

**SKRIPSI**

**RANCANG BANGUN SISTEM DETEKSI DAN PEMETAAN  
JALAN BERLUBANG MENGGUNAKAN AKSELEROMETER  
DAN GPS PADA SMARTPHONE**

**Disusun dan diajukan oleh:**

**ILMI NURRAHMA ISMAIL**

**D121171515**



**DEPARTEMEN TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS HASANUDDIN**

**2022**

**LEMBAR PENGESAHAN SKRIPSI**

**RANCANG BANGUN SISTEM DETEKSI DAN PEMETAAN JALAN  
BERLUBANG MENGGUNAKAN AKSELEROMETER DAN GPS PADA  
SMARTPHONE**

**Disusun dan diajukan oleh**

**ILMI NURRAHMA ISMAIL**


**D121171515**

Telah dipertahankan di hadapan Panitia Ujian yang dibentuk dalam rangka Penyelesaian Studi Program Sarjana Program Studi Teknik Informatika Fakultas Teknik Universitas Hasanuddin pada tanggal 04 Juli 2022 dan dinyatakan telah memenuhi syarat kelulusan.

Menyetujui,

Pembimbing Utama,


Pembimbing Pendamping,

  
Dr. Ir. Zahir Zainuddin, M.Sc.  
Nip. 196404271989101002

  
Dr. Eng. Intan Sari Areni, ST., MT.  
Nip. 197502032000122002

Pft. Ketua Program Studi,



  
Dr. Amel Ahmad Ilham, S.T., M.IT.  
Nip. 197310101998021002

## PERNYATAAN KEASLIAN

Yang bertanda tangan di bawah ini:

Nama : Ilmi Nurrahma Ismail  
NIM : D121171515  
Departemen : Teknik Informatika  
Jenjang : S1

Menyatakan dengan ini karya tulisan saya berjudul:

RANCANG BANGUN SISTEM DETEKSI DAN PEMETAAN JALAN  
BERLUBANG MENGGUNAKAN AKSELEROMETER DAN GPS PADA  
SMARTPHONE

Adalah karya tulisan saya sendiri dan bukan merupakan pengambilan alihan tulisan orang lain bahwa skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri.

Apabila di kemudian hari terbukti atau dapat dibuktikan bahwa sebagian atau keseluruhan skripsi ini hasil karya orang lain, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Makassar, 04 Juli 2022

Yang Menyatakan,



Ilmi Nurrahma Ismail

## **KATA PENGANTAR**

Puji dan syukur penulis panjatkan kepada Tuhan yang Maha Esa atas berkat, rahmat, dan karunia-Nya sehingga tugas akhir yang berjudul “RANCANG BANGUN SISTEM DETEKSI DAN PEMETAAN JALAN BERLUBANG MENGGUNAKAN AKSELEROMETER DAN GPS PADA SMARTPHONE” dapat diselesaikan dengan baik sebagai salah satu syarat dalam menyelesaikan jenjang Strata-1 pada Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin.

Penulis menyadari bahwa dalam penyusunan dan penulisan tugas akhir ini tidak lepas dari bantuan, bimbingan, serta dukungan dari berbagai pihak, dari masa perkuliahan hingga penyusunan tugas akhir. Oleh sebab itu, penulis menyampaikan terima kasih kepada:

1. Kedua orang tua penulis, Bapak Ir. Ismail Derang dan Ibu Dra. Hasriani yang tidak pernah lelah dalam mendidik dan memberikan dukungan, doa, serta semangat kepada penulis.
2. Bapak Dr. Ir. Zahir Zainuddin, M.Sc., selaku pembimbing utama dan Ibu Dr. Eng. Intan Sari Areni, S.T., M.T., selaku pembimbing pendamping yang selalu menyediakan waktu, tenaga, pikiran, dan perhatian yang luar biasa dalam mengarahkan penulis menyelesaikan tugas akhir.
3. Bapak Dr. Amil Ahmad Ilham, S.T., M.IT., selaku Ketua Departemen Teknik Informatika atas segala bimbingan dan dukungan selama masa perkuliahan.

4. Ibu Anugrayani Bustamin, S.T., M.T., selaku Dosen Pembimbing Akademik penulis yang selalu membimbing dan menyediakan waktu, tenaga, dan perhatiannya selama masa perkuliahan.
5. Segenap Dosen dan Staf Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin yang telah banyak membantu semasa perkuliahan dan dalam penyelesaian tugas akhir.
6. Teman-teman RECOGNIZER atas dukungan, bantuan, semangat, serta pengalamannya yang diberikan selama ini.
7. Para sahabat serta seluruh pihak yang tidak sempat disebutkan dan tanpa sadar telah menjadi inspirasi serta banyak membantu penulis dalam penyelesaian tugas akhir.

Akhir kata, penulis menyadari bahwa tugas akhir ini masih jauh dari kata sempurna, oleh karenanya diharapkan segala bentuk saran serta masukan yang membangun dari berbagai pihak. Semoga tugas akhir ini dapat memberikan sumbangsih dan manfaat besar bagi kepentingan bersama.

Gowa, 01 Maret 2022

Penulis

## ABSTRAK

Secara teknis, kerusakan jalan menunjukkan suatu kondisi dimana struktural dan fungsional jalan sudah tidak dapat memberikan pelayanan yang optimal terhadap lalu lintas yang melintasi jalan tersebut. Kecelakaan mayoritas melibatkan kendaraan jenis sepeda motor yaitu sekitar 72% dimana faktor jalan berlubang merupakan penyebab utama adanya kecelekaan tersebut. Metode yang digunakan dalam sistem pemantauan kondisi jalan dibedakan menjadi metode gambar dua dimensi dan metode getaran. Metode gambar dua dimensi terbatas akan kondisi serta membutuhkan pemasangan dan pemeliharaan dengan biaya tinggi. Dalam mendeteksi jalan berlubang, penelitian ini memanfaatkan akselerometer dan GPS pada *smartphone* dengan menggunakan algoritma *Z-Thresh*. Apabila terdeteksi lubang, data akan disimpan lalu ditampilkan pada tampilan peta digital. Analisis kinerja sistem dilakukan menggunakan *Confusion Matrix* untuk memperoleh akurasi. Dari 20 pengujian yang dilakukan, pengujian pada *threshold* dengan nilai lebih besar dari 0 dan lebih kecil atau sama dengan 5 memperoleh akurasi sebesar 66.00%, kemudian pada *threshold* dengan nilai lebih besar dari -5 dan lebih kecil atau sama dengan 0 memperoleh akurasi terbaik sebesar 89.29%. Sedangkan pada *threshold* dengan nilai lebih kecil atau sama dengan -5 memiliki akurasi sebesar 43.05%. Sementara itu, pengujian untuk mengetahui pengaruh kecepatan di atas 35 km/jam memiliki akurasi sebesar 64.76%.

Kata kunci: *Lubang, Akselerometer, GPS, Z-Thresh*

## DAFTAR ISI

KATA PENGANTAR .....	ii
ABSTRAK.....	iv
DAFTAR ISI.....	v
DAFTAR GAMBAR .....	vii
DAFTAR TABEL.....	ix
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang .....	1
1.2. Rumusan Masalah .....	3
1.3. Tujuan Penelitian.....	3
1.4. Manfaat Penelitian.....	4
1.5. Batasan Masalah.....	4
1.6. Sistematika Penulisan.....	4
BAB II TINJAUAN PUSTAKA .....	6
2.1 Jalan.....	6
2.2 Lubang (Pothole).....	15
2.3 Android.....	17
2.4 GPS.....	24
2.5 Google Maps API.....	26
2.6 Firebase .....	27
2.7 Java.....	31
2.8 Algoritma Z-Thresh.....	34
2.9 Confusion Matrix .....	35
BAB III METODOLOGI PENELITIAN .....	37
3.1 Tahapan Penelitian .....	37
3.2 Waktu dan Lokasi Penelitian.....	38
3.3 Instrumen Penelitian.....	38
3.4 Perancangan Sistem.....	39
3.4.1. Perancangan Basis Data.....	41
3.4.2. Perancangan Antarmuka .....	42
3.4.3. Perancangan Proses Data .....	45
3.4.4. Perancangan Pengukuran Kecepatan .....	50

3.5	Pengambilan Data .....	51
3.6	Analisis Kinerja Sistem .....	53
BAB IV HASIL DAN PEMBAHASAN .....		55
4.1	Hasil Penelitian .....	55
4.2	Pembahasan .....	58
BAB V PENUTUP .....		61
5.1	Kesimpulan.....	61
5.2	Saran.....	62
DAFTAR PUSTAKA .....		63
LAMPIRAN.....		66



## DAFTAR GAMBAR

Gambar 2. 1 Perkerasan Kaku.....	9
Gambar 2. 2 Perkerasan Lentur.....	10
Gambar 2. 3 Perkerasan Blok .....	10
Gambar 2. 4 Retak (Cracking) .....	12
Gambar 2. 5 Distorsi .....	13
Gambar 2. 6 Cacat Permukaan.....	13
Gambar 2. 7 Pengausan.....	14
Gambar 2. 8 Kegemukan .....	15
Gambar 2. 9 Lubang (Pothole).....	16
Gambar 2. 10 Desain Standar Lubang .....	17
Gambar 2. 11 Prinsip Kerja Akselerometer .....	21
Gambar 2. 12 Fungsi getRotationMatrix .....	23
Gambar 2. 13 Sistem Koordinat Global.....	23
Gambar 2. 14 Google Maps .....	26
Gambar 2. 15 Firebase .....	28
Gambar 2. 16 Algoritma Z-Thresh .....	34
Gambar 2. 17 Flowchart Z-Thresh.....	35
Gambar 3. 1 Tahapan Penelitian .....	37
Gambar 3. 2 Alur Umum Sistem .....	39
Gambar 3. 3 Desain Sistem.....	40
Gambar 3. 4 Use Case Diagram.....	41
Gambar 3. 5 Data Model.....	41
Gambar 3. 6 Pseudocode Database .....	42
Gambar 3. 7 Antarmuka Pengumpulan Data .....	43
Gambar 3. 8 Antarmuka Peta Digital.....	45
Gambar 3. 9 Flowchart Proses Data.....	46
Gambar 3. 10 Sistem Koordinat Perangkat.....	48
Gambar 3. 11 Sistem Koordinat Kendaraan .....	48
Gambar 3. 12 Pseudocode Reorientasi .....	48
Gambar 3. 13 Pseudocode Z-Thresh.....	50
Gambar 3. 14 Pseudocode Speed.....	51

Gambar 3. 15 Denah Pengambilan Data.....	52
Gambar 3. 16 Proses Pengambilan Data.....	53
Gambar 4. 1 Grafik Akurasi Pengujian Kinerja Sistem.....	60

## **DAFTAR TABEL**

Tabel 3. 1 Tata Cara Pengujian Kinerja Sistem .....	54
Tabel 4. 1 Pengujian Nilai Threshold .....	55
Tabel 4. 2 Skenario Pengujian Kinerja Sistem .....	57
Tabel 4. 3 Hasil Pengujian Kinerja Sistem .....	58

# **BAB I**

## **PENDAHULUAN**

### **1.1. Latar Belakang**

Jalan merupakan prasarana yang sangat menunjang bagi kebutuhan hidup masyarakat. Dalam dimensi yang lebih luas jaringan jalan mempunyai peranan yang sangat penting dalam pengembangan wilayah, baik wilayah nasional, regional, maupun kabupaten/kota sesuai dengan fungsi dari jaringan jalan tersebut. Kondisi jalan yang baik akan memudahkan mobilitas penduduk dalam mengadakan hubungan perekonomian dan kegiatan sosial lainnya. Sedangkan jika terjadi kerusakan jalan akan berakibat bukan hanya terhalangnya kegiatan ekonomi dan sosial namun dapat terjadi kecelakaan. Ketersediaan jalan yang baik dan stabil berpengaruh terhadap kelancaran arus lalu lintas (Yoder, 1975).

Secara teknis, kerusakan jalan menunjukkan suatu kondisi dimana struktural dan fungsional jalan sudah tidak dapat memberikan pelayanan yang optimal terhadap lalu lintas yang melintasi jalan tersebut (Agus Sumarsono, 2017). Jenis kerusakan jalan seperti jalan berlubang dapat sangat membahayakan pengendara di jalan. Menurut Edo Rusyanto, Koordinator Jaringan Aksi Keselamatan Jalan, sekitar 15% dari total kecelakaan di faktor jalan disebabkan oleh jalan berlubang. Kecelakaan mayoritas melibatkan kendaraan jenis sepeda motor yaitu sekitar 72%. Sedangkan faktor jalan berlubang merupakan penyebab utama adanya kecelekaan tersebut.

Secara umum, metode yang digunakan dalam sistem pemantauan kondisi jalan dibedakan menjadi metode gambar dua dimensi dan metode getaran. Metode

gambar dua dimensi terbatas akan kondisi serta membutuhkan pemasangan dan pemeliharaan dengan biaya tinggi. Deteksi kondisi jalan menggunakan metode gambar dua dimensi yang memanfaatkan metode *template matching* cenderung memerlukan waktu yang lama sehingga informasi kerusakan jalan tersebut akan kurang bermanfaat bagi pengendara secara cepat (Hartono et al., 2017). Oleh sebab itu, pemanfaatan akselerometer pada *smartphone* dapat menjadi salah satu solusi terbaik dalam mendeteksi jalan berlubang.

Berdasarkan penelitian yang telah dilakukan sebelumnya, yakni yang pertama ialah CRSM (Chen et al., n.d.) atau disebut juga *crowdsourcing-based road surface monitoring system* dimana dapat mendeteksi jalan berlubang serta mengevaluasi tingkat kekasaran permukaan jalan. Akan tetapi pada penelitian ini mengalami kesulitan dalam membedakan *event* pada saat melewati *bumps* (polisi tidur) dengan lubang dikarenakan memiliki getaran yang mirip. Oleh karenanya, CRSM menggunakan informasi geografis keadaan jalan dalam mengeliminasi jenis *event* seperti lubang dan polisi tidur berdasarkan informasi GPS. Penelitian selanjutnya yang menggunakan sensor akselerometer dan GPS yang diletakkan pada kendaraan untuk memantau keadaan permukaan jalan ialah Pothole Patrol (Grunwald et al., 2008). Pada penelitian ini data diolah menggunakan filter pengolahan sinyal untuk menolak *event* selain jalan berlubang. Penelitian yang juga memanfaatkan sensor akselerometer pada *smartphone* ialah Wolverine (Institute of Electrical and Electronics Engineers. & IEEE Communications Society., n.d.). Untuk mendeteksi *event bump* (polisi tidur), Rafi Bhoraskar dkk menggunakan sensor akselerometer yang perangkatnya terlebih dahulu direorientasi agar dapat digunakan pada posisi

apapun. Pada penelitian ini data yang dikumpulkan diolah menggunakan *Support Vector Machine* (SVM) untuk menentukan kondisi jalan bergelombang dan mulus.

Pada tugas akhir ini, penulis berfokus pada proses data akselerometer untuk mendeteksi jalan berlubang menggunakan *smartphone* secara *realtime*. Data akselerometer yang masuk terlebih dahulu diproses melalui tahap *reorientation* yang mana bertujuan agar tidak mempengaruhi pengukuran pada perangkat dalam posisi apapun di atas kendaraan. Data selanjutnya diolah menggunakan algoritma *Z-Thresh* untuk menghasilkan deteksi yang akurat. Keluaran dari sistem yang dibuat ialah adanya *database* terkait lokasi jalan berlubang yang telah dilalui oleh sistem beserta dengan keterangan waktu saat *event* terjadi. Harapannya, sistem ini dapat menjadi salah satu metode yang dipertimbangkan dalam proses pemantauan kondisi jalan terkhusus jalan berlubang.

## **1.2. Rumusan Masalah**

Adapun rumusan masalah dari penelitian ini ialah sebagai berikut:

1. Bagaimana mendeteksi dan memetakan jalan berlubang menggunakan akselerometer dan GPS pada *smartphone*?
2. Bagaimana akurasi penggunaan algoritma *Z-Thresh* dalam mendeteksi jalan berlubang?

## **1.3. Tujuan Penelitian**

Adapun tujuan dari penelitian ini ialah sebagai berikut:

1. Untuk membuat sistem yang dapat mendeteksi serta memetakan jalan berlubang menggunakan akselerometer dan GPS pada *smartphone*.

2. Untuk mengetahui akurasi penggunaan algoritma *Z-Thresh* dalam mendeteksi jalan berlubang

#### **1.4. Manfaat Penelitian**

Dengan dilakukannya penelitian ini, diharapkan manfaat yang didapatkan antara lain:

1. Adanya *database* terkait lokasi jalan berlubang yang dilalui oleh sistem.
2. Bagi Dinas PU, dapat digunakan sebagai metode deteksi jalan berlubang dalam kegiatan pemantauan kondisi jalan.

#### **1.5. Batasan Masalah**

1. Smartphone yang digunakan memiliki sensor akselerometer dan GPS.
2. Uji coba menggunakan kendaraan sepeda motor.
3. Data diambil pada jalan dengan kriteria:
  - Merupakan tipe jalan perkerasan lentur (*Flexible Pavement*).
  - Terdapat lubang (*Pothole*).
4. Diujikan pada kendaraan dengan estimasi kecepatan 20-45 km/jam.
5. Sistem membutuhkan koneksi internet dan tersambung GPS.
6. Sistem yang dibangun ialah aplikasi android dan peta digital.

#### **1.6. Sistematika Penulisan**

Berikut adalah gambaran singkat mengenai isi tulisan secara keseluruhan:

### **BAB I PENDAHULUAN**

Bab ini menguraikan dampak kerusakan jalan serta keterbatasan kondisi pada penelitian terdahulu yang menggunakan metode gambar dua dimensi

yang mana menjadi latar belakang penelitian. Pada bab ini dijelaskan pula rumusan masalah, tujuan penelitian yaitu untuk mendeteksi dan memetakan jalan berlubang menggunakan akselerometer dan GPS, manfaat penelitian, batasan masalah, dan sistematika penulisan.

## **BAB II TINJAUAN PUSTAKA**

Bab ini menjelaskan landasan teori yang menunjang setiap elemen percobaan yang dilakukan. Beberapa teori yang dijelaskan secara detail pada bab ini ialah konsep jalan, Android, GPS, Google Maps API, Firebase, Java, Algoritma *Z-Threshold*, serta Confusion Matrix.

## **BAB III METODOLOGI PENELITIAN**

Bab ini berisi analisis kebutuhan sistem deteksi dan pemetaan jalan berlubang menggunakan akselerometer dan GPS pada *smartphone*, perancangan sistem, implementasi algoritma, proses pengumpulan data, dan skenario pengujian kinerja sistem deteksi dan pemetaan jalan berlubang.

## **BAB IV HASIL DAN PEMBAHASAN**

Bab ini menjelaskan secara keseluruhan hasil pengujian sistem deteksi dan pemetaan jalan berlubang menggunakan akselerometer dan GPS pada *smartphone* pada tiga kondisi *threshold*, dua rentang kecepatan yang berbeda, dan dilakukan pada tujuh lokasi jalan berlubang.

## **BAB V PENUTUP**

Bab ini berisi kesimpulan dari hasil penelitian dan uraian saran untuk pengembangan lebih lanjut.



## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Jalan**

Jalan adalah prasarana transportasi darat yang meliputi segala bagian jalan, termasuk bangunan pelengkap dan perlengkapannya yang diperuntukkan bagi lalu lintas, yang berada pada permukaan tanah, di atas permukaan tanah, di bawah permukaan tanah dan/atau air, serta di atas permukaan air, kecuali jalan kereta api, jalan lori, dan jalan kabel.

##### **2.1.1. Pengelompokan jalan**

Menurut UU No. 38 Tahun 2004, jalan pada umumnya digolongkan dalam 4 klasifikasi, yaitu:

- Klasifikasi berdasarkan sistem jalan.
  1. Jalan Primer, merupakan sistem jaringan jalan dengan peranan pelayanan distribusi barang dan jasa untuk pengembangan semua wilayah di tingkat nasional, dengan menghubungkan semua simpul jasa distribusi yang berwujud pusat-pusat kegiatan.
  2. Jalan Sekunder, merupakan sistem jaringan jalan dengan peranan pelayanan distribusi barang dan jasa untuk masyarakat di dalam kawasan perkotaan.
- Klasifikasi berdasarkan fungsional jalan.
  1. Jalan Arteri, yakni jalan umum yang berfungsi melayani angkutan utama dengan ciri perjalanan jarak jauh, kecepatan rata-rata tinggi, dan jumlah jalan masuk dibatasi secara berdaya guna.

2. Jalan Kolektor, merupakan jalan umum yang berfungsi melayani angkutan pengumpul atau pembagi dengan ciri perjalanan jarak sedang, kecepatan rata-rata sedang, dan jumlah jalan masuk dibatasi.
  3. Jalan Lokal, merupakan jalan umum yang berfungsi melayani angkutan setempat dengan ciri perjalanan jarak dekat, kecepatan rata-rata rendah, dan jumlah jalan masuk tidak dibatasi.
  4. Jalan Lingkungan, merupakan jalan umum yang berfungsi melayani angkutan lingkungan dengan ciri perjalanan jarak dekat, dan kecepatan rata-rata rendah.
- Klasifikasi jalan berdasarkan status.
    1. Jalan Nasional, merupakan jalan arteri dan jalan kolektor dalam sistem jaringan jalan primer yang menghubungkan antar ibukota provinsi, dan jalan strategis nasional, serta jalan tol.
    2. Jalan Provinsi, merupakan jalan kolektor dalam sistem jaringan jalan primer yang menghubungkan ibukota provinsi dengan ibukota kabupaten/kota, atau antar ibukota kabupaten/kota, dan jalan strategis provinsi.
    3. Jalan Kabupaten, merupakan jalan lokal dalam sistem jaringan jalan primer yang menghubungkan ibukota kabupaten dengan ibukota kecamatan, antaribukota kecamatan, ibukota kabupaten dengan pusat kegiatan lokal, antarpusat kegiatan lokal, serta jalan umum dalam sistem jaringan jalan sekunder dalam wilayah kabupaten, dan jalan strategis kabupaten.

4. Jalan Kota, adalah jalan umum dalam sistem jaringan jalan sekunder yang menghubungkan antarpusat pelayanan dalam kota, menghubungkan pusat pelayanan dengan persil, menghubungkan antarpusat permukiman yang berada di dalam kota.
  5. Jalan Desa, merupakan jalan umum yang menghubungkan kawasan dan/atau antarpermukiman di dalam desa, serta jalan lingkungan.
- Klasifikasi Jalan berdasarkan beban muatan sumbu atau disebut juga kelas jalan berdasarkan UU No. 14 Tahun 1992 tentang Lalu Lintas dan Angkutan Jalan.
    1. Jalan Kelas I, yaitu jalan arteri yang dapat dilalui kendaraan bermotor lebar muatan tidak melebihi 2.500 milimeter dan panjang 18.000 milimeter, dengan sumbu terberat yang diizinkan lebih dari 10 ton.
    2. Jalan Kelas II, yaitu jalan arteri yang dapat dilalui kendaraan bermotor dengan lebar muatan tidak melebihi 2.500 milimeter dan panjang 18.000 milimeter, dengan sumbu terberat diizinkan 10 ton. Contoh penggunaan jalan ini adalah yang sesuai dengan angkutan peti kemas.
    3. Jalan Kelas III A, yaitu jalan arteri atau kolektor yang dapat dilalui kendaraan bermotor dengan lebar muatan tidak melebihi 2.500 milimeter dan panjang 18.000 milimeter, dengan sumbu terberat yang diizinkan 8 ton.

4. Jalan Kelas III B, yaitu jalan kolektor yang dapat dilalui kendaraan bermotor dengan lebar muatan tidak melebihi 2.500 milimeter dan panjang 12.000 milimeter, dengan sumbu terberat yang diizinkan 8 ton.

### **2.1.2. Tipe Jalan Berdasarkan Perkerasan Jalan**

Tipe jalan berdasarkan perkerasan jalan adalah jenis jalan yang dibedakan berdasarkan proses pembuatan jalan menggunakan beberapa metode yaitu:

1. Perkerasan Kaku (*Rigid Pavement*)

Merupakan jenis jalan yang terbuat dari beton semen, yaitu terdiri dari plat beton semen sebagai lapis pondasi di atas tanah dasar.



*Gambar 2. 1 Perkerasan Kaku*

2. Perkerasan lentur (*Flexible Pavement*)

Merupakan jenis jalan yang terbuat dari aspal atau hot mix.



*Gambar 2. 2 Perkerasan Lentur*

### 3. Perkerasan Blok

Merupakan jenis jalan yang terbuat dari campuran pasir dan semen atau disebut juga blok beton terkunci.



*Gambar 2. 3 Perkerasan Blok*

#### **2.1.3. Jenis Kerusakan Jalan**

Menurut Perkerasan Lentur Jalan Raya (Sukirman. S, 1992), kerusakan pada konstruksi perkerasan jalan dapat disebabkan oleh peningkatan beban lalu lintas, air, bahan konstruksi perkerasan, iklim, kondisi

tanah dasar yang labil maupun proses pelaksanaan yang kurang baik. Selain itu, kerusakan konstruksi pada perkerasan lentur dapat pula disebabkan oleh kelelahan permukaan, konsolidasi atau geser, pengembangan di dalam *subgrade*, *subbase*, dan *base course*.

Penilaian tipe dan kondisi kerusakan jalan yang ada merupakan aspek yang paling penting dalam penentuan sebuah proyek perbaikan jalan, sebab karakteristik inilah yang akan menentukan satuan nilai manfaat ekonomis yang ditimbulkan oleh adanya perbaikan jalan (Bina Marga, 1995).

Dalam laporannya, OECD (1997) menjelaskan tentang proses kemerosotan kinerja jalan yang terjadi melalui dua cara:

1. *Progresifly*, artinya kerusakan sejalan dengan habisnya waktu/umur rencana, hal ini bisa dideteksi untuk kemudian ditentukan perencanaan pekerjaan pemeliharaan yang harus dilakukan sehingga kerusakan yang lebih buruk bisa dihindari.
2. *Suddenly/Rapidly*, dimana kerusakan begitu saja terjadi pada area tertentu. Jenis kerusakan seperti ini akan beresiko pada struktur jalan yang ada dan membahayakan bagi pemakai jalan sehingga tidak ada alasan untuk menunda penanganannya.

Secara umum, terdapat 2 jenis kerusakan jalan yaitu kerusakan fungsional dan kerusakan struktural.

#### 1. Kerusakan Fungsional

Kerusakan fungsional merupakan kerusakan pada permukaan jalan yang dapat menyebabkan terganggunya fungsi jalan tersebut.

## 2. Kerusakan Struktural

Kerusakan pada struktur jalan yang sebagian atau seluruhnya menyebabkan perkerasan jalan tidak lagi menahan beban yang bekerja di atasnya.

Menurut manual pemeliharaan jalan No: 03/MN/B/1983 yang dikeluarkan oleh Direktorat Jenderal Bina Marga, Kerusakan jalan dapat dibedakan atas:

### 1. Retak (*cracking*)

Retak yang terjadi pada lapisan permukaan jalan terdiri dari retak halus (*hair cracking*), retak kulit buaya (*aligator crack*), retak pinggir (*edge crack*), retak sambungan bahu dan perkerasan (*edge joint crack*), retak sambungan jalan (*lane joint crack*), retak sambungan pelebaran jalan (*widening crack*), retak refleksi (*reflection crack*), retak susut (*shrinkage crack*), dan retak slip (*slippage crack*).



Gambar 2. 4 Retak (*Cracking*)

## 2. Distorsi (*Distortion*)

Distorsi/perubahan bentuk dapat terjadi akibat lemahnya tanah dasar, pemadatanyang kurang pada lapis pondasi, sehingga terjadi tambahan pemadatan akibat beban lalu lintas. Distorsi dapat dibedakan menjadi alur (*ruts*), keriting (*corrugation*), sungkur (*shoving*), amblas (*grade depressions*), dan jembul (*upheaval*).



*Gambar 2. 5 Distorsi*

## 3. Cacat permukaan (*sisintegration*)

Cacat permukaan yang mengarah kepada kerusakan secara kimiawi dan mekanis dari lapisan perkerasan. Yang termasuk dalam cacat permukaan ini adalah lubang (*pothole*), pelepasan butir (*Ravelling*), dan Pengelupasan lapisan permukaan (*Stripping*).



*Gambar 2. 6 Cacat Permukaan*



#### 4. Pengausan (*Polished Aggregate*)

Permukaan jalan menjadi licin, sehingga membahayakan kendaraan. Pengausan terjadi karena agregat berasal dari material yang tidak tahan aus terhadap roda kendaraan, atau agregat yang dipergunakan berbentuk bulat dan licin, tidak berbentuk cubical. Dapat diatasi dengan menutup lapisan dengan latasir, buras atau latasbun.



*Gambar 2. 7 Pengausan*

#### 5. Kegemukan (*Bleeding or flushing*)

Permukaan menjadi licin. Pada temperatur tinggi, aspal menjadi lunak dan akan terjadi jejak roda. Kegemukan (*bleeding*) dapat disebabkan pemakaian kadar aspal yang tinggi pada campuran aspal, pemakaian terlalu banyak aspal pada pekerjaan *prime coat* atau *tack coat*. Dapat diatasi dengan menaburkan agregat panas dan kemudian dipadatkan atau lapis aspal diangkat dan kemudian diberi lapisan penutup.



*Gambar 2. 8 Kegemukan*

6. Penurunan pada Bekas Penanaman Utilitas (*Utility Cut Depression*)

Terjadi di sepanjang bekas penanaman utilitas. Hal ini terjadi karena pemadatan yang tidak memenuhi syarat. Dapat diperbaiki dengan dibongkar kembali dan diganti dengan yang sesuai.

## **2.2 Lubang (Pothole)**

Lubang (*Pothole*) merupakan jenis kerusakan cacat permukaan yang mana mengarah pada kerusakan secara kimiawi dan mekanis dari lapisan perkerasan. Kerusakan dengan jenis lubang ini berbentuk seperti mangkok yang dapat menampung dan meresapkan air pada badan jalan. Penyebab kerusakan dapat berupa kadar aspal rendah, pelapukan aspal, penggunaan agregat kotor atau tidak baik, suhu campuran tidak memenuhi persyaratan, sistem drainase yang buruk, serta merupakan kelanjutan dari kerusakan lain seperti retak dan pelepasan butir.

Lubang dapat terjadi akibat:

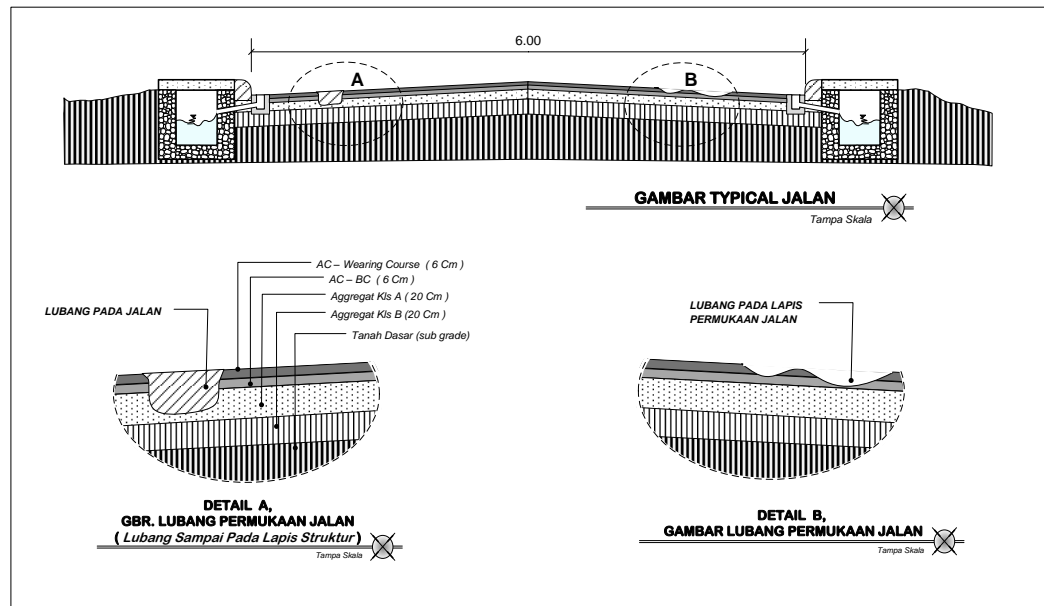
1. Campuran material lapis permukaan jelek, seperti:
  - Kadar aspal rendah, sehingga film aspal tipis dan mudah lepas.

- Agregat kotor sehingga ikatan antara aspal dan agregat tidak baik.
  - Temperatur campuran tidak memenuhi persyaratan.
2. Lapis permukaan tipis sehingga ikatan aspal dan agregat mudah lepas akibat pengaruh cuaca.
  3. Sistem drainase jelek, sehingga air banyak yang meresap dan mengumpul dalam lapis perkerasan.
  4. Retak-retak yang terjadi tidak segera ditangi sehingga air meresap dan mengakibatkan terjadinya lubang-lubang kecil.

Lubang diperbaiki dengan cara dibongkar dan dilapis kembali. Perbaikan yang bersifat permanen disebut juga *deep patch* (tambalan dalam).



*Gambar 2. 9 Lubang (Pothole)*



Gambar 2. 10 Desain Standar Lubang

## 2.3 Android

Android adalah sistem operasi berbasis Linux yang dirancang untuk perangkat bergerak 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. Pada tugas akhir ini, Sistem deteksi dan pemetaan jalan berlubang menggunakan akselerometer dan GPS pada *smartphone* akan dibangun dengan sistem operasi Android.

### 1.3.1. Komponen Aplikasi pada Android

Aplikasi Android menggunakan bahasa pemrograman Java. Android SDK (Software Development Kit) sebagai *compiler* kode aplikasi. Komponen

aplikasi adalah bagian penting dalam pengembangan aplikasi Android. Komponen dapat berupa *entry point*, yaitu kontrol penghubung antara sistem operasi dengan program aplikasi. Setiap komponen memiliki ketergantungan dengan komponen yang lain. Berikut ini adalah 4 tipe komponen aplikasi.

### **1. Activities**

*Activity* merepresentasikan sebuah tampilan pada antarmuka pengguna. Misalnya, aplikasi email memiliki satu *activity* yaitu untuk membuat email. Namun pada satu aplikasi dapat terdapat beberapa *activity* yang bekerjasama untuk membuat aplikasi yang dapat digunakan oleh *user* dengan mudah. Sebuah *activity* dapat diimplementasikan menggunakan subclass *activity*.

### **2. Services**

Komponen *service* merupakan komponen yang bekerja pada latar belakang untuk menjalankan sebuah operasi. *Service* tidak menyediakan antarmuka pengguna. Contoh penggunaan ini adalah menjalankan musik di latar belakang (*background*). Sebuah *service* dapat diimplementasikan menggunakan subclass *Services*.

### **3. SensorEventListener**

*SensorEventListener* merupakan antarmuka publik yang terdapat pada Android yang berfungsi untuk menerima pemberitahuan pembaharuan data dari *sensorManager*. *Public Method* yang digunakan ada 2 yaitu *onAccuracyChanged* dan *onSensorChanged*. *Public Method onAccuracyChanged* adalah

*listener* aplikasi untuk perubahan akurasi pada sensor yang sedang digunakan, sedangkan *onSensorChanged* adalah *listener* perubahan nilai sensor.

#### **4. ValueEventListener**

*ValueEventListener* adalah salah satu jenis *Listener* pada Firebase Database yang terintegrasi dalam android yang mana merupakan metode yang mendengarkan perubahan data ke lokasi tertentu di database, misalnya suatu node. *ValueEventListener* memiliki satu metode *event callback*, yaitu *onDataChange()* untuk membaca cuplikan statis konten pada jalur tertentu, seperti yang ada pada saat *event*. Metode ini dipicu satu kali saat listener terpasang dan dipicu lagi setiap kali data, termasuk *child*, berubah. *Event Callback* diteruskan dengan snapshot yang berisi semua data di lokasi itu, termasuk data turunan (*child*). Jika tidak ada data, snapshot yang dikembalikan adalah null.

#### **5. FusedLocationProviderClient**

*Fused location provider* adalah salah satu API lokasi di layanan Google Play yang digunakan untuk mengambil lokasi terakhir perangkat yang diketahui. Penggunaan *fused location provider* dapat mengoptimalkan penggunaan daya baterai perangkat. *Fused location provider* diakses jika proyek pengembangan aplikasi menyertakan layanan Google Play.

FusedLocationProviderClient menyediakan beberapa metode untuk mengambil informasi lokasi perangkat, antara lain:

- a. *getLastLocation()* mendapatkan perkiraan lokasi lebih cepat dan meminimalkan penggunaan baterai yang dapat diatribusikan ke aplikasi. Namun, informasi lokasi dapat kadaluarsa jika tidak ada *clients* lain yang aktif menggunakan lokasi tersebut.
- b. *getCurrentLocation()* mendapatkan lokasi terbaru dan akurat secara lebih konsisten. Namun, metode ini dapat menyebabkan perhitungan lokasi aktif terjadi pada perangkat.

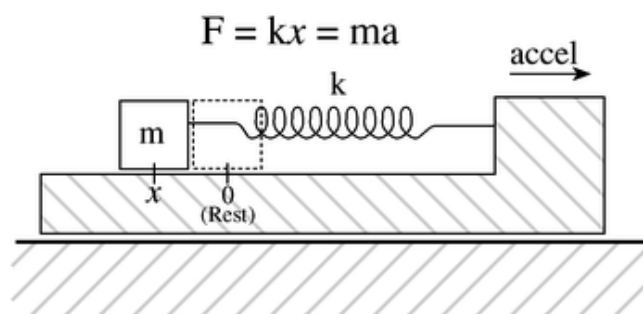
### **2.3.2. Sensor pada Aplikasi**

Sebagian besar perangkat Android telah dilengkapi oleh sensor yang dapat mengukur gerakan, orientasi, dan berbagai jenis keadaan lingkungan sekitarnya. Sensor ini mampu menyediakan data mentah dengan akurasi yang tinggi. Sensor yang digunakan pada penelitian ini adalah sensor akselerometer.

Akselerometer digunakan untuk mengukur percepatan, mendeteksi dan mengukur getaran (vibrasi), dan mengukur percepatan akibat gravitasi. Akselerometer melihat akselerasi terkait dengan fenomena berat yang dialami oleh massa uji pada kerangka acuan perangkat akselerometer. Sebagai contoh, akselerometer di permukaan bumi akan mengukur akselerasi  $g = 9,81 \text{ m/s}^2$

lurus ke atas karena beratnya. Sebaliknya, akselerometer jatuh bebas atau di luar angkasa akan mengukur nol.

Prinsip kerja yang digunakan adalah prinsip percepatan (*acceleration*). Prinsip dasarnya adalah sebuah sistem massa pegas sederhana yang sesuai dengan Hukum Hooke ( $F = kx$ ) dan juga Hukum II Newton ( $F = ma$ ). Jika sistem ini mengalami percepatan, maka sesuai dengan hukum Newton, akan menimbulkan gaya yang menggantikan massa. Gaya ini menyebabkan massa bergeser. Oleh karena itu percepatan akan menyebabkan massa menjadi tergeser oleh  $x = ma/k$ , jika kita mengamati perpindahan  $x$ , diketahui bahwa massa mengalami percepatan  $= kx/m$ . Dengan cara ini, dapat diketahui pengukuran percepatan menjadi salah satu pengukuran perpindahan massa yang terhubung dengan pegas.



Gambar 2. 11 Prinsip Kerja Akselerometer

Saat ini hampir semua sensor/transduser akselerometer sudah dalam bentuk digital sehingga cara kerjanya hanya berdasarkan temperatur yang diolah secara digital dalam satu chip. Kebanyakan akselerometer adalah *Micro-Electro-Mechanical Sensor* (MEMS). Prinsip dasar operasi balik akselerometer MEMS adalah perpindahan kecil dari massa yang terukir di permukaan silikon dari sirkuit terintegrasi dan ditangguhkan oleh balok kecil.



Pada tugas akhir ini, sensor akselerometer pada Android akan digunakan untuk memperoleh data akselerasi. Data akselerasi yang digunakan adalah pada sumbu z yang telah direorientasi dari sumbu koordinat perangkat menjadi koordinat bumi.

### 2.3.3. Fungsi *getRotationMatrix*

Akselerometer memberikan vektor gravitasi yang bertindak vertikal ke bawah menuju pusat bumi. Magnetometer memberikan vektor magnetik, yang bertindak ke arah utara magnet. Perkalian silang dari dua vektor ini memberikan sebuah vektor yang tegak lurus terhadap bidang kedua vektor ini dan bekerja sepanjang magnet barat (aturan tangan kanan). Vektor ini selanjutnya disebut juga *East West Vector*. Perkalian silang Vektor Gravitasi dengan *East West Vector* akan memperoleh *North South Vector*, yang bekerja secara horizontal menuju utara magnet. Ketiga vektor yang saling tegak lurus ini yang mewakili sistem koordinat geometrik. Matriks rotasi yang mewakili sistem ini sebagai sistem target dapat dibentuk sebagai,

$$R^{-1} = \begin{bmatrix} \textit{East West Vector} \\ \textit{North South Vector} \\ \textit{Gravity Vector} \end{bmatrix} \quad (1)$$

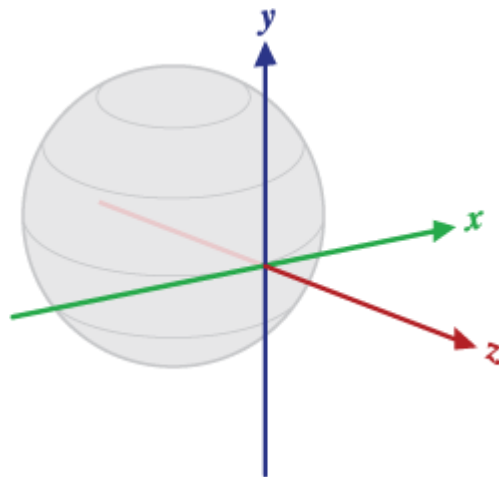
Matriks rotasi dapat diperoleh dari API sensor Android yang menggunakan fungsi *getRotationMatrix*.

```
public static boolean getRotationMatrix (float[] R,  
                                         float[] I,  
                                         float[] gravity,  
                                         float[] geomagnetic)
```

Gambar 2. 12 Fungsi *getRotationMatrix*

Fungsi *getRotationMatrix* menghitung matriks inklinasi I serta matriks rotasi R yang mengubah vektor dari sistem koordinat perangkat ke sistem koordinat global yang didefinisikan sebagai basis ortonormal langsung, di mana:

- X didefinisikan sebagai produk vektor Y.Z (tangensial ke tanah di lokasi perangkat saat ini dan kira-kira menunjuk ke Timur).
- Y bersinggungan dengan tanah di lokasi perangkat saat ini dan mengarah ke Kutub Utara magnet.
- Z menunjuk ke arah langit dan tegak lurus dengan tanah.



Gambar 2. 13 Sistem Koordinat Global

Parameter dalam fungsi *getRotationMatrix* memiliki definisi sebagai berikut:

1.  $R$ , matriks identitas ketika perangkat disejajarkan dengan sistem koordinat dunia, yaitu ketika sumbu  $X$  perangkat mengarah ke Timur, sumbu  $Y$  mengarah ke Kutub Utara dan perangkat menghadap ke langit.
2.  $I$ , matriks rotasi yang mentransformasikan vektor geomagnetik menjadi ruang koordinat yang sama dengan gravitasi.
3. *Gravity*, array yang berisi vektor gravitasi yang dinyatakan dalam koordinat perangkat.
4. *Geomagnetic*, array yang berisi vektor geomagnetik yang dinyatakan dalam koordinat perangkat.

## 2.4 GPS

*GPS* adalah sistem navigasi berbasis ruang yang menyediakan informasi lokasi dan waktu dalam segala kondisi cuaca, dimanapun di bumi yang tak terhalang pandangan 4 atau lebih satelit *GPS*. Sistem ini menggunakan 24 satelit yang mengirimkan gelombang mikro ke bumi. Sinyal ini kemudian diterima oleh alat penerima di permukaan bumi. Data yang diterima itu kemudian digunakan untuk menentukan posisi, kecepatan, arah, dan waktu. Satelit *GPS* mengitari bumi secara sirkular (mendekati bentuk elips) dengan inklinasi sebesar 55 derajat.

*GPS* pada awalnya dikelola oleh *US Air Force* (*USAF*) sebagai operator sistem. *GPS* yang pada awalnya dikembangkan untuk sistem kekuatan tambahan bagi militer memiliki potensi menguntungkan bagi masyarakat sipil. Pada akhirnya *GPS* menyediakan dua layanan, yaitu layanan *Precise Positioning Service* (*PPS*)

yang berguna untuk kekuatan militer AS dan layanan *Standar Positioning Service* (SPS) yang didesain untuk memberikan kemampuan *positioning*.

GPS memiliki 3 segmen, yaitu segmen angkasa, segmen kontrol, dan segmen pengguna. Segmen angkasa terdiri dari 24-satelit yang telah dijelaskan sebelumnya. Setiap satelit GPS secara terus menerus akan mentransmisikan gelombang sinyal radio mikro yang terdiri dari dua *carriers*, dua kode dan pesan navigasi. Ketika GPS receiver diaktifkan, GPS *receiver* akan mengambil sinyal melalui antena penerima. Setelah menerima sinyal GPS, sinyal tersebut akan diproses menggunakan software yang ada di dalam perangkat tersebut. Secara teori hanya dibutuhkan informasi tiga jarak antara GPS *receiver* dengan satelit, artinya dibutuhkan sebanyak minimal tiga satelit yang melacak secara bersamaan.

Pada segmen kontrol sistem GPS terdiri dari stasiun pelacak yang tersebar pada jaringan seluruh dunia, dengan kontrol utamanya berada di negara Amerika Serikat di Colorado Springs, Colorado. Tujuan utama segmen kontrol adalah melacak satelit GPS dengan maksud untuk menentukan dan memprediksi lokasi satelit, integritas satelit, perilaku satelit, jam atom, data atmosfer, dan lain-lain. Pada segmen pengguna sendiri terdiri dari seluruh bagian militer dan pengguna sipil. Dengan adanya GPS *receiver* yang tersambung pada antena GPS, pengguna dapat menerima sinyal GPS, yang dapat digunakan untuk menentukan posisi dirinya dimanapun ia berada. GPS sekarang ini tersedia kepada semua pengguna di seluruh dunia dengan gratis.

Dari beberapa pemakaian GPS, dikategorikan menjadi:

1. Waktu. GPS *receiver* menerima informasi waktu dari jam atom yang mempunyai keakurasian sangat tinggi.
2. Lokasi. GPS memberikan informasi lokasi, latitude, longitude, dan altitude.
3. Kecepatan. Ketika berpindah tempat, GPS dapat menunjukkan informasi kecepatan berpindah tersebut.
4. Arah perjalanan. GPS dapat menunjukkan arah tujuan.
5. 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.
6. Tracking. Membantu untuk memonitoring pergerakan obyek.

## 2.5 Google Maps API

Google Maps API adalah layanan oleh Google yang dapat digunakan untuk menampilkan data dalam bentuk peta digital melalui beberapa model fitur. Google Maps API menyediakan *web service* sebagai antarmuka untuk permintaan data Maps API dari layanan luar dan digunakan pada aplikasi Maps pengguna. Layanan *web* ini menggunakan HTTP *request* ke URL yang spesifik, meneruskan parameter URL sebagai argumen ke layanan.



Gambar 2. 14 Google Maps

Pada umumnya, layanan ini mengembalikan data dengan HTTP *request* berupa JSON atau XML untuk diuraikan dan atau diproses oleh aplikasi. Fitur yang dapat digunakan pada pengembangan sistem tugas akhir ini adalah menampilkan peta dan menandai peta menggunakan *marker*. Library yang digunakan pada penelitian ini ialah Maps SDK for Android.

## 2.6 Firebase

Firebase merupakan platform untuk aplikasi *realtime*. Ketika data berubah, maka aplikasi yang terhubung dengan firebase akan diperbarui secara langsung melalui setiap *device* (perangkat) baik *website* ataupun *mobile*. Firebase mempunyai *library* yang lengkap untuk sebagian besar *platformweb* dan *mobile* dan dapat digabungkan dengan berbagai *framework* lain seperti node, java, javascript, dan lain-lain. *Application Programming Interface* (API) untuk menyimpan dan sinkronisasi data akan disimpan sebagai bit dalam bentuk JSON (*JavaScript Object Notation*) pada *cloud* dan akan disinkronisasi secara *realtime*. Database milik *firebase* merupakan *database* yang bersifat non-relational atau NoSQL, dimana *database* ini merupakan jenis *database* yang tidak menggunakan sistem tabel dalam implementasinya serta tidak menyimpan data secara lokal pada perangkat melainkan pada awan. Selain itu, *firebase database* juga memiliki optimasi dan fungsionalitas yang berbeda bila dibandingkan dengan basis data relasional.



*Gambar 2. 15 Firebase*

Terdapat beberapa fitur yang disediakan oleh firebase adalah sebagai berikut:

1. *Analytics*

Fitur ini membuat para pengembang aplikasi dapat mengerti bagaimana para pengguna menggunakan aplikasi mereka. *SDK analytic* menangkap sendiri data yang dibutuhkan oleh para developer. *Dashboard* juga menyediakan detail seperti pengguna paling aktif atau fitur apa saja yang paling digunakan dalam aplikasi tersebut. *Analytic* juga menyediakan data yang telah dirangkum.

2. *Authentication*

Fitur ini membuat para pengembang dapat mengizinkan pengguna untuk mengakses aplikasi. Firebase menyediakan fitur login melalui Gmail, Github, Twitter, Facebook, dan juga autentikasi buatan sendiri.

3. *Messaging*

FCM (*Firestore Cloud Messaging*) membuat para pengguna dapat mengirimkan pesan ke berbagai *platform* tanpa biaya tambahan. *Messaging* juga dapat digunakan untuk kebutuhan notifikasi.

4. *Realtime Database*

Database di Firebase adalah database berbasis *cloud* dan tidak membutuhkan SQL untuk mengambil dan menyimpan data atau bisa juga disebut NoSQL. Database ini sangatlah cepat dan dapat diandalkan yang

artinya data dapat dibaharui dan disinkronisasi dengan cepat. Data juga dijaga meskipun pengguna kehilangan koneksi internetnya.

#### 5. *Storage*

Firestore juga menyediakan fasilitas penyimpanan. Firestore dapat menyimpan dan mengambil konten seperti gambar, video, dan audio langsung dari SDK. Data yang disimpan akan aman dan hanya pengguna yang diijinkan yang dapat mengaksesnya.

#### 6. *Hosting*

Firestore juga menyediakan fitur *hosting*. Firestore mengirimkan konten web secara cepat dan konten selalu dikirim dengan aman.

#### 7. *Crash reporting*

Fitur *crash reporting* di Firestore membuat pengembang dapat mengetahui kesalahan ketika terjadi *crash*.

### **2.6.1. Firestore Realtime Database**

*Firestore Realtime Database* merupakan basis data *online* yang dapat digunakan sebagai media penyimpanan data dari aplikasi. Data disimpan dalam bentuk JSON dan dapat disinkronkan secara *realtime* ke setiap *client* yang terhubung. Layanan ini memiliki 3 kemampuan inti yaitu:

1. *Realtime*, jika terdapat perubahan pada data *database*, maka seluruh *client* yang terhubung secara otomatis akan mendapatkan perubahannya dengan cepat.
2. *Offline*, yaitu aplikasi yang menggunakan fitur ini akan tetap responsif bahkan saat dalam keadaan luring. Hal ini disebabkan



karena Firebase SDK (*Software Development Kit*) dapat mempertahankan data dan perubahannya pada media penyimpanan *client*. Pada saat *client* terhubung ke jaringan internet, maka *Firebase SDK* akan melakukan penyesuaian otomatis atas catatan perubahan data yang disimpan pada media penyimpanan *client* dengan kondisi terkini dari *Firebase Server*.

3. *Accessible from client devices*. Layanan ini menawarkan kemudahan untuk mengakses *firebase realtime database* secara langsung dari sebuah perangkat *mobile* atau sebuah peramban web tanpa membutuhkan *server application*.

*Firebase Realtime Database* memungkinkan untuk membuat aplikasi kolaboratif dan kaya fitur dengan menyediakan akses yang aman ke database, langsung dari kode sisi *client*. Bahkan saat *offline* sekalipun, peristiwa realtime terus berlangsung, sehingga pengguna akhir akan merasakan pengalaman yang responsif. Ketika koneksi perangkat pulih kembali, *Realtime Database* akan menyinkronkan perubahan data lokal dengan update jarak jauh yang terjadi selama *client offline*, sehingga setiap perbedaan akan otomatis digabungkan.

Pada *Firebase Realtime Database*, data disimpan dalam format *JSON*, dan setiap node dari pohon *JSON* adalah *reference*. Dengan kata lain, referensi adalah tempat tertentu dimana data disimpan dalam *database*. Referensi dapat menyimpan berbagai jenis data seperti string atau angka, dan yang paling penting adalah objek. Objek atau node ini yang menyimpan data

pada gilirannya disebut *children*. Realtime database menyediakan bahasa aturan berbasis ekspresi yang fleksibel, atau disebut juga Aturan keamanan Firebase Realtime Database, untuk menentukan metode strukturisasi data dan kapan data dapat dibaca atau ditulis. Ketika diintegrasikan dengan Firebase Authentication, developer dapat menentukan siapa yang memiliki akses ke data tertentu dan bagaimana mereka dapat mengaksesnya.

API Realtime Database dirancang agar hanya mengizinkan operasi yang dapat dijalankan dengan cepat. Hal ini memungkinkan untuk membangun pengalaman realtime yang luar biasa dan dapat melayani jutaan pengguna tanpa mengorbankan kemampuan respons. Oleh karena itu, perlu dipikirkan bagaimana pengguna mengakses data, kemudian buat struktur data sesuai dengan kebutuhan tersebut.

## **2.7 Java**

Java adalah bahasa pemrograman tingkat tinggi yang bersifat umum/non-spesifik (general-purpose), berorientasi objek, berbasis kelas, konkuren, dan secara khusus didesain untuk memanfaatkan dependensi implementasi seminimal mungkin. Karena fungsionalitasnya yang memungkinkan aplikasi java mampu berjalan di beberapa platform sistem operasi yang berbeda. Bahasa pemrograman ini awalnya dibuat oleh James Gosling saat masih bergabung di Sun Microsystems yang saat ini merupakan bagian dari Oracle dan dirilis pada tahun 1995. Bahasa ini banyak mengadopsi sintaksis yang terdapat pada C dan C++ namun dengan sintaksis model objek yang lebih sederhana. Aplikasi-aplikasi berbasis java

umumnya dikompilasi ke dalam p-code (bytecode) dan dapat dijalankan pada berbagai Mesin Virtual Java (JVM).

Gosling (dibantu oleh rekan-rekannya Patrick Naughton, Chris Warth, Ed Frank, dan Mike Sheridan) mulai mendesain program tersebut dengan bahasa pemrograman C++, karena adanya sifat berorientasi obyek. Saat menggunakan C++, Gosling mulai merasakan bahwa bahasa tersebut tidak cocok untuk tugas yang dibebankan kepada timnya. Beberapa permasalahan yang muncul, seperti: perwarisan majemuk (*multiple inheritance*) dan bug program berupa kebocoran memori (*memory leaks*). Gosling segera memutuskan untuk merancang sendiri bahasa pemrograman komputer yang sederhana dan mampu menghindarkannya dari masalah-masalah tersebut dengan tetap memanfaatkan sintaks dasar dan sifat berorientasi objek dari C++.

Java (awalnya diberi nama Oak) sebagai bahasa pemrograman berbasis - *object-oriented* baru dan dapat diimplementasikan pada beragam peralatan dengan kapasitas memori terbatas. Oleh karenanya, Java dibekali dengan kemampuan untuk dapat dipergunakan pada beragam platform. Hal ini menjadikan aplikasi yang ditulis dengan Java dapat langsung dijalankan pada platform yang mendukung Java tanpa perlu mengkompilasi ulang kode sumber Java tersebut. Itulah mengapa dalam pemrograman Java dikenal istilah "*write once, run everywhere*", sekali menuliskan kode program, dapat dijalankan di bawah platform manapun.

Java menurut definisi dari Sun Microsystem adalah sekumpulan teknologi untuk membuat perangkat lunak dan menjalankannya pada komputer standalone ataupun pada lingkungan jaringan. Java sebagai sebuah platform, terdiri atas dua

bagian utama, yaitu: (a) *Java Virtual Machine* (JVM), dan *Java Application Programming Interface* (Java API). Java platform merupakan perangkat lunak yang menjadi mesin virtual bagi aplikasi Java untuk dieksekusi. Java berdiri di atas sebuah mesin interpreter yang diberi nama JVM. JVM inilah yang membaca file bytecode (ekstensi .class) dari suatu program sebagai representasi langsung program yang berisi bahasa mesin.

Kelebihan-kelebihana java menurut Wahana (2010) antara lain:

a. *Multiplatform*

Kelebihana utama dari Java ialah dapat dijalankan di beberapa platform/sistem operasi komputer. Program cukup ditulis dan dikompilasi sekali lalu hasilnya dapat dijalankan di atas beberapa *platform* tanpa perubahan. Kelebihan ini memungkinkan sebuah program berbasis Java dikerjakan di atas sistem operasi linux namun dijalan dengan baik pula di sistem operasi Microsoft Windows.

b. OOP (*Object Oriented Programming*)

OOP atau Pemrograman Berorientasi Objek artinya semua aspek yang terdapat di Java adalah objek. Java merupakan salah satu bahasa pemrograman berbasis objek secara murni. Semua tipe data diturunkan dari kelas dasar yang disebut *object*.

c. Perpustakaan Kelas yang Lengkap

Java terkenal dengan kelengkapan *library*/perpustakaan (kumpulan program yang disertakan dalam pemrograman java) yang sangat memudahkan dalam penggunaan oleh para pemrogram untuk membangun

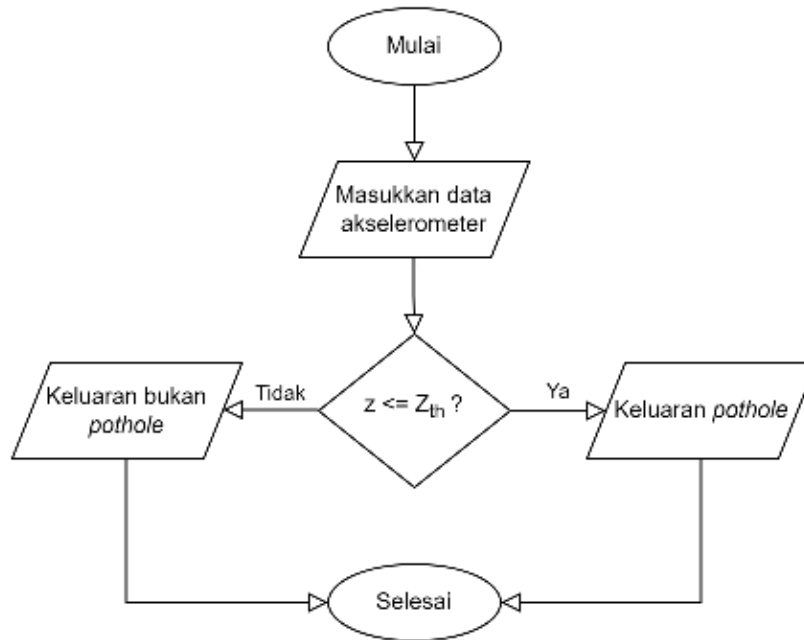
aplikasinya. Kelengkapan perpustakaan ini ditambah dengan keberadaan komunitas Java yang besar yang terus menerus membuat perpustakaan-perpustakaan baru untuk melingkupi seluruh kebutuhan pembangunan aplikasi.

## 2.8 Algoritma Z-Thresh

Algoritma *Z-Thresh* merupakan algoritma yang melakukan *thresholding* pada nilai amplitudo akselerometer di sumbu z. Metode ini mengklasifikasikan pengukuran berdasarkan nilai yang melebihi batas yang telah ditetapkan. Algoritma ini menggunakan informasi tentang posisi sumbu Z di akselerometer. *Event* direpresentasikan sebagai nilai yang melebihi ambang batas atau *threshold* yang ditentukan (Mednis et al., 2011). Masukan berupa data akselerometer pada sumbu x, y, dan z. Keluaran berupa informasi apakah terdeteksi lubang (*pothole*) atau bukan lubang (*pothole*).



Gambar 2. 16 Algoritma Z-Thresh



Gambar 2. 17 Flowchart Z-Threshold

## 2.9 Confusion Matrix

*Confusion matrix* merupakan salah satu metode yang dapat digunakan untuk mengukur kinerja suatu metode klasifikasi. Pada dasarnya *confusion matrix* mengandung informasi yang membandingkan hasil klasifikasi yang dilakukan oleh sistem dengan hasil klasifikasi yang seharusnya (E. Prasetyo, 2012).

Pada pengukuran kinerja menggunakan *confusion matrix*, terdapat 4 (empat) istilah sebagai representasi hasil proses klasifikasi. Keempat istilah tersebut adalah *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN). *True Positive* menunjukkan jumlah data positif yang terklasifikasi dengan benar oleh sistem. *True Negative* merupakan jumlah data negatif yang terklasifikasi dengan benar oleh sistem. *False Positive* menunjukkan jumlah data negatif namun terklasifikasi sebagai data positif oleh sistem. Sementara itu *False Negative*

merupakan jumlah data positif namun terklasifikasi sebagai data negatif oleh sistem.

Berdasarkan nilai *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN), dapat diperoleh nilai akurasi, presisi, dan *recall*. Nilai akurasi merupakan perbandingan antara data yang terklasifikasi benar dengan keseluruhan data. Nilai presisi menggambarkan jumlah data kategori positif yang diklasifikasikan secara benar dibagi dengan total data yang diklasifikasi positif. Presisi dapat diperoleh dengan Persamaan (2). *Recall* menunjukkan berapa persen data kategori positif yang terklasifikasi dengan benar oleh sistem. Nilai *recall* diperoleh dengan Persamaan (3).

Akurasi sangat bagus digunakan sebagai acuan performansi algoritma jika dataset memiliki jumlah data *False Negative* (FN) dan *False Positive* (FP) yang sangat mendekati. Namun jika jumlahnya tidak mendekati, maka digunakan *F-Measure* (*F1 Score*) sebagai acuan. Nilai *F-Measure* dapat diperoleh dengan Persamaan (4).

$$Presisi = \frac{TP}{FP + TP} \times 100\% \quad (2)$$

$$Recall = \frac{TP}{FN + TP} \times 100\% \quad (3)$$

$$F - Measure = 2 \times \frac{Presisi \times Recall}{Presisi + Recall} \quad (4)$$