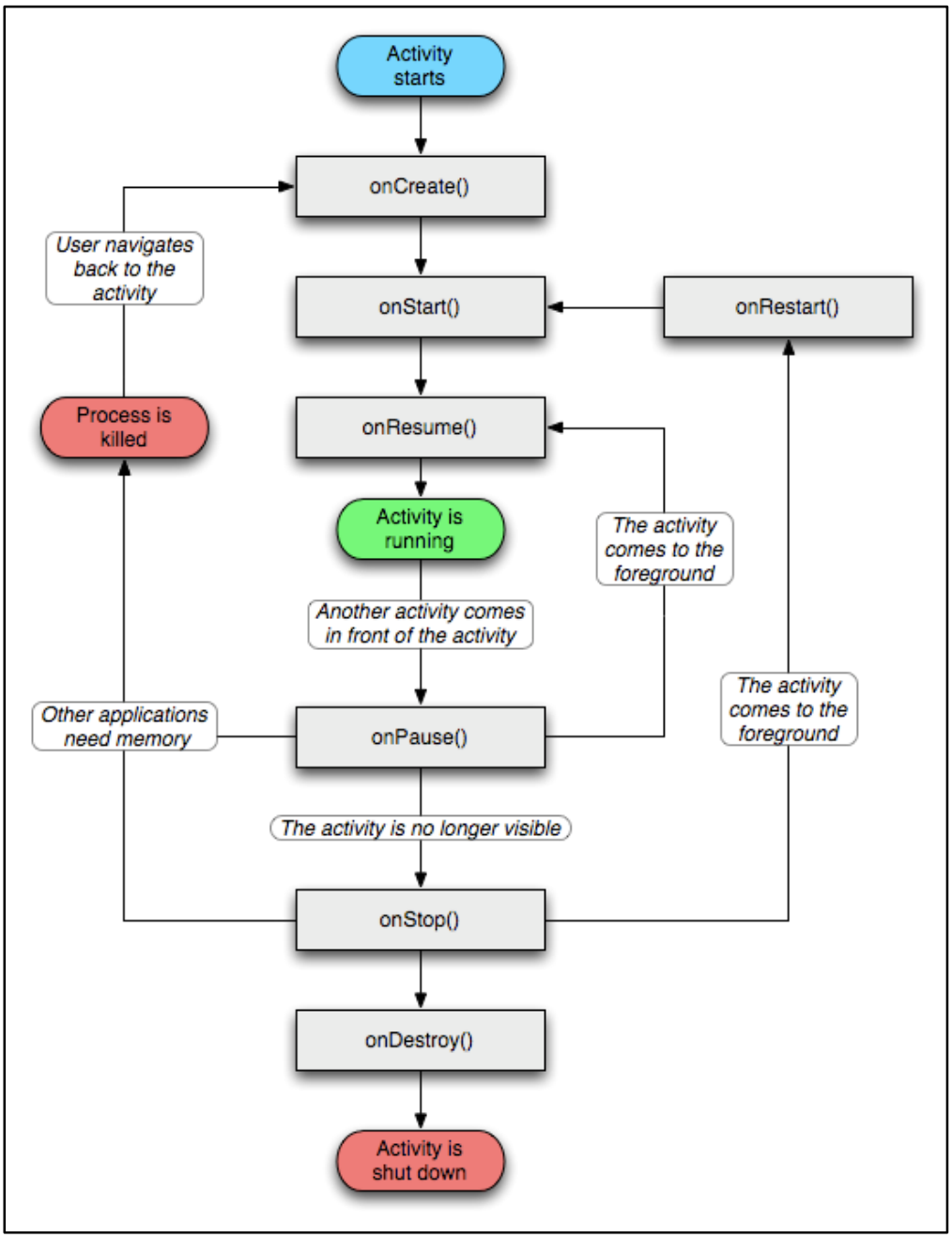
Broadcast receivers merupakan komponen yang sebenarnya tidak melakukan apa-apa kecuali menerima dan bereaksi menyampaikan pemberitahuan. Sebagian besar *broadcast* berasal dari sistem contohnya baterai sudah hampir habis, informasi zona waktu sudah berubah, atau *user* telah mengganti bahasa *default* pada perangkat android. Sama halnya dengan *service*, *broadcast receivers* tidak menampilakan antarmuka *user*. Namun, *broadcast receivers* dapat menggunakan *notification manager* untuk memberitahukan sesuatu kepada *user*.

6. *Notifications*

Notifications merupakan kerangka pemberitahuan untuk *user*. *Notifications* memberikan isyarat atau peringatan tanpa mengganggu aktivitas yang sedang dilakukan oleh *user.*

II.2.5.2 Siklus Hidup Aplikasi *Android*

Siklus hidup aplikasi android dikelola oleh sistem, berdasarkan kebutuhan *user* dan sumber daya yang tersedia. Sistem sangat berperan dalam menentukan apakah aplikasi dijalankan, dihentikan sementara, atau dihentikan sama sekali. Jika *user* sedang menjalankan sebuah *activity,* maka sistem akan memprioritaskan aplikasi tersebut. Sebaliknya, jika suatu *activity* tidak terlihat dan sistem membutuhkan sumber daya yang lebih, maka *activity* yang prioritas rendah akan ditutup [11]. Berikut adalah siklus hidup activity android pada gambar II.1:



Gambar II. 2 Siklus Hidup *Activity* Android [12]

User mengakses *method* ini pada *activity class*, dan android akan memanggilnya menurut waktu yang tepat [12]:

1. *onCreate( )*

Dipanggil ketika *activity* pertama kali dibuat, disini *user* melakukan pengaturan statis secara normal, misal membuat *view*, memasukan data dan sebagainya. Selalu diikuti dengan *onStart( ).*

2. *onStart( )*

Dipanggil sebelum *activity* terlihat oleh *user.* Selalu diikuti dengan *onResume( ).*

3. *onResume( )*

Dipanggil sebelum *activity* mulai berinteraksi dengan *user*. Pada titik ini, *activity* berada pada *stack activity* yang paling atas. Selalu diikuti dengan *onPause( ).*

4. *onPause( )*

Dipanggil ketika sistem akan memulai *activity* lain. *Method* ini biasa digunakan untuk melakukan perubahan yang tidak disimpan kedalam data yang tetap, menghentikan animasi, dan hal lain yang memakai *CPU*. Diikuti *onResume( )* jika *activity* tetap berada dilayar atau *onStop( )* jika *activity* menjadi tidak terlihat oleh *user*.

5. *onStop( )*

Dipanggil ketika *activity* tidak lagi terlihat oleh *user*. Ini mungkin terjadi karena *activity* kemudian dihancurkan atau jika *activity* lain yang sudah ada atau yang baru telah dilanjutkan. Diikuti dengan *onRestart( )* jika *activity* kembali berinteraksi dengan *user* atau dengan *onDestroy( )* jika *activity* akan ditutup.

6. *onRestart( )*

Dipanggil setelah *activity* dihentikan, dan merupakan prioritas untuk dijalankan kembali dan selalu diikuti *onStart( ).*

7. *onDestroy( )*

Dipanggil sebelum *activity* dihancurkan, ini merupakan panggilan terakhir yang diterima *activity*. Bisa dikatakan *activity* diakhiri (*finish( )*), atau karena sistem secara *temporary* menghancurkan *instant activity* untuk menghemat ruang.

8. *onSaveInstanceState( )*

Android akan memanggil *method* ini untuk mengizinkan *activity* menyimpan setiap contoh keadaan, seperti posisi kursor dalam sebuah *text field*.

9. *onRestoreInstanceState( )*

Dipanggil ketika *activity* di inisialisasi ulang sejak keadaan sebelum disimpan dari *onSaveInstanceState( ).*

Android menjalankan setiap aplikasi dalam proses secara terpisah, yang masing-masing memiliki mesin *virtual* pengolah sendiri, dengan melindungi penggunaan memori pada aplikasi. Selain itu juga android dapat mengontrol aplikasi mana yang layak menjadi prioritas utama. Karena itu android menjadi sangat sensitif dengan siklus hidup aplikasi dan komponen-komponennya.

II.2.5 JSON (Javascript Object Notation)

JSON (JavaScript Object Nation) adalah format pertukaran data ringan, mudah dibaca dan ditulis oleh manusi a, serta mudah diterjemahkan dan dibuat (generate) 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 apapuun karena menggunakan gaya bahasa yang umum digunakan oleh programmer keluarga C termasuk C, C++, C#, Java, JavaScript, Perl, Phyton dll. Oleh karena sifat-sifat tersebut , menjadikan JSON ideal sebagai bahasa pertukaran data [13].

II.2.6 API (Application Programming Interface)

Application programming interface (API) adalah sekumpulan perintah, fungsi, komponen, dan protokol yang disediakan oleh sistem operasi ataupun bahasa pemograman tertentu yang dapat digunakan oleh programmer saat membangun perangkat lunak.

Dalam suatu pemograman dibutuhkan setidaknya ribuan *system call* per detik oleh karena itu banyak programmer yang menggunakan API. Didalam API terdapat fungsi-fungsi/perintah untuk menggantikan bahasa yang digunakan dalam s*ystem calls* dengan bahasa yang lebih mudah dimengerti oleh programmer. Fungsi yang dibuat dengan menggunakan API tersebut kemudian akan memanggil *system call* sesuai dengan sistem operasinya. Tidak tertutup kemungkinan nama dari *system call* sama dengan nama API [14].

Kelebihan pemograman menggunakan API adalah:

1. Portabilitas

Programmer yang menggunakan Api dapat menjalankan programnya dengan sistem operasi mana saja asalkan sudah terinstal API tersebut [14].

2. Mudah Dimengerti

API menggunakan bahasa yang lebih terstruktur dan mudah dimengerti daripada bahasa *system call* [14].

II.2.7 LBS (Location Based *Service*)

Location based service adalah *service* yang berfungsi untuk mencari dengan teknologi *Global Positioning Service* (GPS) dan *Google's Cell-based Location*. *Map* dan layanan berbasis lokasi menggunakan garis lintang dan garis bujur untuk menentukan letak geografis, namun sebagai user kita membutuhkan alamat atau posisi real time kita bukan nilai garis lintang dan garis bujur. Android menyediakan *geocoder* yang mendukung *forward* dan *reverse geocoding*. Menggunakan *geocoder*, anda dapat mengkorversi nilai garis lintang dan garis bujur menjadi alamat dunia nyata atau sebaliknya [15].

II.2.8 Structure Analysis and Design Technique

Structure analysis and design technique atau SADT adalah metode untuk pengembangan sistem terstruktur yang dikembangkan pada tahun 1969 dan dikembangkan lebih lanjut pada tahun 1974. SADT membagi sistem kedalam 2 hal yaitu [16]:

1. Sebagai benda (*object*) : objek, dokumen dan data
2. Sebagai kejadian (*event*) : kegiatan yang dilakukan manusia, perangkat lunak dan mesin

II.2.8.1 Diagram Konteks

Diagram konteks adalah diagram yang terdiri dari suatu proses dan menggambarkan ruang lingkup suatu sistem. Diagram konteks merupakan level tertinggi dari Data flow diagram yang menggambarkan seluruh input ke sistem atau output dari sistem. diagram konteks memberi gambaran tentang keseluruan sistem.

Sistem dibatasi oleh boundary (dapat digambarkan dengan garis putus). Dalam diagram konteks hanya ada satu proses [17].

II.2.8.2 Data Flow Diagram

Data flow diagram atau DFD adalah alat untuk menggambarkan komponen-komponen sebuah sistem, aliran data pada sistem, dimana data berasal, tujuan data serta penyimpanan data. Ada 4 komponen utama dalam DFD yaitu [5]:

1. Entitas eksternal (*External Entity*)
2. Arus data (*Data Flow*)
3. Proses (*Process*)
4. Penyimpanan data (*Data Store*)

DFD dapat dikatakan turunan dari diagram konteks, analisis lebih detail dari sistem dapat dilakukan dengan menurunkan DFD ke level 1,2,3 dan seterusnya. Penjelasan simbol pada diagram konteks dan DFD dapat dilihat di daftar simbol.

II.2.8.3 Spesifikasi Proses

Spesifikasi proses digunakan untuk mendeskripsikan proses yang terjadi pada level yang paling dasar dalam DFD. Spesifikasi Proses berfungsi mendeskripsikan tahapan yang dilakukan untuk merubah masukan menjadi keluaran. Spesifikasi proses ditampilkan dalam bentuk form, logika proses yang ditulis pada spesifikasi proses harus jelas [18].berikut adalah tabel II.1 bentuk contoh form dari spesifikasi proses :

Tabel II. 1 Form Spesifikasi Proses [19]

No	Proses	Keterangan
1.	No Proses	(Menyatakan nomor proses)
	Nama Proses	(Menyatakan nama proses)
	Sumber	(Menyatakan sumber data yang menuju ke proses)
	Input	(Menyatakan isi data yang masuk ke proses)
	Output	(Menyatakan informasi yang keluar dari proses)
	Tujuan	(Menyatakan tujuan informasi keluaran dari proses)
	Logika Proses	(Menyatakan algoritma dari proses)

II.2.8.4 Kamus Data

Kamus data adalah deskripsi formal mengenai seluruh komponen-komponen yang ada pada data flow diagram. Pada tahap perancangan, komponen-komponen pada kamus data akan menjadi bahan untuk menyusun basis data [20]. Berikut adalah contoh tabel II.2 contoh form kamus data :

Tabel II. 2 Form Kamus Data [19]

Nama aliran data	(Data yang digunakan)
How used	(Proses apa saja yang digunakan)
Keterangan	(Uraian singkat data yang digunakan)
Struktur data	(Daftar komponen yang ada pada data yang digunakan)
Deskripsi	(Jenis data dalam representasi komputer untuk masing-masing data.)

II.2.9 Object Oriented Analysis and Design

Konsep OOAD mencakup analisis dan desain sebuah sistem dengan pendekatan objek, yaitu analisis berorientasi objek (OOA) dan desain berorientasi objek (OOD). OOA adalah metode analisis yang memerika requirement (syarat/keperluan) yang harus dipenuhi sebuah sistem) dari sudut pandang kelas-kelas dan objek-objek yang ditemui dalam ruang lingkup sistem. Sedangkan OOD adalah metode untuk mengarahkan arsitektur software yang didasarkan pada manipulasi objek-objek sistem atau subsistem [21].

II.2.9.1 Unified Modelling Language

Unified Modeling Language (UML) adalah termasuk ke dalam rumpun jenis pemodelan notasi grafis yang didukung oleh meta-model tunggal, Pemodelan ini berguna untuk membantu dalam menjelaskan dan merancang perangkat lunak yang dibangun dengan object-oriented (OO). UML merupakan standar terbuka yang dikelola oleh Open Management Group (OMG) yang berada dibawah naungan perusahaan-perusahaan konsorsium terbuka [21]. UML merupakan suatu bahasa pemodelan yang terdiri banyak model diantaranya adalah :

1. Use case Diagram

Use case adalah teknik untuk menggambarkan fungsional sebuah sistem. *Use case* diagram memperlihatkan hubungan diantara *actor* dan *use case*. Aktor mempresentasikan seorang user atau subsistem lain yang akan berinteraksi dengan sistem. Sedangkan *use case* merupakan urutan kejadian yang menggambarkan interaksi antara user dengan sistem. Fungsionalitas sistem didefinisikan ke dalam *use case* dari sudut eksternal sistem yang berguna untuk diuji kelayakan sistem [21].

2. Sequence Diagram

Sequence diagram menggambarkan interaksi diagram yang menunjukan bagaimana kelompok-kelompok objek saling berkolaborasi dalam beberapa behavior, Sebuah *Sequence diagram* secara khusus menjabarkan behavior sebuah skenario tunggal. Diagram tersebut menunjukan sejumlah objek contoh dan pesan-pesan yang melewati objek ini dalam *use case*. *Sequence diagram* adalah suatu diagram yang memperlihatkan atau menampilkan interaksi-interaksi antar objek di dalam sistem yang disusun pada sebuah urutan atau rangkaian waktu. Interaksi antar objek tersebut termasuk pengguna, display, dan sebagainya berupa pesan/message. *Sequence diagram* digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/even untuk menghasilkan output tertentu [21].

3. Class diagram

Class diagram mendeskripsikan jenis-jenis objek dalam sistem dan berbagai macam hubungan statis yang terdapat diantara mereka. Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendifinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi. Atribut merupakan variable-variabel yang dimiliki oleh suatu kelas, atribut mendeskripsikan properti dengan sebaris teks di dalam kotak kelas tersebut. Operasi atau metode adalah fungsi fungsi yang dimiliki oleh suatu kelas [21].

II.2.10 Skala Likert

Skala *likert* merupakan salah satu metode yang dapat digunakan untuk mendapatkan hasil pengujian angket atau kuesioner. Skala *likert* sering digunakan untuk mengukur sikap, pendapat, dan presepsi seseorang atau sekelompok orang tentang suatu gejala atau fenomena. Skala *likert* meminta kepada responden untuk menjawab suatu pernyataan dengan jawaban sangat setuju (SS), setuju (S), belum memutuskan (N), tidak setuju (T), dan sangat tidak setuju (ST). Untuk masing-masing jawaban mempunyai bobot nilai yang berbeda, adapun pemberian bobot nilai itu sendiri tergantung jenis pernyataan yang dapat dilihat pada Tabel II.3 berikut [22]

Tabel II. 3 Tabel Pembobotan Nilai Skala *Likert*

Jenis Pertanyaan	Bobot Pendapat				
	SS	S	N	T	ST
Positif	5	4	3	2	1
Negatif	1	2	3	4	5

Ketika jawaban sudah dihitung berdasarkan bobot nilainya masing-masing, maka dilanjutkan menghitung rata-rata dari setiap konten pernyataan dan indikator pernyataan itu sendiri dengan menggunakan rumus:

$$x = \sum_{i=1}^{n} xi/n$$

Dimana n adalah jumlah responden, sedangkan x adalah jumlah konten nilai ataupun konten pernyataan. Sedangkan untuk menghitung rata-rata dari indikator, n akan berupa jumlah konten pernyataan, sedangkan x adalah jumlah dari konten indikator pernyataan [22].