



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 Quick Response Code (QR Code)

QR Code adalah kode berbentuk bar atau batang yang dapat dipindai dengan ponsel pintar menggunakan kamera dan aplikasi tertentu (Joe Murphy, 2012:13). QR Code dapat dikatakan sebagai teknologi berbasis lokasi karena QR Code menggunakan objek secara fisik di lokasi tertentu sebagai penunjuk ke tautan yang menyediakan informasi digital (Joe Murphy, 2012:55). Selain itu, Joe Murphy (2012) memaparkan, terdapat pertimbangan-pertimbangan yang diperlukan dalam penggunaan QR Code di perpustakaan yaitu

- a. Mendapatkan teknologi untuk membuat dan memindai QR Code
- b. Menentukan layanan dan program perpustakaan yang dapat didukung oleh QR Code
- c. Menentukan data yang akan digunakan pada QR Code
- d. Menugaskan karyawan untuk membuat dan mengurus QR Code

C&RL News (2010) mengungkapkan pertumbuhan perangkat *mobile* secara besar-besaran akan menghadirkan berbagai jenis layanan baru. Berbagai jenis perangkat genggam seperti ponsel pintar, *e-book readers*, dan iPads akan turut meningkatkan permintaan dan ekspektasi *user* terhadap perpustakaan. QR Code adalah salah satu yang dipercaya dapat diimplementasikan di perpustakaan untuk mengimbangi pesatnya perkembangan teknologi dunia.

QR Code pertama kali diumumkan oleh sebuah perusahaan Jepang bernama DENSO WAVE (anak perusahaan dari DENSO CORPORATION) pada 1994. Huruf QR pada QR Code merupakan kepanjangan dari *Quick Response*, yang merepresentasikan konsep pengembangan kode yang berfokus pada pembacaan berkecepatan tinggi. QR Code memiliki kontribusi dalam membuat berbagai kegiatan sehari-hari lebih efisien. QR Code tidak hanya dapat digunakan pada dokumen tercetak, tapi juga pada media elektronik, sistem informasi, dan berbagai macam bidang seperti pada pabrik, penyimpanan logistik, *retailing*, kesehatan, dan lain-lain. Oleh karena penggunaannya yang beragam, QR Code dapat menjadi alat yang sangat diperlukan dalam kehidupan sehari-hari. Selain itu, pada 2012 QR Code juga mendapat penghargaan *Good Design Award*, penghargaan yang berasal dari *Japan Institute of Design Promotion* (DENSO WAVE).

2.1.1. Jenis QR Code

Dalam pengembangannya, QR Code memiliki beberapa variasi dengan uraian sebagai berikut (DENSO WAVE).

1. QR Code model 1 dan model 2

Model 1 adalah versi asli QR Code dan memiliki kapasitas maksimal 1,167 numerik, sedangkan model 2 dibuat untuk mengembangkan model 1 dengan kapasitas maksimal 7089 numerik. Istilah



Gambar 2.1 QR Code Model 1&2 (Sumber: DENSO WAVE)

QR Code pada umumnya adalah istilah yang mengacu pada tipe ini;

2. *Micro* QR Code

QR Code ini hanya menggunakan satu buah pendeteksi pola sehingga memungkinkan ruang yang lebih minimal dalam pemakaiannya. Jenis ini memiliki kapasitas maksimal sebanyak 35 numerik;



Gambar 2.2 *Micro* QR Code
(Sumber: DENSO WAVE)

2. iQR Code

Kode yang dapat dibuat dengan modul persegi atau persegi panjang. QR Code ini dapat menampung sebanyak 40,000 numerik dengan versi 61 (422x422 modul);



Gambar 2.3 iQR Code
(Sumber: DENSO WAVE)

3. SQRC

QR Code ini memiliki fungsi yang dibatasi dalam pemindaianya. Oleh karena itu, SQRC dapat digunakan untuk menyimpan informasi yang bersifat rahasia dan internal;



Gambar 2.4 iQR Code
(Sumber: DENSO WAVE)

4. LogoQ

QR Code yang didesain dengan berbagai penyesuaian seperti logo, karakter, dan gambar.

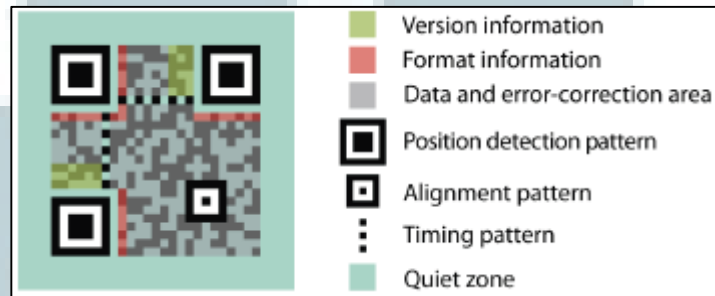
QR Code ini dibuat untuk menambah pengalaman secara visual.



Gambar 2.5 LogoQ
(Sumber: DENSO WAVE)

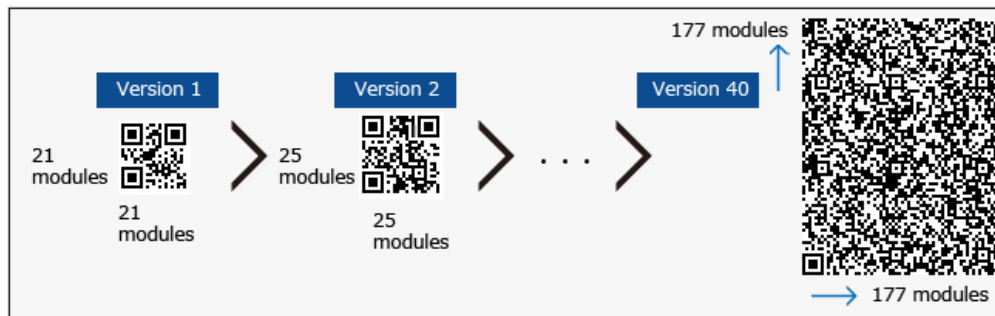
2.1.2 Struktur QR Code

Secara garis besar, QR Code terdiri dari beberapa area yang menyimpan informasi terkait struktur kode seperti versi, format, *data and error correction*, pola pendeteksi posisi, pola penyejajaran, pola *timing*, dan *quiet zone*.



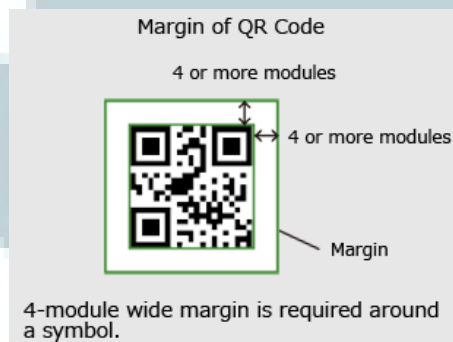
Gambar 2.6 Struktur Umum QR Code
(Sumber: DENSO ADC, 2011:4)

QR Code memiliki versi dengan rentang 1-40. Setiap versi memiliki jumlah modul yang berbeda-beda. Dalam hal ini, modul diartikan sebagai titik-titik hitam dan putih yang menyusun suatu QR Code. QR Code versi satu tersusun dari modul dengan ukuran 21x21 hingga versi empat puluh dengan ukuran modul 177x177. Setiap kenaikan satu modul, akan menambahkan empat modul baru dalam sebuah QR Code. Modul yang terdapat pada sebuah QR Code diukur hingga satuan milimeter. Setiap versi QR Code memiliki keterbatasannya masing-masing dalam menyimpan informasi. Makin banyak informasinya, makin banyak pula jumlah modul yang diperlukan sehingga mengakibatkan ukuran QR Code yang makin besar. Hal tersebut ditunjukkan pada gambar 2.7.



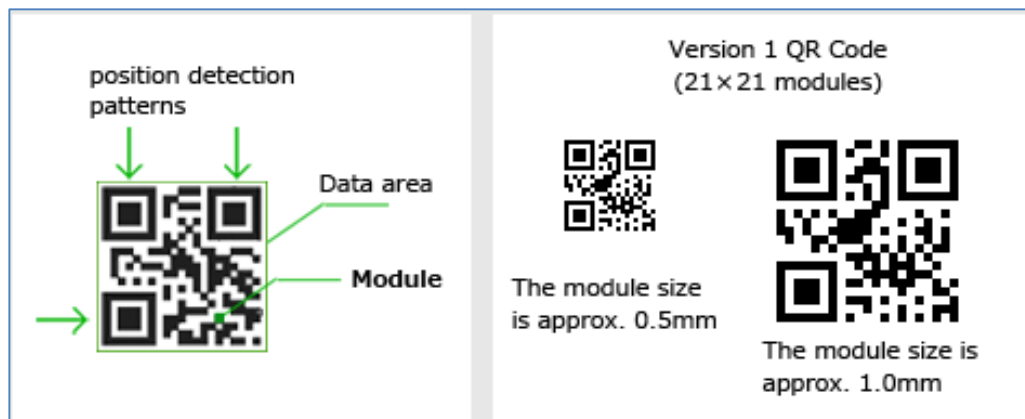
Gambar 2.7 QR Code dengan Berbagai Macam Versi
(Sumber: DENSO WAVE)

Ketika versi dan ukuran modul telah ditentukan, maka QR Code pun dapat dibuat. QR Code memerlukan sebuah batas pinggir (*quiet zone*) di sekitarnya yang menunjukkan area tanpa adanya suatu simbol apa pun yang tercetak seperti yang terlihat pada gambar 2.8.



Gambar 2.8 QR Code dengan Batas Pinggir
(Sumber: DENSO WAVE)

Makin besar modul, makin stabil dan mudah bagi alat pemindai untuk memperoleh informasi yang terdapat di dalamnya. Pembaca QR Code akan mencari pola-pola unik yang telah didesain sedemikian rupa dalam proses pendeteksian.



Gambar 2.9 Pola Unik dan Ukuran Modul QR Code
(Sumber: DENSO WAVE)

2.1.3. Karakteristik QR Code

QR Code memiliki beberapa fitur sebagai keunggulannya. Fitur-fitur tersebut dideskripsikan sebagai berikut (DENSO WAVE).

1. *High capacity encoding of data*

Jika *bar code* memiliki kemampuan untuk menyimpan data sebanyak dua puluh angka, QR Code dapat menampung sebanyak 7,089 karakter (alfabet, angka, huruf kanji, kana, hiragana, simbol, *binary*, dan kode kontrol);

2. *Small printout size*

Oleh karena QR Code dapat menyimpan informasi secara horizontal dan vertikal, QR Code dapat membuat kode pada data yang sama sebesar 1/10 dari *barcode* tradisional;

3. *Kanji and kana capability*

QR Code mampu menyimpan karakter Kana dan Kanji Jepang yang konon merupakan lokasi pengembangan awal QR Code;

4. *Dirt and damage resistant*

QR Code memiliki kemampuan untuk mengoreksi kesalahan. Kode tetap dapat dipindai walaupun terdapat kerusakan pada sebagian simbol. Akan tetapi, perbaikan yang dilakukan relatif terhadap tingkat kerusakan QR Code;

5. *Readable from any direction in 360°*

QR Code memiliki tiga simbol atau kotak utama pada tiga sudutnya sebagai pendeteksi pola. Dengan begitu, QR Code dapat dipindai dari berbagai sudut secara stabil dengan kecepatan tinggi sehingga mampu menangani gangguan dari latar belakang yang menjadi lokasi QR Code berada;

5. *Structured appending feature*

QR Code dapat dibagi menjadi beberapa area data. Begitu juga sebaliknya, informasi yang disimpan pada beberapa QR Code dapat direkonstruksi menjadi sebuah simbol baru.

Seperti yang telah diuraikan pada poin empat di atas, QR Code memiliki fitur yang memungkinkan dilakukannya perbaikan pada simbol yang rusak. Fitur tersebut dikenal juga dengan istilah *error correction feature* yang terdiri dari empat tingkat pengoreksian.

QR Code Error Correction Capability*	
Level L	Approx 7%
Level M	Approx 15%
Level Q	Approx 25%
Level H	Approx 30%

Gambar 2.10 *Error Correction Level* pada QR Code
(Sumber: DENSO WAVE)

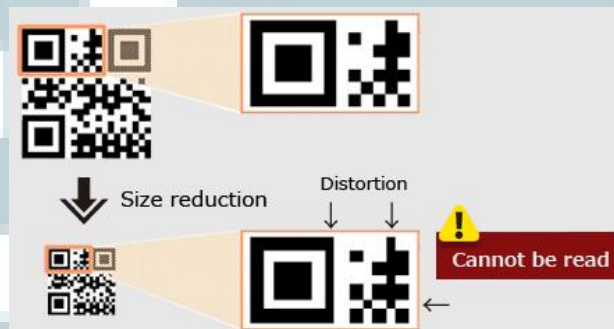
Berbagai faktor dapat dipertimbangkan ketika memilih tingkat pengoreksian QR Code. seperti lingkungan tempat QR Code akan digunakan dan ukuran QR Code itu sendiri. Penambahan tingkat pengoreksian akan memberikan kemampuan koreksi yang lebih baik. Akan tetapi, hal tersebut berdampak pada ukuran QR Code yang lebih besar. Hal tersebut berhubungan erat dengan adanya pengimplementasian metode Reed-Solomon Code pada pembuatan QR Code. Reed-Solomon Code sendiri adalah metode pengoreksian kesalahan secara matematis pada CD musik. Contoh penerapan Reed Solomon Code pada QR Code: terdapat seratus *codeword* (unit yang menyusun suatu area data) yang akan disandikan dan lima puluh *codeword* perlu dikoreksi. Reed-Solomon Code akan memerlukan dua kali lipat *codeword* (dua ratus) dari jumlah *codeword* awal yang ingin dibuat, sedangkan *codeword* yang akan dikoreksi tetap berjumlah lima puluh. Dengan demikian, tingkat koreksi QR Code sebesar 25% (tingkat Q) dan ukurannya menjadi lebih besar

Selain keunggulan yang dimiliki QR Code, terdapat juga beberapa kasus yang membuat QR Code sulit dikenali oleh pemindai seperti yang diungkapkan oleh DENSO WAVE, sebuah QR Code dapat dipindai dengan baik ketika pembuatannya memenuhi standar yang telah ditetapkan. Sebuah simbol yang terlihat seperti QR Code, tapi tidak memenuhi standar tidak dapat disebut sebagai QR Code

Deskripsi kasus-kasus yang membuat QR Code sulit dibaca:

1. modul QR Code yang terdistorsi

Ketika sebuah QR Code diperbesar atau diperkecil, setiap modulnya dapat mengalami penyimpangan. QR Code yang ukuran modulnya telah berubah tetap terlihat seperti layaknya QR Code biasa. Akan tetapi, kesulitan atau ketidakmungkinan pembacaan dapat terjadi ketika pemindaian dilakukan;



Gambar 2.11 QR Code yang Mengalami Distorsi
(Sumber: DENSO WAVE)

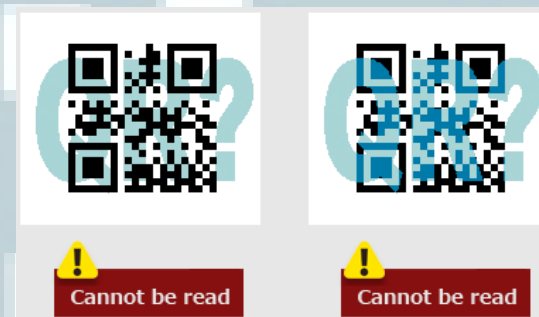
2. QR Code dengan variasi gambar dan huruf di sekitarnya

Jika terdapat huruf atau gambar yang berada di sekitar QR Code, batas pinggir yang diperlukan dalam pembacaan QR Code menjadi tidak jelas;



Gambar 2.12 QR Code dengan Berbagai Variasi Huruf dan Gambar di Sekitarnya
(Sumber: DENSO WAVE)

3. QR Code dengan variasi gambar dan huruf yang saling bertumpukan
- Ketika berbagai jenis huruf atau gambar dijadikan sebagai latar belakang pada QR Code, kejelasan antara area terang dan gelap pada QR Code akan menjadi sulit untuk dibedakan sehingga menyebabkan kesulitan dalam pembacaannya;



Gambar 2.13 QR Code dengan Berbagai Variasi Huruf dan Gambar yang Bertumpuk
(Sumber: DENSO WAVE)

4. jarak alat pemindai dengan QR Code
- Jarak yang dibuat oleh *user* dengan QR Code menjadi salah satu faktor yang memengaruhi proses pemindaian. Higginson (tt) menyatakan, sepuluh berbanding satu adalah perbandingan yang ideal ketika *user* memindai QR Code. Jika QR Code berukuran 1x1 cm, jarak pemindaian yang ideal adalah 10 cm dari QR Code. Selain itu, sudut yang dibuat antara alat pemindai dan QR Code harus sebesar 90° .

QR Code yang digunakan pada sistem UMN LibConn telah digunakan pada buku-buku perpustakaan UMN sebagai *book identification*. QR Code tersebut akan dipindai dengan menggunakan *scanner* tertentu pada ponsel Android. *Library scanner* yang digunakan pada aplikasi ini diperoleh dari tim ZXing, yang

diantaranya adalah Sean Owen dan Daniel Switkin. Penggunaan *library* tersebut disesuaikan kembali berdasarkan kebutuhan sistem.

2.2. Geolocation

Definisi *geolocation* pada umumnya dikaitkan dengan istilah GPS (*Global Positioning System*). Akan tetapi, GPS hanyalah salah satu teknologi yang digunakan untuk mengimplementasikan *geolocation* (Carmen, 2013). Carmen menyatakan, *geolocation* mengacu pada kemampuan untuk menentukan posisi dari suatu objek atau manusia dengan menggunakan koordinat geografisnya (lintang dan bujur) pada suatu peta melalui teknologi-teknologi tertentu. Selain itu, Janssen menyatakan definisi *geolocation* sebagai proses pencarian, penentuan, dan penyajian lokasi dari suatu komputer secara akurat.

2.2.1. Teknologi pada Geolocation

Setiap teknologi yang digunakan pada *geolocation* akan menentukan proses pemerolehan informasinya masing-masing seperti melalui identifikasi *IP Address*, *network provider*, GPS. Berikut ini uraian beberapa teknologi yang digunakan dalam *geolocation* (Carmen, 2013).

1. GPS (*Global Positioning System*)

Adalah teknologi yang memanfaatkan GPS *receiver* dalam mendapatkan posisi geografis suatu objek. Walaupun teknologi ini membutuhkan waktu yang lama dalam pemrosesannya, namun hasil geografis yang didapat melalui GPS jauh lebih akurat dibandingkan yang lainnya. GPS diimplementasikan dengan suatu jaringan dari dua puluh empat satelit yang ditempatkan di orbit luar angkasa dengan jarak dua puluh ribu kilometer dari

Bumi. Satelit-satelit tersebut memiliki *atomic clocks* yang memungkinkan waktu dan posisi dapat dikirim secara akurat;

2. *Wi Fi networks proximity*

Teknologi ini menganalisis nama dan *MAC address* dari suatu jaringan Wi-Fi yang akan digunakan untuk mengetahui posisi seseorang. Istilah *hybrid system* digunakan ketika teknologi Wi-Fi dikombinasikan dengan GPS sebagai solusi dari *indoor geolocation*, situasi ketika GPS sulit digunakan karena sinyal yang terhalang oleh berbagai material padat;

3. *mobile repeater triangulation*

Teknologi ini paling sering digunakan pada *mobile devices browsers* yang penggunaannya didasarkan pada jarak dengan beberapa *cell tower* tertentu. Pemrosesan yang terjadi pada teknologi ini sangat cepat. Akan tetapi, biaya yang dibutuhkan sangat mahal. Keakuratan yang diperoleh sangat baik jika seseorang berada di lingkungan kota yang memiliki banyak *phone tower*. Sebaliknya, masalah pemerolehan lokasi akan terjadi jika seseorang berada di lingkungan pedesaan dengan jumlah menara yang minim;

4. *IP Address*

Teknologi ini memiliki tingkat keakuratan yang lebih buruk dibandingkan dengan teknologi lainnya karena *IP Address* yang digunakan adalah *IP address* yang disediakan oleh ISP (*Internet Service Provider*), bukan *IP Address* dari perangkat *user* sendiri.

Selain klasifikasi menurut Carmen, terdapat juga berbagai referensi yang dapat memperjelas penggunaan teknologi pada *geolocation*. Salah satunya

diungkapkan oleh Burch dalam artikel “Understanding Android: Three Ways to Find Your Location”. Beliau mengatakan terdapat tiga cara pada Android yang dapat menentukan keberadaan seseorang:

1. Cell-ID

Teknologi pencarian lokasi dengan *Cell-ID* menentukan posisi seseorang dengan menggunakan ID yang terdapat pada *cell tower*. Setiap ponsel akan selalu terkoneksi ke sebuah *cell tower* terdekat. Posisi dari *cell tower* itu sendiri disimpan sebagai *Cell ID* yang akan diakses oleh perangkat ponsel.

2. Wi-Fi

Pada umumnya, penggunaan Wi-Fi dalam pencarian lokasi diiringi dengan informasi Cell ID pada ponsel. Hal tersebut dilakukan untuk mendapatkan hasil pencarian yang lebih akurat.

3. GPS

Secara garis besar, proses pencarian lokasi menggunakan GPS diawali dengan *request* terhadap jarak dari satelit-satelit pada suatu jaringan dan menggambarkan titik perpotongan pada batas bidang yang telah ditentukan.

Makin banyak satelit yang merespon, estimasi lokasi yang diperoleh akan makin akurat.

Dalam proses pencarian lokasi, GPS dan *network* dapat digunakan bersamaan untuk menghasilkan pencarian terbaik. Android menyatakan, penggunaan GPS dan *Android's Network Location Provider* dapat diutilisasi untuk mendapatkan lokasi *user*: Walaupun GPS memiliki hasil yang lebih akurat, namun GPS hanya berfungsi dengan baik di luar ruangan. *Android's Network Location Provider* menentukan

lokasi *user* dengan menggunakan *cell tower* dan sinyal Wi-Fi yang memberikan hasil yang berfungsi, baik di luar maupun di dalam ruangan.

2.2.2. Tantangan dan Strategi Geolocation pada Android

Proses untuk mendapatkan lokasi *user* dari sebuah perangkat *mobile* dapat menjadi sebuah hal yang rumit. Alasan-alasan yang dapat menyebabkan kesalahan dan ketidakakuratan dalam penentuan lokasi *user* (Android):

1. *multitude of location sources*

GPS, *Cell-ID*, dan Wi-Fi dapat menunjukkan lokasi keberadaan *user* dengan caranya masing-masing. Proses pemilihan teknologi yang akan digunakan relatif terhadap keakuratan, kecepatan, dan efisiensi baterai;

2. *user movement*

Perpindahan *user* dalam pencarian lokasi menentukan hasil yang akan diperoleh. Oleh karena itu, perpindahan *user* perlu dicatat dengan mengestimasi kembali lokasi *user* secara berkala;

3. *varying accuracy*

Keakuratan yang didapat pada pengestimasian lokasi tidaklah selalu konsisten. Lokasi yang diperoleh beberapa detik lalu mungkin saja lebih akurat dibandingkan dengan lokasi yang baru saja diperoleh.

Demi keakuratan dan pengalaman *geolocation* yang responsif, berikut strategi-strategi yang dapat digunakan ketika menentukan lokasi (Android).

1. *requesting location update*

Mendapatkan lokasi pengguna pada Android dilakukan dengan memanfaatkan kelas `LocationManager` dan memanggil

fungsi `requestLocationUpdates()` dan mengirimkan parameter `locationListener` yang telah dibuat. `locationListener` tersebut akan menerapkan beberapa metode *callback* yang dipanggil oleh `LocationManager` ketika terjadi perubahan lokasi atau perubahan status layanan.

```
1 // Acquire a reference to the system Location Manager
2 LocationManager locationManager = (LocationManager) this.getSystemService(Context.LOCATION_SERVICE);
3
4 // Define a listener that responds to location updates
5 LocationListener locationListener = new LocationListener() {
6     public void onLocationChanged(Location location) {
7         // Called when a new location is found by the network location provider.
8         makeUseOfNewLocation(location);
9     }
10
11     public void onStatusChanged(String provider, int status, Bundle extras) {}
12
13     public void onProviderEnabled(String provider) {}
14
15     public void onProviderDisabled(String provider) {}
16 };
17
18 // Register the listener with the Location Manager to receive location updates
19 locationManager.requestLocationUpdates(LocationManager.NETWORK_PROVIDER, 0, 0, locationListener);
```

Gambar 2.14 Kode *Request Location Update*
(Sumber: Android)

Parameter pertama dalam `requestLocationUpdates()` adalah tipe layanan lokasi yang digunakan (dalam contoh ini digunakan *Network Location*). Frekuensi *listener* dalam menerima *update* dapat ditentukan dengan parameter kedua dan ketiga. Parameter kedua menentukan selang waktu minimal *listener* dalam menerima *update*, sedangkan parameter ketiga menentukan perubahan jarak minimal *listener* dalam menerima *update*. Kemudian, parameter terakhir menunjukkan objek kelas dari `locationListener`. Untuk menerima *update* lokasi dari `GPS_PROVIDER` atau `NETWORK_PROVIDER`, aplikasi perlu meminta izin pengguna dengan mendeklarasikan `ACCESS_COARSE_LOCATION` atau `ACCESS_FINE_LOCATION` dalam *file* `AndroidManifest`.


```

1 <manifest ... >
2   <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
3   ...
4 </manifest>

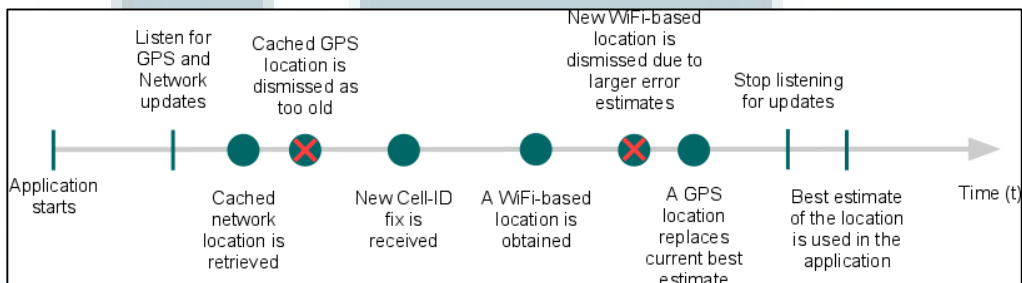
```

Gambar 2.15 Kode *Request User Permission*
(Sumber: Android)

2. *flow for obtaining user location*

Berikut ini adalah bagian yang menjelaskan alur yang dilakukan untuk memperoleh lokasi pengguna.

- A. Memulai aplikasi;
- B. Beberapa waktu kemudian, dengarkan perubahan lokasi dari penyedia;
- C. Pertahankan lokasi terbaik saat ini (*current best estimate*) dengan memilih lokasi yang baru tetapi tidak lebih akurat;
- D. Berhenti mendengarkan perubahan lokasi;
- E. Ambil lokasi terbaik saat ini untuk digunakan dalam aplikasi.



Gambar 2.16 *Timeline* Aplikasi dalam Mendengarkan Perubahan Lokasi
(Sumber: Android)

3. *deciding when to start listening for updates*

Aplikasi dapat langsung memulai mendengarkan perubahan lokasi ketika aplikasi tersebut diaktifkan atau ketika pengguna mengaktifkan fitur tertentu. Hal yang perlu diperhatikan adalah jangka waktu yang panjang dalam mendeteksi perubahan lokasi dapat menghabiskan daya baterai secara

signifikan. Sebaliknya, penggunaan dalam jangka waktu yang pendek akan berimbas pada minimnya akurasi.

4. *getting a fast fix with the last known location*

Waktu yang diperlukan oleh *listener* dalam mendapatkan lokasi pertama biasanya memerlukan waktu yang cukup lama bagi pengguna. Sebelum sebuah lokasi yang lebih akurat dapat diperoleh, aplikasi harus dapat memanfaatkan hasil pencarian lokasi terakhir dengan memanggil fungsi `getLastKnownLocation(String)`.

```
1 String locationProvider = LocationManager.NETWORK_PROVIDER;
2 // Or use LocationManager.GPS_PROVIDER
3
4 Location lastKnownLocation = locationManager.getLastKnownLocation(locationProvider);
```

Gambar 2.17 Kode *Get Last Known Location*
(Sumber: Android)

5. *deciding when to stop listening for updates*

Logika ketika menentukan lokasi yang baru tidak lagi diperlukan dapat bersifat sederhana atau kompleks. Perlu diingat bahwa dengan mendeteksi perubahan lokasi untuk waktu yang lama dapat menghabiskan daya baterai. Oleh karena itu, ketika data lokasi telah didapatkan, aplikasi harus segera berhenti mendeteksi perubahan lokasi dengan memanggil fungsi `removeUpdates(PendingIntent)`.

```
1 // Remove the listener you previously added
2 locationManager.removeUpdates(locationListener);
```

Gambar 2.18 Kode *Stop Listening Location Update*
(Sumber: Android)

6. *maintaining a current best estimate*

Data lokasi terbaru tidak berarti yang paling akurat karena tingkat akurasi yang berbeda-beda untuk setiap data lokasi yang diperoleh. Pemilihan lokasi terbaik dapat dilakukan dengan menerapkan beberapa kriteria. Kriteria-kriteria tersebut juga dapat bervariasi tergantung penggunaannya. Berikut ini beberapa langkah yang dilakukan untuk memvalidasi tingkat keakurasian dari suatu data lokasi.

- A. Cek apakah lokasi yang baru didapatkan jauh lebih baru dibandingkan dengan lokasi sebelumnya;
- B. Cek apakah lokasi yang baru didapatkan memiliki tingkat akurasi yang lebih baik dibandingkan dengan lokasi sebelumnya;
- C. Cek apakah data lokasi yang baru didapatkan berasal dari penyedia lokasi yang sama atau tidak dan tentukan penyedia lokasi mana yang lebih dipercaya.

UMMN