BAB II
LANDASAN TEORI

II.1. Sejarah *Barbershop*

Pelayanan jasa perawatan pria sudah ada sejak 2000 tahun yang lalu. Praktek pemotongan rambut pada pria (*barbershop*) berawal dari Wilayah Macedonia sekitar 400 tahun sebelum masehi lalu menyebar ke Mesir dan daerah-daerah lainnya. Kata "barber" berasal dari bahasa latin "barba" yang artinya janggut. Bangsa pertama yang mengklaim dirinya paling ahli dalam jasa pelayanan pemotongan rambut adalah bangsa Roma sekitar 296 tahun sebelum masehi. Akan tetapi baik pada bangsa Roma maupun Mesir, *Barbershop* memiliki reputasiyang kurang baik karena orang-orang elit kelas atas pada waktu itu memiliki tukang cukur pribadi. Pada masa itu, janggut pada lelaki menjadi symbol kekuatan dan intelegensi sehingga harus dirawat dengan baik dan teratur.

Barbershop modern mulai berdiri sekitar awal abad ke-20 di wilayah Amerika Serikat. Pada tahun 1920 “Associated Master Barbers of America” dan “Nacional Association of Barber School” menjadi 2 organisasi formal yang mengatur profesi ini. Kecepatan dan efisiensi mencukur juga semakin baik seiring dengan berkembangnya teknologi, misalnya dengan penggunaan berbagai alat-alat elektronik pandukung seperti kliper maupun blowdryer[5].

II.2. Pemesanan (*Booking*)

Menurut penulis pemesanan (*booking*) adalah sebuah proses perjanjian berupa pemesanan sebuah produk baik barang maupun jasa. Sedangkan menurut Kamus Besar Bahasa Indonesia pemesanan adalah “ proses, perbuatan, cara memesan (tempat,barang,dsb) kepada orang lain.”

II.3. Sistem Informasi

Sistem adalah kumpulan dari komponen-komponen peralatan model *Requirement*s, function dan *interface*. Sistem Informasi merupakan kombinasi teratur apapun dari sumber daya manusia, hardware, software, data, dan jaringan. Daur hidup suatu sistem meliputi[6]:

1. Investigasi
2. Analisis
3. Desain
4. Implementasi
5. Pemeliharaan

II.3.1. Komponen Sistem Informasi

1. Komponen Masukan

   Mewakili sejumlah data yang masuk ke dalam sistem informasi. Input tersebut termasuk metode-metode dan media untuk memperoleh data yang akan dimasukkan, yang dapat berupa dokumen-dokumen dasar.
2. Komponen Proses

   Terdiri dari kombinasi prosedur, logika, dan model matematika yang akan memanipulasi data input dan data yang tersimpan di database dengan cara tertentu untuk menghasilkan keluaran (output) yang diinginkan.
3. Komponen Keluaran

   Produk dari sistem informasi adalah keluaran yang merupakan informasi dan dokumentasi yang dapat digunakan untuk semua tingkatan manajemen dan semua pemakai sistem.
4. Komponen Teknologi

   Teknologi merupakan tool-box dalam sistem informasi. Teknologi digunakan untuk menerima input, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran dan membantu pengendalian dari *system* secara keseluruhan.

5. Komponen Hardware

   Hardware berperan penting sebagai suatu peralatan vital bagi sistem informasi yang digunakan untuk menyimpan dan mengolah informasi.

6. Komponen Software

   Software berfungsi sebagai kumpulan perintah komputerisasi yang berguna untuk mengolah, menghitung dan memanipulasi data yang diambil dari hardware untuk menciptakan suatu informasi.

7. Komponen Basis data

   Database merupakan kumpulan dari data yang saling berhubungan satu sama lainnya, tersimpan pada perangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya. Pengelolaan database umumnya dikenal dengan nama DBMS (Database *Management System*).

8. Komponen Kendali

   Bagian pengendalian dirancang dan diterapkan untuk memelihara *system* dari hal-hal yang dapat merusaknya, seperti faktor-faktor alamiah temperatur, air, api, debu, virus, sabotase/hijacking, dan sebagainya.

II.4. Android

Android adalah sistem operasi berbasis Linux yang dirancang untuk perangkat seluler layar sentuh seperti telepon pintar dan komputer *tablet*. Android awalnya dikembangkan oleh Android, Inc., dengan dukungan finansial dari Google, yang kemudian membelinya pada tahun 2005. Sistem operasi ini dirilis secara resmi pada tahun 2007, bersamaan dengan didirikannya *Open* Handset Alliance, konsorsium dari perusahaan-perusahaan perangkat keras, perangkat lunak, dan telekomunikasi yang bertujuan untuk memajukan standar terbuka perangkat seluler. Ponsel Android pertama mulai dijual pada bulan Oktober 2008[7].

II.4.1. Android Life Cycle

Aplikasi android terdiri dari beberapa fungsi dasar seperti mengedit catatan, memutar file musik, membunyikan alarm, atau membuka kontak telepon. Fungsi-

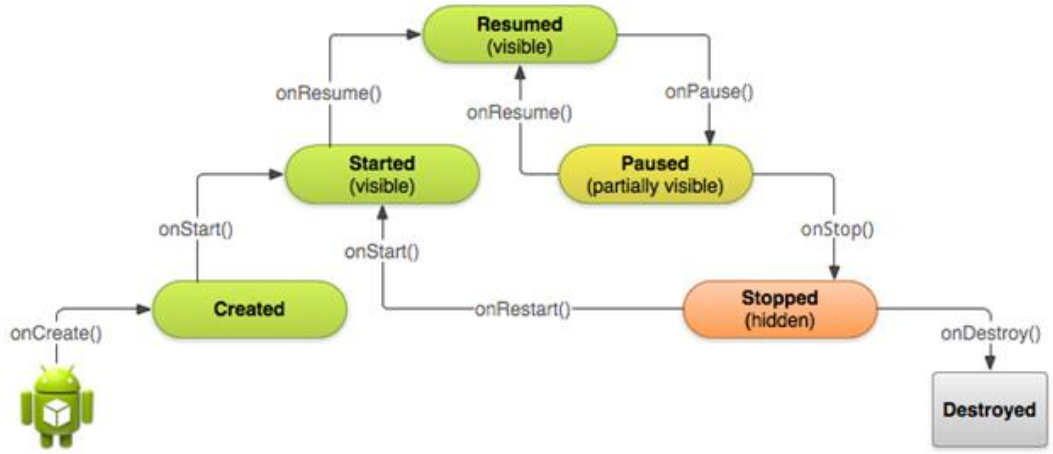
fungsi tersebut dapat diklasifikasikan ke dalam empat komponen android yang berbeda seperti ditunjukkan pada , klasifikasi tersebut berdasarkan kelas-kelas dasar java yang digunakan[8].

Tabel II.1 Komponen Aplikasi Android

Functionality	***Java Base Class***	***Examples***
Focused thing a user can do	*Activity*	*Edit a note, play a game*
Background process	*Service*	*Play music, update weather icon*
Receive messages	*BroadcastReceiver*	*Trigger alarm upon event*
Store and retrieve data	*ContentProvider*	*Open a phone contact*

Setiap aplikasi pasti menggunakan minimal satu dari komponen tersebut, akan tetapi terdapat beberapa komponen yang mengharuskan mencantumkan *specified permission* sebelum digunakan seperti komponen *Service, BroadcastReceiver. ContentProvider*[7].

Android memiliki paradigma pemrograman lain tidak seperti paradigma pemrograman biasa di mana aplikasi yang dijalankan pada fungsi main(), sistem android menjalankan kode dalam method Activity dengan menerapkan metode callback tertentu yang sesuai dengan tahap tertentu dari siklus hidup. Setiap aplikasi yang berjalan dalam sistem operasi android memiliki siklus hidup yang berbeda dengan aplikasi desktop atau web, Hal ini dikarenakan aplikasi mobile memiliki tingkat interupsi proses yang lumayan tinggi seperti ketika handling panggilan masuk aplikasi diharuskan menghentikan proses sementara, Penerapan siklus hidup juga berguna untuk memastikan aplikasi tidak menghabiskan sumber daya baterai pengguna.



Gambar II-1 Siklus Hidup Android[7]

Terdapat beberapa state dalam siklus hidup android yang terjadi seperti diilustrasikan pada Gambar II 1 Siklus Hidup Android, akan tetapi hanya beberapa dari state tersebut yang menjadi statis diantaranya:

1. **Resumed**

   Resumed terjadi ketika aplikasi berjalan setelah state paused . State ini akan menjalankan perintah program yang ditulis pada method onResume()[7].
2. **Paused**

   Dalam keadaan ini aktivitas yang terjadi dihentikan secara sementara tetapi masih terlihat oleh pengguna karena terdapat proses yang memiliki prioritas lebih tinggi seperti panggilan telepon. Aplikasi tidak dapat menjalankan perintah apapun ataupun menampilkan apapun dalam state ini[7].
3. **Stopped**

   Dalam keadaan ini, aplikasi benar-benar tidak ditampilkan dan tidak terlihat oleh pengguna tetapi masih meninggalkan *service* dibackground[7].

State lain seperti *Created* dan Started bersifat sementara dan sistem dengan cepat menjalankan state berikutnya dengan memanggil metode life cycle callback berikutnya. Artinya, setelah sistem OnCreate() dipanggil, dengan cepat sistem akan memanggil method OnStart(), krmudian diikuti oleh onResume()[8].

II.4.2. Fitur

Android memiliki beberapa fitur utama yang sering digunakan dalam proses pembangunan aplikasi diantaranya adalah:

1. Multi-proses dan *App Widgets*

   Sistem operasi android tidak melarang prosesor menjalankan lebih dari satu aplikasi dalam satu waktu. Sistem operasi android dapat mengatur aplikasi dan thread yang berjalan secara multitasking. Keuntungan yang didapat adalah ketika aplikasi berjalan dan berinteraksi dengan pengguna di layer depan sistem operasi, proses dari aplikasi lain dapat berjalan untuk melakukan pembaruan informasi. Sebagai contoh misalnya ketika pengguna memainkan game, proses lain dapat berjalan di belakang aplikasi seperti memeriksa harga saham dan memunculkan peringatan[7].

   App Widgets adalah mini aplikasi yang dapat embedded dalam aplikasi seperti home screen. *App widgets* dapat menjalankan proses *request* seperti musik streaming atau mendeteksi suhu ruangan secara background[7].

   Multi-proses dapat memberikan manfaat berupa *user* experience yang lebih banyak, namun penggunaan fitur tersebut dapat menghabiskan banyak energi baterai jika penggunaan tidak benar[8].

2. *Touch* Gestures dan Multi-*touch*

   *Touch*screen adalah *user interface* intuitif yang digunakan banyak *smartphone* di dunia. Dengan fitur ini interaksi dapat dibuat lebih mudah karena cukup dengan menggunakan jari tangan. Multi-*touch* adalah kemampuan yang dapat melakukan tracking lebih dari satu tangan dalam satu waktu, Fitur ini sering digunakan untuk interaksi memperbesar atau memutar objek. Selain itu pengembang dapat membuat interaksi baru dengan memanfaatkan fitur tersebut[8].

3. Hard dan Soft Keyboard

   Salah satu fitur pada perangkat *smartphone* adalah tombol fisik dan non fisik, tombol fisik digunakan untuk navigasi pendukung dalam

pengoperasian android.. Pengembang aplikasi tidak perlu secara manual untuk mengintegrasikan tombol tersebut dalam aplikasi. Tombol non fisik adalah tombol yang dibuat oleh sistem operasi seperti keyboard virtual, dan tombol navigasi aplikasi[8].

II.4.3. Prinsip Desain

Android memiliki beberapa prinsip desain yang dapat menjadi acuan dalam membuat desain aplikasi android diantaranya adalah

1. Multiple Assets

   Android mendukung jutaan *smartphone*, *table*t dan perangkat lain dalam berbagai ukuran layar dan ukuran, untuk itu Multiple Assets sangat disarankan digunakan untuk mengatasi fragmentasi pada android. Seperti ilustrasi pada Gambar II 2 Klasifikasi Ukuran Ikon, android menciptakan beberapa klasifikasi ukuran icon yaitu MDPI, HDPI, XHDPI, XXHDPI dan XXXHDPI. MDPI dan HDPI dikhususkan untuk icon yang akan digunakan pada device berukuran *smartphone* sedangkan untuk XHDPI, XXHDPI dan XXXHDPI digunakan pada device berukuran *table*t[9].



Gambar II-2 Klasifikasi Ukuran Ikon

2. *Touch* Feedback

   Touch Feedback dalam android digunakan sebagai respon setiap objek yang ditekan pengguna. Hal ini bertujuan untuk memberi tahu pengguna objek mana yang berinteraksi dengan pengguna[9].



Gambar II-3 *Touch* Feedback

3. Pattern Gesture

   Touch gesture berguna untuk meningkatkan experience pengguna dalam menggunakan aplikasi. Terdapat beberapa gesture yang didukung oleh android diataranya adalah[9]:

4. *Touch*

   Touch adalah gesture default sebagai event untuk item tertentu yang paling sering digunakan dalam perangkat android, hampir seluruh kontrol utama aplikasi ini menggunakan *touch*.

5. *Long Press*

   Biasanya digunakan untuk seleksi data , dengan gesture ini dimungkinkan untuk memilih satu atau lebih item dalam sebuah tampilan dan menjalankan suatu fungsionalitas tertentu[9].

6. *Swipe Or Drag*

   Swipe adalah menyentuh sebuah titik pada layar dan menggerakkan jari yang tetap tersentuh pada layar ke titik lain pada layar. Swipe biasanya dilakukan untuk kebutuhan berpindah layar atau tergantung dari aplikasi. Swipe dapat dilakukan dari dan ke arah mana saja[9].

7. *Double Touch*

   Pada *Smartphone* dan *tablet* android, melakukan dua kali t*app*ing secara berturut-turut pada satu obyek, fungsinya berbeda dengan *double* klik mouse komputer. Pada Android, teknik ini biasanya dipakai untuk melakukan zoom in dan zoom out atau memperbesar / memperkecil sebuah obyek gambar[9].

8. Pinch *Open*

   Teknik lain yang biasa digunakan adalah dengan menggunakan dua jari, di mana kedua jari tersebut menyentuh dua titik pada layar yang terpisah di mana ujung dari dua jari tersebut tidak bersentuhan, kemudian kedua jari tersebut sambil tetap menyentuh layar bergerak saling mendekati. Teknik ini digunakan untuk membuka aplikasi tertentu[9].

9. Pinch Close

   Teknik Pinch Close adalah kebalikan dari Pinch *Open* di mana spread dilakukan dengan berawal dari dua jari yang bersentuhan pada ujungnya dan ditempelkan sebuah titik yang sama pada layar, kemudian kedua jari tersebut bergerak memisahkan atau menjauhi satu sama lain Gerakan ini untuk menutup layar[9].

II.5. *User-Generated Content*

User-generated Content atau disingkat UGC, dikenal juga dengan istilah consumer-*generated* media (CGM) atau *user-created content* (UCC) adalah merujuk pada berbagai tipe konten materi atau media yang tersedia untuk umum dan diproduksi oleh end-*user* atau para non-media profesional. Kemajuan teknologi web memungkinkan konten website tidak lagi dimonopoli oleh profesional media, melainkan dapat dibuat oleh para penggunanya. UGC merupakan salah satu ciri dominan Web 2.0. Contoh penerapan pada UGC adalah YouTube hampir semua konten yang dimiliki situs tersebut dibuat dan diupload oleh penggunanya[10].

Konsep UGC pada dasarnya telah banyak merubah cara berinteraksi pengguna dengan internet begitu juga dalam media periklanan. Bagi media

periklanan jejaring sosial dengan konsep UGC memiliki potensi besar menyediakan market yang lebih terarah dan terpusat bagi mereka[10].

II.6. Web *Service*s

W3C mendefinisikan web *service* sebagai sebuah software aplikasi yang dapat teridentifikasi oleh URI dan memiliki *interface* yang didefinisikan, dideskripsikan, dan dimengerti oleh XML atau JSON dan juga mendukung interaksi langsung dengan software aplikasi yang lain dengan menggunakan message berbasis XML atau JSON melalui protokol internet Web *service* adalah sebuah software aplikasi yang tidak terpengaruh oleh platform, menyediakan method-method yang dapat diakses oleh network. Web *Service* juga akan menggunakan XML untuk pertukaran data, khususnya pada dua entities bisnis yang berbeda. Beberapa karakteristik dari web *service* adalah:

1. *Message-based*
2. *Standards-based*
3. *Programming language independent*
4. *Platform-neutral*

Beberapa key standard di dalam web *service* adalah: JSON, XML, SOAP, WSDL and UDDI[11].

II.7. API (*Application Programming Interface*)

Application programming interface (API) merupakan suatu dokumentasi yang terdiri dari interface, fungsi, kelas, struktur dan sebagainya untuk membangun sebuah perangkat lunak. Dengan adanya API ini, maka memudahkan programmer untuk "membongkar" suatu software, kemudian dapat dikembangkan atau diintegrasikan dengan perangkat lunak yang lain. API dapat dikatakan sebagai penghubung suatu aplikasi dengan aplikasi lainnya yang memungkinkan programmer menggunakan sistem function. Proses ini dikelola

melalui sistem operasi. Keunggulan dari API ini adalah memungkinkan suatu aplikasi dengan aplikasi lainnya dapat saling berhubungan dan berinteraksi[12].

II.8. GPS (*Global Positioning System*)

GPS adalah singkatan dari *Global Positioning System*, yang merupakan sistem navigasi dengan menggunakan teknologi satelit yang dapat menerima sinyal dari satelit. Sistem ini menggunakan 24 satelit yang mengirimkan sinyal gelombang mikro ke bumi. Sinyal ini diterima oleh alat penerima (*receiver*) di permukaan, dimana GPS *receiver* ini akan mengumpulkan informasi dari satelit GPS, seperti:

1. Waktu. GPS *receiver* menerima informasi waktu dari jam atom yang mempunyai keakurasian sangat tinggi.
2. Lokasi. GPS memberikan informasi lokasi dalam tiga dimensi:
   a. Latitude
   b. Longitude
   c. Elevasi
3. Kecepatan. Ketika berpindah tempat, GPS dapat menunjukkan informasi kecepatan berpindah tersebut.
4. Arah perjalanan. GPS dapat menunjukkan arah tujuan.
5. Simpan lokasi. Tempat-tempat yang sudah pernah atau ingin dikunjungi bisa disimpan oleh GPS *receiver*.
6. Komulasi data. GPS *receiver* dapat menyimpan informasi track, seperti total perjalanan yang sudah pernah dilakukan, kecepatan rata-rata, kecepatan paling tinggi, kecepatan paling rendah, waktu/jam sampai tujuan, dan sebagainya[13].

II.9. JSON

JSON (JavaScript *Object* Notation) adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (generate) oleh komputer. Format ini dibuat berdasarkan bagian dari Bahasa

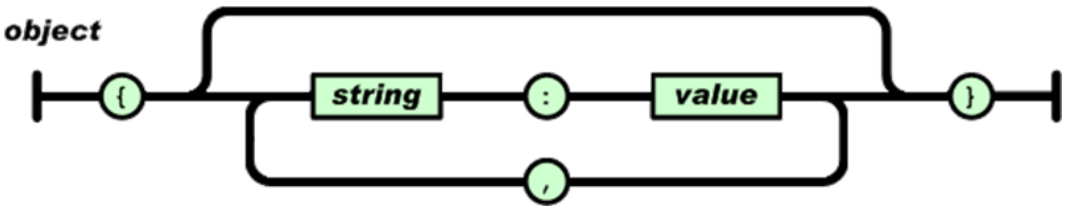
Pemprograman JavaScript, Standar ECMA-262 Edisi ke-3 - Desember 1999. JSON merupakan format teks yang tidak bergantung pada bahasa pemrograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh programmer keluarga C termasuk C, C++, C#, Java, JavaScript, Perl, Python dll. Oleh karena sifat-sifat tersebut, menjadikan JSON ideal sebagai bahasa pertukaran-data. JSON terbuat dari dua struktur:

1. 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[14].
2. Daftar nilai terurutkan (an *ordered list* of *values*). Pada kebanyakan bahasa, hal ini dinyatakan sebagai larik (array), vektor (vector), daftar (*list*), atau urutan (sequence)[14].

Struktur-struktur data ini disebut sebagai struktur data universal. Pada dasarnya, semua bahasa pemprograman modern mendukung struktur data ini dalam bentuk yang sama maupun berlainan. Hal ini pantas disebut demikian karena format data mudah dipertukarkan dengan bahasa-bahasa pemprograman yang juga berdasarkan pada struktur data ini. JSON menggunakan bentuk sebagai berikut:

1. Objek

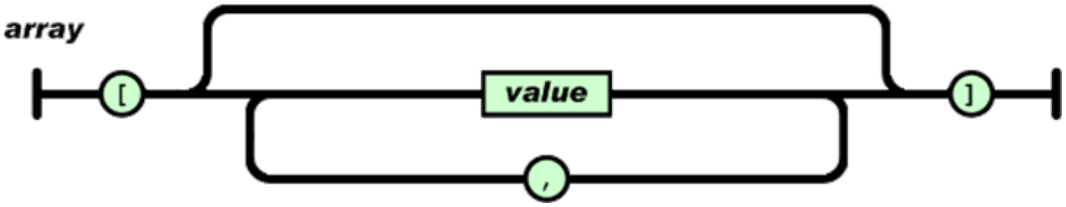
   Objek adalah sepasang nama / nilai yang tidak terurutkan. Objek dimulai dengan { (kurung kurawal buka) dan diakhiri dengan } (kurung kurawal tutup). Setiap nama diikuti dengan : (titik dua) dan setiap pasangan nama/nilai dipisahkan oleh , (koma). Objek biasanya digunakan untuk menyimpan data tunggal dalam bentuk JSON[14].



Gambar II-4 Objek JSON

2. Larik

Larik adalah kumpulan nilai yang terurutkan. Larik dimulai dengan [(kurung kotak buka) dan diakhiri dengan] (kurung kotak tutup). Setiap nilai dipisahkan oleh , (koma). Larik dalam JSON dapat digunakan sebagai value dari JSON *object* hal ini dapat berguna jika JSON menyimpan data bertingkat[14].



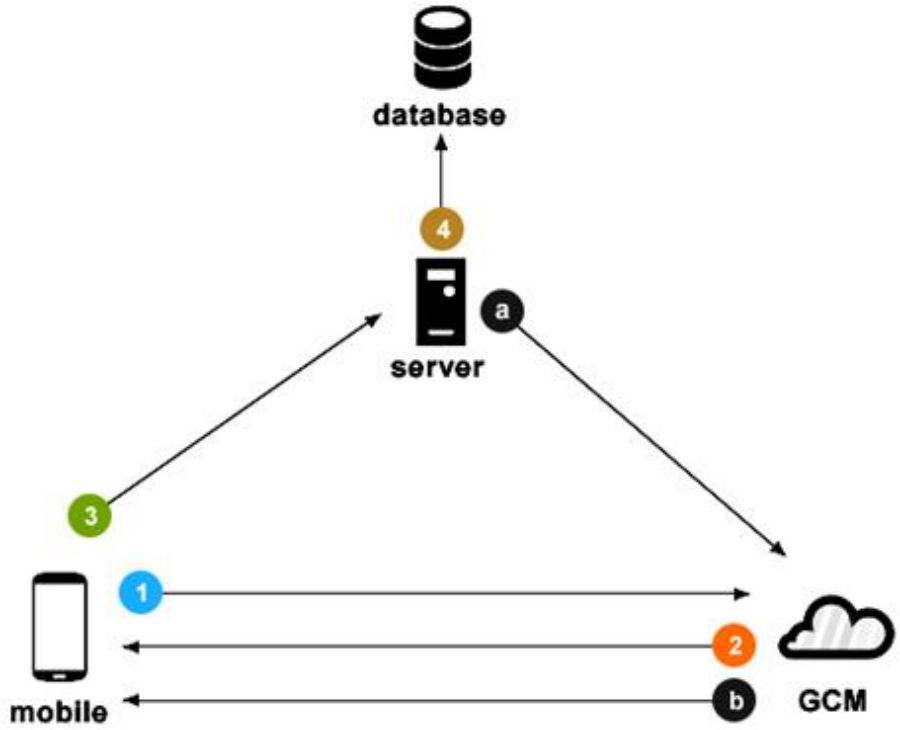
Gambar II-5 Array JSON

Bentuk data JSON objek dan larik dapat saling dikombinasikan untuk mendukung struktur data yang lebih kompleks. JSON mendukung beberapa tipe data untuk menjadi value seperti Angka, String, Bollean dan nilai NULL[14].

II.10. Google *Cloud Messaging*

Google *Cloud Messaging* adalah teknologi untuk membantu pengembang mengirim data dari server menuju device android secara langsung melalui server google, pemanfaatan teknologi ini biasanya digunakan untuk push *messaging*. Pengiriman pesan menggunakan teknologi ini mengharuskan adanya API key, key

ini terkait langsung dengan akun google yang digunakan untuk mempublikasikan aplikasi melalui google play[8].



Gambar II-6 GCM Model[7]

Berdasarkan ilustrasi dari Gambar II 6 GCM Model terdapat dua server yang terlibat dalam proses push *messaging* , pada setiap server yang terlibat terdapat proses pengiriman data yang berbeda-beda diantaranya[8]:

1. Device android mengirimkan sender id dan *application* id menuju GCM Server untuk registrasi.
2. Setelah berhasil registrasi, GCM server akan mengirimkan registration id menuju device android.
3. Setelah device android mendapatkan registration id, device tersebut akan mengirimkan registration id menuju server aplikasi.
4. Server aplikasi akan menyimpan registrasi id yang dikirimkan untuk digunakan sebagai id saat menggunakan push messaging.

II.11. *Struct*ured *Analysis* and *Design*

Structured analysis and design adalah pendekatan sistem analisis dan desain sistem informasi secara hierarki. SADM diproduksi untuk CCTA, kantor pemerintah Inggris yang berkaitan dengan penggunaan teknologi di pemerintahan dari tahun 1980 dan seterusnya[15].

II.11.1. Diagram Konteks

Diagram Context, adalah data flow diagram tingkat atas (DFD Top Level), yaitu diagram yang paling tidak detail, dari sebuah sistem informasi yang menggambarkan aliran-aliran data ke dalam dan ke luar sistem dan ke dalam dan ke luar entitas-entitas eksternal. (CD menggambarkan sistem dalam satu lingkaran dan hubungan dengan entitas luar. Lingkaran tersebut menggambarkan keseluruhan proses dalam sistem) [16].

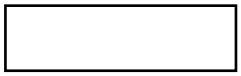
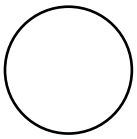
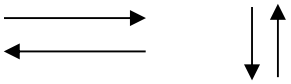
II.11.2. Data Flow Diagram

Data Flow Diagram (DFD) adalah representasi grafik dari sebuah sistem. DFD menggambarkan komponen-komponen sebuah sistem, aliran-aliran data di mana komponen-komponen tersebut, dan asal, tujuan, dan penyimpanan dari data tersebut. DFD terdiri dari empat komponen utama diantaranya adalah

1. Entitas
2. Proses
3. Data Store
4. Data Flow

DFD dapat dikatakan merupakan breakdown dari diagram konteks. Analisis mengenai sistem yang lebih detail dapat dilakukan dengan melakukan breakdown pada DFD level 1, 2 dan seterusnya. Berikut adalah Tabel II.2 Simbol Diagram Konteks dan DFD[16].

Tabel II.2 Simbol Diagram Konteks dan DFD[15]

Simbol	Nama Simbol	Keterangan
	Entitas Luar	Menggambarkan entitas eksternal yang berhubungan dengan sistem
	Sistem (Konteks) / Proses (DFD)	Menggambarkan proses yang ada dalam suatu sistem
	Aliran Data / Informasi	Menggambarkan aliran data antar proses, data store dan entitas luar
	Data *Storage*	Mengambarkan tempat penyimpanan data di dalam sistem

II.11.3. Spesifikasi Proses

Spesifikasi Proses adalah tabel yang berisi keterangan atau deskripsi dari semua proses yang terdapat pada DFD. Logika proses yang ditulis pada spesifikasi proses harus dituliskan secara jelas baik menggunakan bahasa deskriptif atau pseudecode tetapi tidak boleh mengkombinasikan kedua bahasa tersebut[16].

Berikut adalah Tabel II.3 Contoh Spesifikasi Proses:

Tabel II.3 Contoh Spesifikasi Proses[10]

No	Proses	Keterangan
1.	No Proses	Menyatakn no proses
	Nama Proses	Menyatakan nama proses

No	Proses	Keterangan
	Source (Sumber)	(Menyatakan sumber data input menuju proses)
	Input	(menyatakan isi data yang masuk ke proses)
	Output	(menyatakan informasi yang keluar dari proses)
	Destination	(menatakan tujuan informasi output dari proses)
	Logika Proses	(menyatakan algoritma dari proses)

II.11.4. Kamus Data

Kamus data adalah tabel yang berisi deskripsi dari data yang mengalir pada DFD, Penjelasan struktur data dapat berupa field tiap data harus sama dengan yang sudah dimodelkan di ERD[16].

Berikut adalah Tabel II.4 Contoh Kamus Data:

Tabel II.4 Contoh Kamus Data

Nama aliran data	Data Siswa Nama dari data yang digunakan
Where used/How used	Daftar pada poin proses apa data digunakan
Keterangan	Uraian singkat dari data yang digunakan
Struktur Data	Daftar komponen data yang ada pada data yang digunakan
Deskripsi	Jenis data dalam representasi komputer untuk masing-masing data.

II.12. *Object* Oriented *Analysis* Desain

Konsep OOAD mencakup analisis dan desain sebuah sistem dengan pendekatan objek, yaiut 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[17].

II.12.1. Unified Modeling Language (UML)

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. UML merupakan suatu bahasa pemodelan yang terdiri banyak model diantaranya adalah[17]:

1. Use Case Diagram

   Use case diagram merupakan suatu pemodelan yang menitik beratkan pada fungsionalitas utama dalam suatu konteks sistem. Use case diagram juga menekankan tingkah laku fungsional utama dalam sistem berinteraksi dengan objek diluar sistem tersebut. Selain itu, use case diagram juga telah menitik beratkan jenis hubungan diantara fungsi utama. Adapun komponen-komponen dalam use case diagram antaranya[17]:

   a. Aktor

      Aktor merupakan suatu entitas yang berkaitan dengan sistem tapi bukan dari bagian dalam sistem itu sendiri. Aktor berada diluar sistem namun berkaitan erat dengan fungsionalitas didalamnya. Aktor dapat memiliki hubungan secara langsung terhadap fungsi utama baik terhadap salah satu atau semua fungsionalitas utama. Aktor juga dapat dibagi terhadap berbagai jenis atau tingkatan dengan cara digeneralisasi atau dispesifikasi tergantung kebutuhan sistemnya. Aktor biasanya

dapat berupa pengguna atau database yang secara pandang berada dalam suatu ruang lingkup sistem tersebut[17].

b. Use Case

Use case merupakan gambaran umum dari fungsi atau proses utama yang menggambarkan tentang salah satu perilaku sistem. Perilaku sistem ini terdefinisi dari proses bisnis sistem yang akan dimodelkan. Tidak semua proses bisnis digambarkan secara fungsional pada use case, tetapi yang digambarkan hanya fungsionalitas utama yang berkaitan dengan sistem. Use case menitik beratkan bagaimana suatu sistem dapat berinteraksi baik antar sistem maupun diluar sistem[17].

2. *Class* Diagram

Class diagram merupakan diagram yang selalu ada di permodelan sistem berorientasi objek. *Class* diagram menunjukkan hubungan antar *class* dalam sistem yang sedang dibangun dan bagaimana mereka saling berkolaborasi untuk mencapai suatu tujuan. Kelas pada kelas diagram terdiri dari 3 bagian utama yaitu nama kelas, isi properti dari kelas beserta metode yang ada pada kelas tersebut. Kelas juga memiliki jenis-jenis hubungan seperti asosiatif, dependensi, agregasi, komposisi, spesifikasi dan generalisasi. Hubungan ini digunakan untuk menggambarkan bagaimana hubungan dan interaksi yang terjadi antar kelas. Masing-masing komponen penyusun kelas memiliki hak akses seperti public, private, dan protected[17].

3. Sequence Diagram

Sequence diagram menjelaskan secara detil urutan proses yang dilakukan dalam sistem untuk mencapai tujuan dari use case: interaksi yang terjadi antar *class*, operasi apa saja yang terlibat, urutan antar operasi, dan informasi yang diperlukan oleh masing-masing operasi[17].

II.13. *Object Relational* Mapping

Object Relational Mapping (ORM) adalah salah satu teknik untuk memetakan basis data relasional ke model objek. Pemetaan dapat dikategorikan menjadi 4, yaitu[17]:

1. *Basic mapping*
2. *Relationship mapping*
3. *Composition mapping*
4. *Inheritance mapping*