

BAB II

DASAR TEORI

2.1 TEKNOLOGI PADA APLIKASI

2.1.1 *AUGMENTED REALITY*^[4]

Augmented Reality adalah merupakan sebuah teknologi yang dapat menampilkan suatu objek baik 2D maupun 3D dalam dunia maya menuju dunia yang nyata secara *real-time*.

Augmented Reality memiliki kemampuan untuk memberikan info dari dunia *virtual* atau maya dan ditampilkan kedalam bentuk nyata di dunia nyata dengan menggunakan perlengkapan seperti *webcam*, HP dan bahkan sekarang sudah merambah pada kaca mata.

Benda-benda maya menampilkan informasi yang tidak dapat diterima oleh pengguna dengan inderanya sendiri. Hal ini membuat realitas bertambah sesuai sebagai alat untuk membantu pemahaman dan interaksi penggunaanya dengan dunia nyata. Informasi yang ditampilkan oleh benda maya membantu pengguna melaksanakan kegiatan-kegiatan dalam dunia nyata.

Dalam penggunaanya teknologi *Augmented Reality* ini sudah diterapkan pada berbagai bidang seperti dalam bidang kesehatan, bidang kemiliteran, bidang hiburan, maupun diterapkan pada kegiatan seperti penentuan lokasi pada telepon genggam.



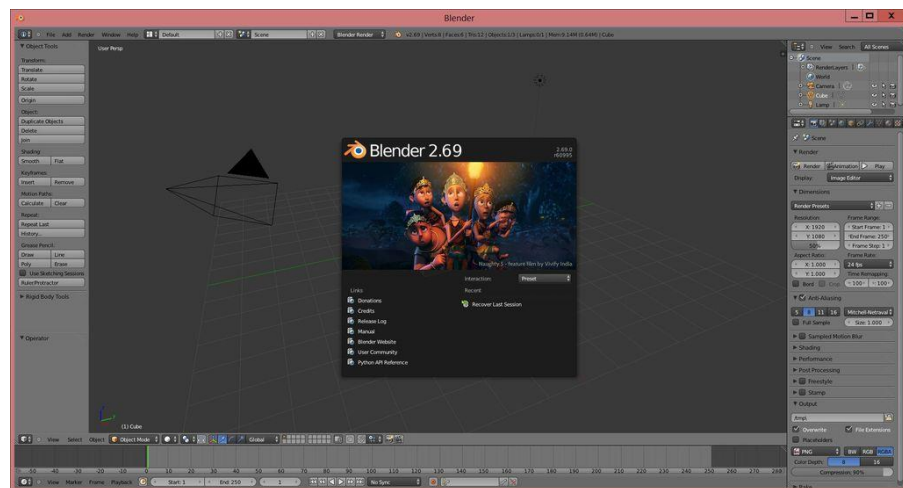
Gambar 2.1 *Augmented Reality*^[5]

2.2 PERANGKAT PENYUSUN *SOFTWARE*

2.2.1 3D *Blender*^[5]

3D *Blender* adalah salah satu aplikasi yang berbasis *open source* yang berfungsi sebagai aplikasi dalam pembuatan model bentuk 3D, animasi, proses *texturing model*, *rigging*, simulasi, *rendering*, *motion tracking*, *video editing* bahkan pembuatan *game*.

Aplikasi *software Blender* 3D ini memiliki kelebihan selain merupakan salah satu aplikasi *open source* namun juga merupakan aplikasi yang *multiplatform* yang memungkinkan dapat digunakan di berbagai macam sistem operasi manapun.



Gambar 2.2 *Blender* 2.69^[5]

2.2.2 *Vuforia* SDK^[6]

Vuforia SDK merupakan sebuah *software development kit* buatan perusahaan Qualcomm yang berfungsi sebagai perangkat tambahan yang dapat mendeteksi sebuah gambar (*marker*) dan memungkinkan untuk menciptakan *Augmented Reality* pada perangkat *mobile*, pada pengaplikasiannya sering diterapkan pada *software* Unity.

Gambar 2.3 Vuforia logo^[6]

2.2.3 Unity^[7]

Unity merupakan *software* yang dipergunakan sebagai *game engine* atau merupakan *software* untuk pembuatan sebuah *game* dan juga dapat difungsikan sebagai pemodelan visualisasi interaktif lainnya. Dalam perkembangannya *software* ini juga merambah pada teknologi *Augmented Reality* dengan menambahkan Vuforia yaitu merupakan suatu ekstensi yang mendukung pada aplikasi Unity yang memungkinkan dapat membuat sebuah teknologi *Augmented Reality*.

Pada ekstensi Vuforia di *software* Unity ini juga memungkinkan untuk membuat sebuah teknologi *Augmented Reality* dalam kemasan *mobile*.

Gambar 2.4 *Augmented Reality* pada Unity^[7]

2.3 PERANGKAT *HARDWARE*

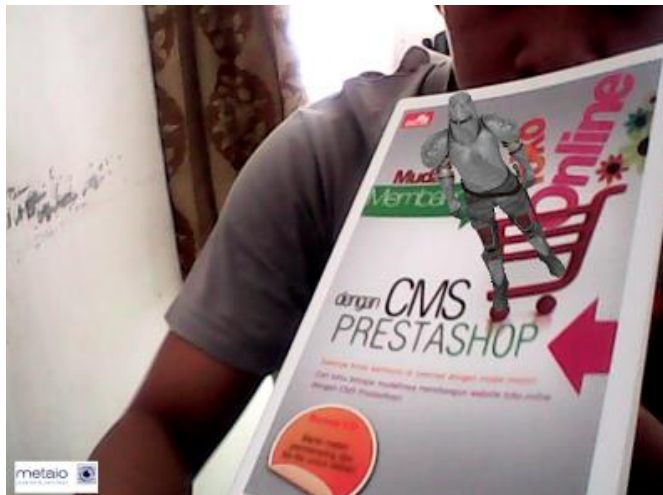
2.3.1 *Marker*^[8]

Marker merupakan sebuah perangkat tambahan yang nantinya digunakan sebagai objek acuan agar sebuah objek 3D dapat muncul pada

teknologi *Augmented Reality*, pada perkembangannya terdapat dua jenis *marker* yang perlu diketahui yaitu yang pertama adalah sebagai berikut :

a. *Markerless Augmented Reality*

Markerless Augmented Reality merupakan salah satu metode *Augmented Reality* tanpa menggunakan *frame marker* sebagai objek yang dideteksi. Dengan adanya *Markerless Augmented Reality*, maka penggunaan *marker* sebagai *tracking* objek yang selama ini menghabiskan ruang akan digantikan dengan gambar, atau permukaan apapun yang berisi dengan tulisan, logo, atau gambar sebagai *tracking object* (objek yang dilacak) agar dapat langsung melibatkan objek yang dilacak tersebut sehingga dapat terlihat hidup dan interaktif, juga tidak lagi mengurangi efisiensi ruang dengan adanya *marker*.



Gambar 2.5 Bentuk *markerless*^[8]

b. *Marker Augmented Reality*

Ini adalah salah satu jenis metode *marker* pada *Augmented Reality* dimana biasanya menggunakan gambar persegi dengan kombinasi warna putih dan hitam yang bervariasi, dengan begitu komputer akan mengetahui letak dan orientasi *marker* terhadap sumbu x, y, maupun z.

Gambar 2.6 Bentuk *marker basic*^[8]

2.4 SISTEM OPERASI ^[9]

2.4.1 Android

Pada Tugas Akhir ini, *handphone* dengan sistem operasi Android digunakan sebagai perangkat keras yang nantinya berfungsi untuk menjalankan aplikasi dan juga sebagai alat untuk mendeteksi *marker* melalui kamera. Android adalah sebuah sistem operasi bergerak (*mobile operating system*) yang mengadopsi sistem operasi LINUX, namun telah dimodifikasi. Android diambil alih oleh Google pada tahun 2005. Google menginginkan agar Android bersifat terbuka dan gratis, oleh karena itu hampir setiap kode program Android diluncurkan berdasarkan lisensi *open-source*.^[9]

Android telah mengalami *update* versi pengembangan sejak diluncurkan pertama kali. Berikut versi pengembangan OS android dapat dilihat pada tabel 2.1^[9]

Tabel 2.1 *Update OS (Operating Sistem) Android*^[9]

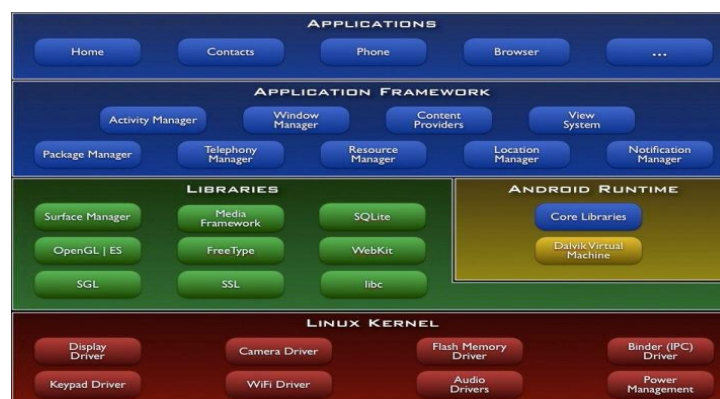
Versi Android	Diluncurkan	Nama Versi Android
Beta	5 November 2007	
1.0	23 September 2008	
1.1	9 Februari 2009	
1.5	30 April 2009	Cupcake
1.6	15 September 2009	Donut
2.0/2.1	26 Oktober 2009	Eclair
2.2	20 Mei 2010	Froyo
2.3	6 Desember 2010	Gingerbread
3.0	22 Februari 2011	Honeycomb

4.0.1	19 Oktober 2011	Ice Cream Sandwich
4.1	Sekitar Pertengahan 2012	Jelly Bean

Android tersedia secara *Open Source* bagi manufaktur perangkat keras untuk memodifikasi kebutuhan. Meskipun konfigurasi perangkat Android tidak sama antara satu perangkat dengan perangkat lainnya, namun Android sendiri mendukung fitur-fitur berikut :

- Penyimpanan (*Storage*) – menggunakan SQLite yang merupakan *databaserelational* yang ringan untuk menyimpan data.
- Koneksi (*Concetivity*) – mendukung GSM/EDGE, CDMA, EV-DO, UMTS, *Bluetooth*, *WiFi*, LTE, dan *WiMAX*.
- Pesan (*Messaging*) – mendukung SMS dan MMS.
- Web Browser
- Media – media yang didukung antara lain (MPEG-4, MP3, MIDI, dan media lainnya).
- *Hardware* – terdapat *Accelerometer*, Sensor, Kamera, *Digital Compass*, *Proximity Sensor* dan GPS.
- *Multi-touch* – mednukung *screen multi-touch*.
- *Multi-tasking* – mendukung aplikasi *multi-tasking*
- Dukung Flash – Android 2.4 medukung Flash 10.1.^[9]

Sistem operasi Android mempunyai lapisan - lapisan yang berisi dari beberapa program yang mendukung jalannya fungsi dari sistem operasi. Lapisan-lapisan tersebut dapat dilihat pada gambar 2.8.^[9]



Gambar 2.7 Arsitektur Android^[9]

1. *Applications dan Widgets*

Applications dan Widgets adalah *layer* di mana berhubungan dengan aplikasi saja yang mana sering diakses oleh pengguna, pada *layer* ini pengguna biasanya mengunduh aplikasi kemudian melakukan instalasi dan jalankan aplikasi tersebut. Di *layer* ini pengguna menemukan aplikasi inti termasuk klien email, program SMS, kalender, peta, *browser*, kontak.

2. *Applications Frameworks*

Android adalah "*Open Development Platform*" yaitu Android memberikan kesempatan kepada pengembang untuk membangun aplikasi yang bagus, inovatif serta lebih kompleks. *Application framework* merupakan sekumpulan peralatan dasar seperti alokasi *resource smartphone*, aplikasi telepon, pergantian antar-proses atau program, serta pelacakan lokasi fisik telepon. Pengembang memiliki akses penuh menuju *Android Protocol Interface (API) framework* seperti yang dilakukan oleh aplikasi yang kategori inti. Arsitektur aplikasi dirancang agar pengguna dengan mudah dapat menggunakan kembali komponen yang sudah digunakan (*reuse*). Dengan demikian, aplikasi *framework* merupakan *layer* di mana bagi para pembuat aplikasi dapat mengembangkan/membuat aplikasi yang akan dijalankan di sistem operasi Android. Hal ini dikarenakan pada *layer* ini aplikasi dapat dirancang dan dibuat.

3. *Libraries*

Libraries adalah *layer* di mana fitur - fitur Android berada, biasanya para pembuat aplikasi mengakses *libraries* untuk menjalankan aplikasinya. Pada *layer* ini berisi *library-library* dalam bahasa C++ yang digunakan oleh komponen- komponen yang berada pada Android. Beberapa *core library* seperti *libraries media* untuk pemutaran media audio dan video, *libraries surface manager* untuk mengatur tampilan, *libraries Graphics* mencakup SGL untuk grafis 2D dan OpenGL untuk grafis 3D, *libraries SQLite* untuk dukungan *database*, *libraries SSL* dan *Webkit* terintegrasi dengan *web browser* dan *security*, *libraries Live*

Webcore mendukung modern *web browser* maupun *engine embedded web view*.

4. *Android Run Time*

Pada *layer* ini yang menjadikan aplikasi Android dapat berjalan, dimana dalam prosesnya menggunakan implementasi Linux. Di dalam *Android Run Time* dibagi menjadi dua bagian yaitu:

Aplikasi Android yang dibangun menggunakan bahasa *Java*, sementara *Dalvik* sebagai *virtual* mesinnya bukan *Virtual Machine Java*. Dengan demikian, *core libraries* merupakan *library* yang berfungsi menterjemahkan bahasa *Java*.

- a) *Core Libraries* merupakan virtual mesin berbasis *register* yang dioptimalkan untuk menjalankan fungsi-fungsi secara efisien.
- b) *Dalvik Virtual Machine* adalah mesin yang membuat kerangka dasar pada aplikasi Android.

5. *Linux Kernel*

Linux kernel merupakan *layer* yang berisi inti dari sistem operasi Android tersebut berada. Pada *layer* ini terdapat *file -file system* yang mengatur sistem *processing*, *memory*, *resources*, *drivers*, dan sistem-sistem operasi Android lainnya. Google menggunakan *linux kernel* versi 2.6 untuk membangun sistem Android.^[9]

Android menggunakan bahasa pemrograman *Java*. Kode *Java* dan *file resource* yang dibutuhkan oleh suatu aplikasi dikompilasi secara bersama yang kemudian dijadikan dalam satu paket oleh *tool* ke dalam paket Android yang kemudian menghasilkan *file* berformat.[dot]apk. Berikut enam jenis komponen pada aplikasi Android yaitu:

1. *Activities*

Suatu *activity* menampilkan *user interface* (UI) kepada pengguna. Satu *activity* bertindak sebagai *user interface* dan menunjukkan aplikasi satu layar tersebut kepada pengguna. Apabila pindah dari satu *activity* ke *activity* yang lain dapat dilakukan dengan satu langkah, misalnya dengan mengklik tombol, memilih opsi atau menggunakan *triggers* tertentu.

2. *Service*

Service tidak memiliki *Graphic User Interface* (GUI), melainkan berjalan secara *background*. *Service* berjalan pada *thread* utama dari proses aplikasi. Sehingga, pengguna dapat melakukan aktifitas ganda contohnya pada saat pengguna sedang memutar musik pengguna juga dapat mengetik pesan atau menggunakan aplikasi lain.

3. *Broadcast Receiver*

Broadcast receiver digunakan untuk menerima dan menyampaikan pemberitahuan atau informasi. Contoh *broadcast* seperti notifikasi zona waktu berubah, ketika baterai sudah dalam keadaan akan habis, ketika gambar sudah selesai diambil oleh *kamera*, atau yang lainnya. *Broadcast receiver* tidak mempunyai *user interface* (UI), namun memiliki *activity* yang dapat memberikan respon informasi yang diterima, atau dapat menggunakan *Notification Manager* untuk memberitahu kepada pengguna, seperti lampu latar menyala atau terdapat getaran.

4. *Content Provider*

Content provider membuat kumpulan aplikasi data secara spesifik sehingga bisa digunakan oleh aplikasi lain. Data disimpan dalam *file* sistem seperti *database SQLite*. *Content provider* menyediakan cara untuk mengakses data yang dibutuhkan oleh suatu *activity*, misalnya ketika menggunakan aplikasi yang membutuhkan peta atau aplikasi yang membutuhkan untuk mengakses data kontak dan navigasi, maka disinilah fungsi *content provider*.^[9]

2.5 KONTEN

2.5.1 *DRIVE TEST*^[11]

Drive Test adalah merupakan suatu aktifitas maupun kegiatan bertujuan untuk mengumpulkan data suatu pengukuran kualitas sinyal pada jaringan yang nantinya berguna untuk meningkatkan kualitas suatu jaringan dan pada perkembangan kapasitas jaringan.

Dalam kegiatannya *Drive Test* dapat dilakukan baik *indoor* maupun *outdoor*, pada kegiatan *Drive Test outdoor* maka diperlukan mobil sebagai alat

transportasinya dengan kecepatan rendah agar sinyal yang dapat di peroleh lebih banyak dan celahnya akan semakin sedikit, ini berguna agar mempermudah *engineer* menganalisa daerah tersebut, adapun *Drive Test* yang dilakukan pada lokasi *indoor* atau sering disebut dengan *walk test* yang biasanya dikerjakan pada sebuah gedung atau area dekat dengan BTS.

Pada kegiatan *Drive Test* ini perlu adanya beberapa *tools* yang nantinya digunakan dalam kegiatan *Drive Test* ini yaitu diantaranya :

1. Laptop atau PC yang sudah terinstal *software* TEMS yang digunakan untuk *monitoring* jaringan saat melakukan kegiatan *Drive Test*.
2. *Mobile station* (MS) yang sudah terdapat *software* TEMS didalamnya guna sebagai alat untuk menerima sinyal dari BTS yang akan dicek.
3. GPS (*Global Positioning System*) merupakan alat yang dipergunakan untuk mengambil data *longitude* dan *latitude* yang nantinya hasilnya dapat berupa *measurement* TEMS yang dipetakan pada aplikasi seperti Map info maupun Google Map.



Gambar 2.8 Konsep dasar *Drive Test*^[11]

kualitas jaringan GSM, memiliki parameter-parameter diantaranya adalah sebagai berikut :^[11]

1. RxLevel merupakan kuat *level* sinyal yang diterima oleh *Mobile Station* (MS) dalam satuan dBm. Semakin kecil nilai dBm maka semakin lemah daya yang diterima.

Tabel 2.2 Nilai RxLevel^[11]

Nilai (dBm)	Keterangan
-75 to 0	Baik
-95 to -75	Buruk
-120 to -95	Sangat Buruk

2. RxQual merupakan kualitas sinyal yang diterima *Mobile Station* (MS). Semakin besar nilai RxQual maka semakin buruk kualitas sinyalnya.

Tabel 2.3 Nilai RxQual^[11]

Nilai	Keterangan
0 to 5	Baik
5 to 6	Buruk
6 to 8	Sangat Buruk

3. *Speech Quality Indicator* (SQI) merupakan kualitas suara yang diterima oleh *Mobile Station* (MS). Semakin besar nilai SQI semakin baik.

Tabel 2.4 Nilai SQI GSM^[11]

Nilai	Keterangan
>30	Baik
<30	Buruk

4. *Carrier to Interference* (C/I) merupakan rasio antara kuat sinyal bit-bit informasi dan kuat sinyal bit-bit *interference* yang tidak diinginkan. Pada GSM *specification* rasio antara C/I harus lebih besar dari 9 dB. Tetapi biasanya operator menentukan bahwa rasio C/I harus lebih besar dari 12 dB.

Tabel 2.5 Nilai C/I GSM^[11]

Nilai (dB)	Keterangan
>12	Baik
<12	Buruk

Sedangkan pada WCDMA juga memiliki parameter-parameter yang perlu diketahui, diantaranya sebagai berikut :

1. RSCP merupakan kuat *level* sinyal yang diterima oleh *Mobile Station* (MS) dalam satuan dBm. Semakin kecil nilai dBm maka semakin lemah daya yang diterima pada WCDMA.

Tabel 2.6 Nilai RSCP^[11]

Nilai (dBm)	Keterangan
-75 to 0	Baik
-95 to -75	Buruk
-120 to -95	Sangat Buruk

2. EC/NO merupakan kualitas sinyal yang diterima *Mobile Station* (MS). Semakin besar nilai EC/NO maka semakin buruk kualitas sinyalnya pada WCDMA.

Tabel 2.7 Nilai EC/NO^[16]

Nilai	Keterangan
0 to -6	Sangat Baik
-6 to -11	Baik
-11 to -16	Buruk
-16 >=	Sangat Buruk

3. *Speech Quality Indicator* (SQI) merupakan kualitas suara yang diterima oleh *Mobile Station* (MS). Semakin besar nilai SQI semakin baik.

Tabel 2.8 Nilai SQI WCDMA^[11]

Nilai	Keterangan
>30	Baik
<30	Buruk

4. *Carrier to Interference* (C/I) merupakan rasio antara kuat sinyal bit-bit informasi dan kuat sinyal bit-bit *interference* yang tidak diinginkan. Pada WCDMA *specification* rasio antara C/I harus lebih besar dari 9 dB. Tetapi biasanya operator menentukan bahwa rasio C/I harus lebih besar dari 12 dB.

Tabel 2.9 Nilai C/I WCDMA^[11]

Nilai (dB)	Keterangan
>12	Baik
<12	Buruk

2.5.2. TEMS^[12]

TEMS merupakan sebuah aplikasi yang dirancang guna untuk membantu suatu kegiatan *Drive Test*, beberapa hal yang dapat diperoleh pada *software* ini diantaranya yaitu adalah BCCH carrier ARFCN, kode Negara *mobile station*, kode jaringan, kode area *cell*, identitas *cell*, kode identitas *base station* pada *cell* yang sedang melakukan pelayanan atau *servicing* sinyal.

Selain itu juga pada aplikasi ini, TEMS memberikan beberapa informasi mengenai suatu *power level* sinyal yang diterima, kualitas sinyal yang diterima, BSIC, ARFCN, nomor *channel*, nomor *timeslot*, tipe *channel*, mode *channel*, nomor *sub-channel*, indikasi *hopping channel*, *mobile allocation index offset*, nomor *hopping sequence* pada *dedicated channel* dan masih banyak lagi sebagai data guna untuk analisa pada kegiatan *Drive Test*.

Gambar 2.9 Aplikasi TEMS *Investigation* ^[12]

Persiapan standar yang dilakukan dalam pengoperasian TEMS saat melakukan kegiatan *Drive Test* adalah sebagai berikut :^[11]

1. Menyiapkan TEMS

Dalam persiapan TEMS langkah awal yang dilakukan ialah membuat *workspace* baru, didalam *workspace* yang baru itu terdapat

beberapa tampilan jendela yang diperlukan seperti *Radio Parameter*, *Serving + Neighbor GSM/WCDMA*, *Map* dan sebagainya.

Lalu file yang disimpan akan memiliki ekstensi (*.cell), presentasi MAP pada TEMS mendukung Map Info sehingga map seperti jalan, gedung, kontur dan sebagainya dapat dibuka dalam jendela TEMS, Map tersebut berasal dari file *GeoSet* yang dimiliki

2. Menghubungkan *External Equipment*

Sebelum Menghubungkan *External Equipment* pastikan *driver* sudah terinstal, hubungkan MS maupun GPS dengan USB Laptop, pastikan perangkat yang sudah terhubung dapat terhubung dengan melakukan *Enabling* atau *Adding* yang nantinya berguna ketika akan melakukan definisi pada *port* mana yang digunakan pada *Equipment*, dan terakhir tekan tombol *Connect All* atau *Connect* sehingga semua perangkat dapat terdaftar otomatis atau terkoneksi semua.

3. Penanganan *Log file*

Hal yang diperhatikan pertama kali itu adalah melakukan persiapan *Command Squence*, *Command Squence* adalah sebuah setingan *otomatis* dimana TEMS dapat melakukan panggilan MOC (*Mobile Originating Call*) secara *otomatis* dan waktunya bias disesuaikan. Pada *Command Squence* kita dapat membuat *command* sendiri untuk kebutuhan saat *Drive Test*, bila sudah melakukan penambahan *command* kita dapat menekan tombol *record* yang berfungsi untuk menyimpan hasil *Drive Test* dalam bentuk *log file*.

2.5.3. *HANDOVER*^[13]

Handover adalah merupakan suatu situasi dimana suatu *mobile station* atau pelanggan berganti pelayanan dari sebuah sektor menuju sektor yang lain baik dalam sebuah BTS maupun berganti pada antar BTS dengan tidak adanya pemutusan hubungan yang telah berpindah pada suatu kanal maupun sebuah frekuensi secara *otomatis* yang dilakukan oleh sistem.

Fungsi dari *Handover* ini adalah sebagai penjaga suatu kualitas panggilan agar *mobile station* dan BTS dapat tetap terhubung ketika sedang melakukan pergerakan atau berpindah layanannya.

Proses ini terjadi karena dipengaruhi dari suatu gangguan dari *eksternal* seperti adanya *interferensi* maupun batas pelayanan suatu BTS pada MS sudah melewati batas dari segi jarak, hal ini juga dipengaruhi oleh daya sinyal dan kualitas pada sinyal yang dapat dilihat pada TEMS pada parameter *RXqual* atau *RXlevel* di 2G dan *RSCP* dan *EC/NO* pada 3G.

Terdapat tiga model *Handover* yaitu adalah pada system WCDMA :^[11]

1. *Soft Handover* adalah merupakan suatu *Handover* yang terjadi antar *cell* dengan *frekuensi carrier* yang sama terjadi pada teknologi 3G.
2. *Softer Handover* adalah merupakan jenis *Handover* yang terjadi antara sektor dalam satu *cell* dengan *frekuensi* pembawa dan *node B* yang sama, *soft*, dan *softer Handover* biasanya terjadi ketika UE berada pada daerah *overlapping* dari dua *adjacent cells* terjadi pada teknologi 2G maupun 3G.
3. *Hard Handover* adalah tipe *Handover* yang terjadi pemutus hubungan dengan kanal *trafik* lama sebelum terjadi hubungan baru. tipe *Handover* ini digunakan dalam sistem seluler GSM dimana tiap *cell* menggunakan band *frekuensi* yang berbeda, terjadi pada teknologi 2G maupun 3G.
4. *Handover* antar BSC adalah tipe *Handover* dimana *Handover* ini terjadi saat MS berpindah dari satu BTS ke BTS lain yang memiliki BSC yang berbeda, terjadi pada teknologi 2G maupun 3G.
5. *Handover* antar MSC adalah tipe *Handover* dimana *Handover* ini terjadi saat MS berpindah dari satu BTS ke BTS lain yang memiliki MSC berbeda, biasanya *handover* ini dilakukan saat lintas provinsi, terjadi pada teknologi 2G maupun 3G.

2.5.4. *DROP CALL*^[14]

Drop call yaitu merupakan kondisi dimana putusnya hubungan pada saat pelanggan melakukan panggilan yang disebabkan oleh beberapa kemungkinan diantaranya seperti *timing advance* (TA) yang berlebih. TA adalah jarak waktu yang diperlukan MS untuk mencapai *base station*, kualitas suatu sinyal sinyal buruk, *power serving* sinyal rendah.

Dalam permasalahan suatu *mobile station* terkadang mengalami kendala dalam memperoleh suatu sinyal dari faktor *eksternal* diantaranya yaitu seperti adanya kontur permukaan bumi yang mengakibatkan *power* maupun kualitas sinyal menjadi buruk, adanya *interferensi* dari *eksternal*, terjadi *blocking*, *traffic* penuh, maupun terjadi *overshoot* pada BTS yang nantinya berpengaruh pada sinyal yang diterima oleh *Mobile Station*.

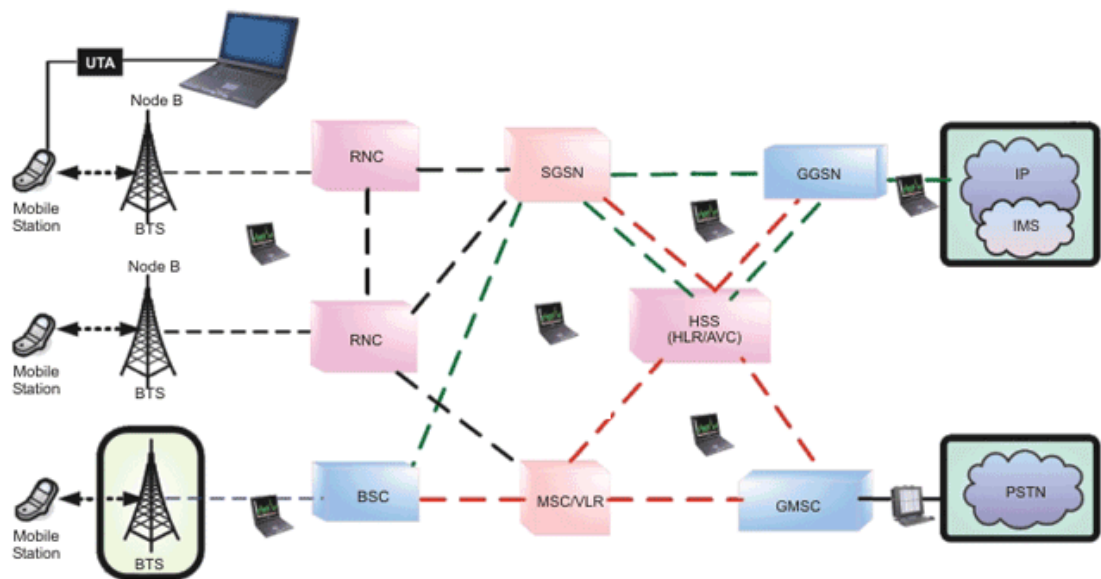
Adapun beberapa hal yang dapat menyebabkan terjadinya *Dropcall* diantaranya sebagai berikut :

1. *Obstacle* (Penghalang) : suatu penyebab terjadinya *dropcall* yang diakibatkan karena terhalangnya sinyal dari BTS ke MS karena terhalang oleh permukaan padat seperti gunung, pohon, gedung, dan sebagainya. Solusi yang dapat dilakukan ketika terjadi *Dropcall* karena *Obstacle* adalah dengan cara menambahkan *site* baru agar daerah yang mengalami *Obstacle* dapat menerima sinyal dengan baik.
2. *Handover failure* : kondisi dimana *nodeB/BTS* tidak dapat mengenal *BTS/nodeB* yang ada disebelahnya maka akan terjadi *dropcall*. Solusi yang dapat dilakukan ketika terjadi *Dropcall* karena *Handover failure* adalah melakukan proses pengenalan BTS dengan menambah *database* BTS agar dikenal oleh BTS sebelahnya, proses tersebut disebut *create neighbor*.
3. *Overshoot signal* : salah satu penyebab *dropcall* karena kondisi antena yang jauh posisi antena tidak benar / mengarah terlalu tinggi sehingga *coverage/cakupan* melebar, kondisi tersebut dapat mengganggu *user* pada *nodeB* daerah lain. Solusi yang dapat dilakukan ketika terjadi *Dropcall* karena *Overshoot signal* adalah

pengaturan antena *coverage/cakupan (re-azimuth)*, merubah posisi antena agar arah pancaran tidak mengganggu daerah cakupan BTS lain

4. Kanal penuh : salah satu penyebab *dropcall* dimana MS mendapat sinyal bagus dari segi *power* namun jelek dari segi kualitas dikarenakan BTS yang sedang mencakup MS dilokasi tersebut memiliki kanal yang sudah terisi atau penuh sehingga MS tersebut tidak mendapat jalur kanal. Solusi yang dapat dilakukan ketika terjadi *Dropcall* karena Kanal Penuh adalah dapat menambah kanal pada *site* yang sering mengalami pembengkakan penggunaan kanal agar *user* dapat mendapat jalur kanal dengan leluasa.
5. *Interferensi Co-channel* : salah satu penyebab *dropcall* dimana terdapat BTS yang berdekatan yang memiliki *frekuensi* yang sama maka akan timbul *interferensi* dan mengakibatkan *dropcall*. Solusi yang dapat dilakukan ketika terjadi *Dropcall* karena *Interferensi Co-channel* adalah agar mengurangi terjadi *interferensi* maka dilakukan perancangan penggunaan *frekuensi / plan frekuensi* agar tidak terjadi pengulangan *frekuensi* pada BTS yang dekat.
6. *Interferensi Adjacent* : salah satu penyebab *dropcall* dimana pada BTS yang berdekatan memiliki jalur *frekuensi* kanal yang berdekatan maka akan terjadi *interferensi* dan mengakibatkan *dropcall*. Solusi yang dapat dilakukan ketika terjadi *Dropcall* karena *interferensi Adjacent* adalah agar mengurangi terjadi *interferensi* maka dilakukan perancangan penggunaan *frekuensi* kanal / *plan frekuensi* agar tidak berdekatan dengan *frekuensi* kanal pada BTS yang dekat.

2.5.5 ARSITEKTUR JARINGAN 2G dan 3G^[15]



Gambar 2.10 Arsitektur jaringan 2G dan 3G

Dari gambar 2.10 terlihat arsitektur jaringan 2G dan 3G, dari alur jaringan dapat dibedakan dengan melihat arsitektur jalur pengiriman dari *mobile* satu menuju *mobile* yang lainnya, dan dapat dijelaskan sebagai berikut :

A. Arsitektur 2G

Pada arsitektur 2G secara garis lurus dapat diurutkan proses jalur pengiriman dimulai dari *mobile station* / telepon genggam yang berfungsi sebagai alat komunikasi bagi *user*, lalu diarahkan menuju ke BTS (*Base Transceiver Station*) yang memiliki fungsi sebagai alat terima dan kirim sinyal lalu, diteruskan pada BSC (*Base Station Control*) yang memiliki fungsi sebagai *monitoring* / pemantauan BTS dan mengatur jalurnya data BTS, kemudian diarahkan menuju ke MSC (*Mobile Switched Center*) yang berfungsi sebagai tempat *switching* / perpindahan menuju ke *mobile station* berlangsung, adapun MSC akan meneruskan ke BSC bila *mobile station* tujuan masih berada dalam satu MSC, namun dapat juga dipindahkan ke MSC lain jika *mobile station* tujuan adalah *mobile station* yang memiliki MSC yang berbeda atau MSC yang memiliki beda operator, dalam arsitektur 2G juga terdapat GMSC (*Gateway Mobile Switching Center*) yang berfungsi

untuk menghubungkan antara perangkat *mobile* menuju telpon rumah / menuju ke PSTN (*Public Switched Telephone Network*).

B. Arsitektur 3G

Pada arsitektur 3G secara garis lurus dapat diurutkan proses jalur pengiriman dimulai dari *mobile station* / telepon genggam yang disebut dengan UE (*User Equipment*) berfungsi sebagai alat komunikasi bagi *user*, lalu diarahkan menuju ke *node B* atau sering disebut BTS namun dipergunakan untuk menerima dan mengirimkan sinyal 3G, setelah dari *node B* akan dikirim pada RNC (*Radio Network Control*) yang berfungsi sebagai pengendali atau *monitoring* dari *node B* yang berada dibawah pengawasannya, lalu diteruskan kepada CN (*Core Network*) yang berfungsi sebagian bagian inti yang didalamnya terdapat MSC (*Mobile Switched Center*) yang berfungsi sebagai tempat *switching* / perpindahan data seperti pada layanan video, video call, dan sebagainya, dalam CN juga terdapat VLR (*Visitor Location Register*) yang berfungsi sebagai kumpulan data yang berisi informasi sementara pelanggan terutama mengenai lokasi, HLR (*Home Location Register*) berfungsi sebagai kumpulan data informasi mengenai pelanggan tetap terutama mengenai lokasi pelanggan terakhir, SGSN (*Serving GPRS Support Node*) sebagai mengantarkan packet data ke MS, *Update* pelanggan VLR, dan sebagainya, GGSN (*Gateway GPRS Support Node*) berfungsi sebagai gerbang yang menghubungkan menuju akses diluar jaringan *local* seperti internet.