

SKRIPSI

**PENGEMBANGAN APLIKASI PEMANTAUAN GETARAN
GEDUNG MENGGUNAKAN WSN**



Andrianto Chandra

NPM: 2016730017

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI DAN SAINS
UNIVERSITAS KATOLIK PARAHYANGAN
2020**

UNDERGRADUATE THESIS

**DEVELOPMENT OF BUILDING VIBRATION MONITORING
APPLICATION USING WSN**



Andrianto Chandra

NPM: 2016730017

**DEPARTMENT OF INFORMATICS
FACULTY OF INFORMATION TECHNOLOGY AND SCIENCES
PARAHYANGAN CATHOLIC UNIVERSITY
2020**

LEMBAR PENGESAHAN

PENGEMBANGAN APLIKASI PEMANTAUAN GETARAN GEDUNG MENGGUNAKAN WSN

Andrianto Chandra

NPM: 2016730017

Bandung, «tanggal» «bulan» 2020

Menyetujui,

Pembimbing Utama

Pembimbing Pendamping

Elisati Hulu, M.T.

«pembimbing pendamping/2»

Ketua Tim Penguji

Anggota Tim Penguji

«penguji 1»

«penguji 2»

Mengetahui,

Ketua Program Studi

Mariskha Tri Adithia, P.D.Eng

PERNYATAAN

Dengan ini saya yang bertandatangan di bawah ini menyatakan bahwa skripsi dengan judul:

PENGEMBANGAN APLIKASI PEMANTAUAN GETARAN GEDUNG MENGUNAKAN WSN

adalah benar-benar karya saya sendiri, dan saya tidak melakukan penjiplakan atau pengutipan dengan cara-cara yang tidak sesuai dengan etika keilmuan yang berlaku dalam masyarakat keilmuan.

Atas pernyataan ini, saya siap menanggung segala risiko dan sanksi yang dijatuhkan kepada saya, apabila di kemudian hari ditemukan adanya pelanggaran terhadap etika keilmuan dalam karya saya, atau jika ada tuntutan formal atau non-formal dari pihak lain berkaitan dengan keaslian karya saya ini.

Dinyatakan di Bandung,
Tanggal «tanggal» «bulan» 2020

Meterai Rp. 6000

Andrianto Chandra
NPM: 2016730017

ABSTRAK

«Tuliskan abstrak anda di sini, dalam bahasa Indonesia»

Kata-kata kunci: «Tuliskan di sini kata-kata kunci yang anda gunakan, dalam bahasa Indonesia»

ABSTRACT

«Tuliskan abstrak anda di sini, dalam bahasa Inggris»

Keywords: «Tuliskan di sini kata-kata kunci yang anda gunakan, dalam bahasa Inggris»

«kepada siapa anda mempersembahkan skripsi ini...?»

KATA PENGANTAR

«Tuliskan kata pengantar dari anda di sini ...»

Bandung, «bulan» 2020

Penulis

DAFTAR ISI

KATA PENGANTAR	xv
DAFTAR ISI	xvii
DAFTAR GAMBAR	xix
DAFTAR TABEL	xxi
1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan	2
1.4 Batasan Masalah	2
1.5 Metodologi	2
1.6 Sistematika Pembahasan	3
2 LANDASAN TEORI	5
2.1 Gedung	5
2.1.1 Klasifikasi Gedung	5
2.1.2 Pemantauan Kesehatan Struktural	6
2.2 Getaran	8
2.2.1 Jenis getaran	9
2.3 <i>Wireless Sensor Network</i>	9
2.3.1 Penerapan <i>Wireless Sensor Network</i> di berbagai bidang	9
2.3.2 Struktur Node Sensor	11
2.3.3 Arsitektur <i>Wireless Sensor Network</i>	12
2.3.4 Topologi <i>Wireless Sensor Network</i>	14
2.3.5 Protokol <i>Wireless Sensor Network</i>	17
2.4 Akselerometer	18
2.5 PreonVM	19
2.5.1 Fitur PreonVM	19
2.5.2 Kelebihan PreonVM	19
2.5.3 Class Library PreonVM	20
2.5.4 Preon32	21
2.5.5 Spesifikasi Sensor-Sensor Preon32	21
3 ANALISIS	23
3.1 Analisis Aplikasi Pemantauan Getaran Gedung Berbasis <i>Wireless Sensor Network</i>	23
3.1.1 Analisis Arsitektur dan Topologi <i>Wireless Sensor Network</i>	23
3.1.2 Analisis Akselerometer	24
3.1.3 Analisis Fungsi Aplikasi	24
3.1.4 Analisis Kelas	28
3.1.5 Analisis Paket/Pesan	31

DAFTAR GAMBAR

2.1	Ilustrasi penerapan <i>Wireless Sensor Network</i>	11
2.2	Struktur Node Sensor	11
2.3	Ilustrasi penerapan <i>Wireless Sensor Network</i>	12
2.4	Arsitektur <i>flat</i> pada <i>Wireless Sensor Network</i>	13
2.5	Arsitektur <i>single hop</i> pada <i>Wireless Sensor Network</i>	13
2.6	Arsitektur <i>multi hop</i> pada <i>Wireless Sensor Network</i>	14
2.7	Topologi <i>point-to-point</i>	14
2.8	Topologi <i>bus</i>	15
2.9	Topologi <i>tree</i>	15
2.10	Topologi <i>star</i>	16
2.11	Topologi <i>ring</i>	16
2.12	Topologi <i>partially connected mesh</i>	17
2.13	Topologi <i>fully conneted mesh</i>	17
2.14	Layer pada <i>Wireless Sensor Network</i>	17
2.15	<i>Virtual Machine</i> yang membuat aplikasi dapat dijalankan secara independen . . .	19
2.16	Preon32 Board	21
3.1	<i>Wireless Sensor Network</i> dengan Topologi <i>Star</i>	24
3.2	Arsitektur dan Topologi WSN	24
3.3	Diagram Use Case Aplikasi	26
3.4	Diagram Kelas NodeSensor	29
3.5	Diagram Kelas BaseStation	30
3.6	Diagram Kelas Tester	30

DAFTAR TABEL

2.1	Tabel Route Packages	20
2.2	Tabel Radio Packages	20
2.3	Tabel Other Virtenio Packages	20
2.4	Tabel Java Related Pakcages	21
3.1	Tabel Skenario Memeriksa node aktif	27
3.2	Tabel Skenario Memberi perintah sense ke sensor node	27
3.3	Tabel Skenario Stop Sensing	28
3.4	Tabel Skenario Keluar aplikasi	28

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Kesehatan sebuah bangunan/gedung merupakan salah satu persyaratan teknis yang harus ada saat mengurus IMB. IMB atau yang biasa dikenal dengan Izin Mendirikan Bangunan adalah sebuah perizinan yang diberikan oleh Kepala Daerah kepada pemilik bangunan untuk membangun baru, mengubah, memperluas, mengurangi, dan/atau merawat bangunan sesuai dengan persyaratan administratif dan persyaratan teknis yang berlaku. IBM ini sangat penting khususnya untuk bangunan atau gedung yang bertingkat seperti gedung-gedung yang ada di Universitas Katolik Parahyangan. Salah satu gedung yang ada di Universitas Katolik Parahyangan adalah gedung 10 yang termasuk juga area *rooftop*.

Salah satu parameter dalam pemantauan kesehatan sebuah gedung atau bangunan adalah getaran. Getaran yang terjadi pada sebuah bangunan dapat dipengaruhi oleh dua faktor. Faktor yang pertama adalah faktor dari bumi dan faktor kedua adalah faktor dari dalam gedung itu sendiri. Getaran merupakan salah satu faktor penyebab gempa bumi dimana terjadi pada kerak bumi sebagai gejala aktivitas tektonis maupun vulkanis. Pada umumnya getaran ini diakibatkan oleh adanya pergeseran lempeng pada permukaan bumi sehingga dapat terjadi gelombang gempa bumi. Getaran yang berasal dari gedung itu sendiri dapat dicontohkan dengan adanya mesin bertenaga besar yang terdapat dalam gedung tersebut seperti lift. Mesin dengan tenaga yang besar ini perlahan-lahan dapat menyebabkan sebuah gedung akan merasakan sebuah getaran yang lama kelamaan membuat gedung ini menjadi tidak stabil dan membuat kesehatan gedung menjadi memburuk.

Seiring berkembangnya teknologi, getaran pada suatu bangunan dapat dilihat untuk memberikan kemudahan dalam melakukan pengukuran data agar menjadi lebih efektif. Proses ini digunakan sebagai pengamatan, perekaman, dan pengevaluasian dalam hal ini adalah getaran pada sebuah bangunan atau gedung untuk menilai kesehatan secara berkelanjutan. Getaran pada sebuah gedung dapat ditangkap dengan menggunakan sensor *Accelerometer* yang disebar di sisi gedung yang akan membentuk sebuah jaringan sensor nirkabel atau istilah lainnya *Wireless Sensor Network*.

Wireless Sensor Network (WSN) adalah kumpulan sejumlah node yang diatur dalam sebuah jaringan kerjasama. Masing-masing node dalam jaringan sensor nirkabel biasanya dilengkapi dengan radio transceiver atau alat komunikasi wireless lainnya, mikrokontroler kecil, dan sumber energy seperti baterai. Banyak aplikasi yang bisa dilakukan menggunakan jaringan sensor nirkabel, misalnya pengumpulan data kondisi lingkungan, *security monitoring*, dan *node tracking scenarios*. Menggunakan *Wireless Sensor Network* (WSN) dapat menjadi alternatif untuk melakukan pemantauan getaran sebuah gedung. Dengan adanya *Wireless Sensor Network*, pemantauan getaran pada sebuah gedung dapat dilakukan dengan lebih mudah sehingga dapat memungkinkan pengguna mendapatkan informasi seperti amplitudo secara *daily* atau *real time* dengan tingkat akurasi yang tinggi.

Pada skripsi ini, akan dibuat sebuah perangkat lunak yang dapat menampilkan hasil pantauan getaran sebuah gedung yang ditampilkan dalam bentuk *graph* dengan menggunakan *Wireless Sensor Network* (WSN). Dengan menggunakan perangkat lunak tersebut, dapat diketahui seberapa besar getaran yang terjadi pada sebuah gedung sehingga dapat mengetahui besar kecilnya getaran yang

dihasilkan tersebut.

1.2 Rumusan Masalah

Masalah-masalah yang ingin diselesaikan dalam skripsi ini adalah sebagai berikut:

- Bagaimana sensor *Accelerator* bekerja?
- Bagaimana *Wireless Sensor Network* bekerja?
- Bagaimana komunikasi *wireless* 802.15.14 (*Zigbee Standard*) bekerja?
- Bagaimana membangun aplikasi pemantauan getaran gedung dengan menggunakan jaringan *wireless* sensor?

1.3 Tujuan

Tujuan-tujuan dari pembuatan perangkat lunak adalah sebagai berikut:

- Mempelajari cara kerja sensor *Accelerator*.
- Mempelajari cara kerja *Wireless Sensor Network*.
- Mempelajari cara kerja *wireless* 802.15.14 (*Zigbee Standard*).
- Membangun aplikasi pemantauan getaran gedung menggunakan *Wireless Sensor Network* (WSN).

1.4 Batasan Masalah

Penelitian ini dibuat berdasarkan batasan-batasan sebagai berikut:

- Sensor yang digunakan sebagai penelitian hanya sensor untuk mengukur getaran.
- Sensor hanya digunakan untuk mengukur getaran pada gedung-gedung yang ada di Universitas Katolik Parahyangan.
- Fokus utama penelitian ini adalah membangun aplikasi yang menangkap hasil *sensing* pada sensor *Accelerometer*

1.5 Metodologi

Berikut adalah metode penelitian yang digunakan dalam penelitian ini:

- Melakukan studi literatur mengenai *Wireless Sensor Network*.
- Mempelajari cara kerja sensor *Accelerometer*.
- Mempelajari sistem *Structural Health Monitoring* (SHM).
- Mempelajari pemograman pada *Wireless Sensor Network* dengan Bahasa Pemograman JAVA.
- Membangun infrastruktur jaringan *Wireless Sensor Network* (WSN).
- Mengimplementasi kode program pada sensor *Accelerator*.
- Melakukan pengujian sensor *Accelerator* yang telah diimplementasikan ke gedung yang ada di Universitas Katolik Parahyangan

1.6 Sistematika Pembahasan

Setiap bab dalam penelitian ini memiliki sistematika penulisan yang dijelaskan sebagai berikut:

Bab 1 Pendahuluan, yaitu membahas mengenai gambaran umum penelitian yang dilakukan ini. Berisi tentang latar belakang, rumusan masalah, tujuan, batasan masalah, metode penelitian, dan sistematika pembahasan.

Bab 2 Dasar Teori, yaitu membahas teori-teori yang mendukung berjalannya penelitian ini.

Bab 3 Analisis, yaitu membahas mengenai analisis masalah.

Bab 4 Perancangan, yaitu membahas

Bab 5 Implementasi dan Pengujian, yaitu membahas implementasi dari hasil rancangan dan pengujian dari aplikasi WSN yang telah dibuat dan pengujian aplikasi yang telah dibuat.

Bab 6 Kesimpulan, yaitu membahas kesimpulan dari hasil pengujian dan saran untuk pengembangan selanjutnya.

BAB 2

LANDASAN TEORI

2.1 Gedung

Berdasarkan KBBI(Kamus Besar Bahasa Indonesia), gedung dapat diartikan sebagai bangunan tembok dan sebagainya yang berukuran besar sebagai tempat kegiatan, seperti perkantoran, pertemuan, perniagaan, pertunjukan, olahraga, dan sebagainya. Secara umum, gedung didefinisikan sebagai sebuah struktur buatan manusia yang terdiri atas dinding dan atap yang didirikan secara permanen di suatu tempat. Gedung juga memiliki beragam bentuk, ukuran, dan fungsi, serta telah mengalami penyesuaian sepanjang sejarah yang disebabkan oleh beberapa faktor, seperti bahan bangunan, kondisi cuaca, harga, kondisi tanah, dan alasan estetika.

2.1.1 Klasifikasi Gedung

- Berdasarkan tingkat kompleksitas
 - Gedung Sederhana

Bangunan sederhana merupakan bangunan gedung dengan karakter sederhana, serta memiliki kompleksitas dan teknologi yang sederhana. Bangunan gedung sederhana meliputi gedung kantor dengan jumlah s.d. 2 lantai dengan luas maksimal mencapai 500m².
 - Gedung Tidak Sederhana

Bangunan gedung tidak sederhana merupakan bangunan gedung yang memiliki karakter, kompleksitas dan teknologi yang tidak sederhana. Bangunan gedung tidak sederhana. Bangunan gedung tidak sederhana meliputi gedung kantor bertingkat lebih dari 2 lantai yang memiliki luas di atas 500m².
 - Gedung Khusus

Bangunan gedung khusus merupakan bangunan gedung yang digunakan untuk kepentingan khusus, yang mempunyai tingkat kerahasiaan tinggi atau yang penyelenggaraannya dapat membahayakan lingkungan sekitar. Bangunan gedung ini memiliki kompleksitas tertentu, oleh karena itu dalam pembangunan atau pemanfaatannya membutuhkan pengelolaan dan persyaratan khusus. Bangunan gedung khusus meliputi gedung istana negara, gedung laboratorium, bangunan gedung reaktor nuklir, instalasi pertahanan dan keamanan, dan bangunan gedung sejenisnya yang ditetapkan oleh menteri.
- Berdasarkan Fungsinya
 - Gedung Rumah Tinggal

Pembuatan gedung rumah tinggal bertujuan untuk memenuhi kebutuhan manusia akan tempat tinggal. Pembuatan rumah tinggal ini harus memperhatikan faktor keamanan dan kenyamanannya. Beberapa contoh gedung rumah tinggal adalah rumah, rumah susun, asrama, mess, kontrakan dan apartemen.

- Gedung Komersial
Gedung komersial didirikan untuk mendukung aktivitas komersial meliputi jual, beli, dan sewa. Gedung komersial ditujukan untuk keperluan bisnis sehingga faktor lokasi yang strategis memegang peranan penting bagi kesuksesan bangunan tersebut. Beberapa contoh gedung komersial di antaranya pasar, *supermarket*, *mall*, *retail*, pertokoan, perkantoran dan kompleks kios.
- Gedung Fasilitas Penginapan
Gedung penginapan tercipta dari kebiasaan manusia yang sering melakukan aktivitas dengan berpindah-pindah tempat secara mobilitas. Keberadaan gedung ini memungkinkan seseorang bisa menyewa gedung untuk sementara waktu dengan keperluan menginap. Beberapa contoh gedung penginapan yaitu *motel*, *hotel*, *cottage* dan wisma tamu.
- Gedung Fasilitas Pendidikan
Gedung pendidikan didirikan untuk mendukung proses belajar dan mendapatkan ilmu dan pengetahuan yang baru. Beberapa contoh gedung fasilitas pendidikan adalah sekolah, universitas, perpustakaan dan gedung.
- Gedung Fasilitas Kesehatan
Gedung kesehatan didirikan untuk membantu masyarakat untuk melakukan berobat agar dapat sembuh dari sakit dan dapat menjalankan aktivitasnya. Beberapa contoh dari gedung fasilitas kesehatan adalah rumah sakit, puskesmas, apotek dan pusat rehabilitasi.
- Gedung Fasilitas Peribadatan
Gedung ibadah didirikan untuk memenuhi kebutuhan rohani manusia sebagai makhluk hidup yang memiliki Tuhan. Gedung peribadatan biasanya digunakan sebagai tempat beribadah dan upacara keagamaan. Beberapa contoh gedung fasilitas peribadatan adalah vihara, gereja, kelenteng, masjid dan pura.
- Gedung Fasilitas Transportasi
Gedung transportasi didirikan untuk sebagai pusat dari alat transportasi tertentu. Misalnya terminal untuk tempat berhentinya bis, pelabuhan sebagai tempat menepinya kapal, stasiun untuk pemberhentian kereta api, dan bandara sebagai tempat mendaratnya pesawat. Di gedung fasilitas transportasi ini juga umumnya dilengkapi dengan fasilitas-fasilitas layanan yang menunjang alat transportasi tersebut.
- Gedung Fasilitas Budaya dan Hiburan
Gedung budaya merupakan gedung yang dipakai untuk melestarikan dan atau memper-tunjukkan suatu kebudayaan. Sedangkan gedung hiburan adalah gedung yang dipakai sebagai tempat menciptakan hal-hal yang menghibur. Pada gedung, hubungan antara faktor budaya dan faktor hiburan ini saling merekat dan mendukung satu sama lain. Sebagai contoh gedung pertunjukan yang menampilkan drama sarat budaya yang dapat menghibur penonton. Begitu juga dengan bioskop dan museum.
- Gedung Fasilitas Pemerintah dan Layanan Publik
Gedung pemerintahan adalah gedung yang digunakan oleh pemerintah untuk menunaikan tugas dan kewajibannya. Di samping itu, gedung pemerintah ini juga dipakai sebagai gedung layanan publik misalnya dalam pengurusan data kependudukan, berkas-berkas resmi, surat perijinan, laporan pengaduan, dan lain-lain. Itu sebabnya, pembuatan gedung ini harus dirancang sedemikian rupa agar dapat mendukung kegiatan-kegiatan tersebut. Adapun contoh-contoh bangunan pemerintahan dan layanan publik yaitu kantor polisi, kantor perizinan, kantor dinas, dan balai pemerintahan.

2.1.2 Pemantauan Kesehatan Struktural

Pemantauan kesehatan struktural atau biasa disebut Structural Health Monitoring (SHM) merupakan sebuah proses penerapan deteksi kerusakan dan karakterisasi untuk struktur teknik seperti

jembatan dan bangunan. Kerusakan dalam hal ini diartikan sebagai perubahan pada material atau sifat-sifat geometris dari suatu sistem struktural yang secara negatif mempengaruhi kinerja sistem. Proses pemantauan kesehatan struktural melibatkan pengamatan suatu sistem dari waktu ke waktu menggunakan pengukuran respons sampel secara berkala dari berbagai sensor (sering yang digunakan adalah akselerometer), ekstraksi fitur yang peka terhadap kerusakan dari pengukuran ini, dan melakukan analisis statistik fitur untuk menentukan keadaan saat ini kesehatan sistem.

Salah satu contoh struktural adalah gedung. Gedung harus diukur kesehatannya untuk melihat apakah suatu gedung layak digunakan atau tidak. Gedung yang kesehatannya baik menjadi syarat untuk mendapatkan dokumen IMB (Izin Mendirikan Bangunan). Dokumen IMB merupakan dokumen dimana sebuah gedung yang akan didirikan mendapatkan izin dari pemerintah. Sebuah gedung dapat diukur kesehatannya dengan melakukan pemantauan atau *monitoring* gedung. Pemantauan kesehatan gedung adalah sebuah proses pemantauan informasi kondisi dan keamanan dari sebuah gedung. Tujuan dari pemantauan kesehatan gedung ini adalah:

- Memantau secara terus-menerus kondisi kesehatan gedung
- Mengetahui sesegera mungkin gejala-gejala tidak normal yang mungkin dapat terjadi pada sebuah gedung
- Mencatat perilaku-perilaku beban yang dikirim oleh gedung
- Sebagai sumber data untuk menganalisis dalam pengambilan keputusan dalam tindakan mencegah atau perawatan pada sebuah gedung

Faktor-Faktor Kesehatan Gedung

Kesehatan sebuah gedung dapat dilihat dari beberapa faktor. Faktor-faktor yang mempengaruhi kesehatan gedung antara lain:

- Faktor Suhu

Suhu merupakan salah satu faktor alam yang berpengaruh kepada kesehatan sebuah gedung. Suhu yang ekstrim dan terjadi secara terus-menerus menyebabkan kesehatan gedung menjadi rusak terutama dibagian luar gedung tersebut. Beberapa contoh komponen yang harus dilindungi karena pengaruh suhu adalah lapisan water proofing diatas atap plat beton, cat pada listplank kayu, serta cat eksterior yang sering terkena panas ataupun dingin secara terus menerus

- Faktor Air Hujan

Faktor air hujan menjadi salah satu faktor yang berpengaruh kepada kesehatan suatu gedung. Air hujan dapat membuat kerusakan pada gedung dan kasus yang sering terjadi akibatnya adalah kebocoran pada gedung.

- Faktor Angin

Faktor angin merupakan salah satu alam yang berpengaruh terhadap kesehatan gedung. Faktor angin dapat membuat gedung mengalami kerusakan dan salah satu komponen gedung yang sering terkena akibat angin adalah penutup atap genteng. Pada serangan angin yang kencang, menimbulkan gerakan-gerakan pada atap yang menyebabkan atap mudah bergeser satu sama lain sehingga mudah lepas jika terjadi angin kencang.

- Faktor Getaran

Salah satu faktor yang dapat digunakan untuk pemantauan kesehatan gedung adalah getaran. Getaran merupakan salah satu faktor penting dalam kesehatan gedung. Getaran dalam sebuah gedung dapat berasal dari alat-alat mesin yang sedang berjalan, adanya sebuah kerusakan

dalam gedung tersebut ataupun adanya sebuah pergeseran lempengan bumi atau biasa disebut gempa bumi. Getaran-getaran yang dihasilkan gedung seringkali tidak dapat dirasakan oleh manusia sehingga dibutuhkan sensor untuk mendeteksi getaran tersebut. Gedung akan dinyatakan baik dan dapat digunakan apabila getaran yang dihasilkan oleh sebuah gedung semakin kecil.

- **Faktor Petir**

Faktor petir merupakan faktor kesehatan gedung yang jarang, tetapi kerusakan yang terjadi pada gedung akibat petir tidak dapat dianggap sepele karena kerusakan yang dihasilkan oleh petir dapat berakibat fatal diantaranya listrik pada gedung mati, jaringan telepon ataupun internet di gedung juga dapat mati.

- **Faktor Hama**

Faktor hama merupakan faktor yang dapat mempengaruhi kesehatan gedung. Bedanya untuk faktor hama, gedung yang dapat mengalami kerusakan akibat faktor hama ini adalah gedung yang terbuat dari kayu. Salah satu contoh penyebab kerusakan akibat gedung dari faktor hama adalah rayap.

Sifat Kerusakan Pada Gedung

Sifat kerusakan yang terjadi pada gedung dapat ditinjau dari pengaruh komponen tersebut hingga akibat dari kerusakan komponen sifat tersebut. Sifat-sifat kerusakan gedung dibagi menjadi tiga bagian yaitu:

- *Emergency*

Kerusakan yang memiliki pengaruh sangat tinggi terhadap aktivitas penghuni pada gedung dan mempengaruhi komponen lain dalam gedung tersebut. Contoh kerusakannya adalah kerusakan kran air, atap bocor, instalasi listrik.

- *Urgent*

Kerusakan yang memiliki pengaruh tinggi terhadap aktivitas penghuni dan kerusakan komponen lainnya pada gedung. Contoh kerusakannya adalah kerusakan pada keramik yang sering dilalui, jalan berlubang.

- *Normal*

Kerusakan kecil yang menyebabkan fungsi kurang sempurna atau penurunan tampak pada komponen yang mempunyai pengaruh kecil pada aktivitas penghuni. Contoh kerusakannya adalah cat dinding yang udah rusak.

2.2 Getaran

Getaran adalah suatu peristiwa gerak bolak balik secara teratur suatu benda melalui suatu titik seimbang terhadap suatu titik acuan. Keseimbangan yang dimaksud adalah suatu keadaan dimana suatu benda berada pada posisi diam jika tidak ada gaya yang bekerja pada benda tersebut. Besar kecilnya suatu getaran yang dihasilkan oleh suatu benda dipengaruhi oleh jumlah energi yang diberikan. Semakin besar energi yang diberikan maka semakin kuat getaran yang terjadi. Satu getaran sama dengan satu kali gerakan bolak balik penuh dari benda tersebut.¹

¹<https://www.gurupendidikan.co.id/getaran/>

2.2.1 Jenis getaran

Secara umum getaran dapat diklasifikasikan menjadi beberapa jenis yaitu:

- **Getaran Bebas dan Paksa**

Jika sebuah sistem diberi inisial gangguan, sehingga akan terjadi getaran dengan sendirinya, maka getaran tersebut dinamakan **getaran bebas**. Tidak ada gaya eksternal bekerja pada sistem. Contoh dari getaran bebas adalah gerakan bolak-balik yang ada pada sebuah pendulum. Sedangkan jika sebuah sistem diberikan suatu gaya dari luar yang secara berkala (berulang-ulang) maka getaran yang dihasilkan pada sistem disebut dengan **getaran paksa**. Salah satu contoh getaran paksa adalah getaran yang dihasilkan oleh mesin yang sedang bekerja. Apabila frekuensi suatu gaya eksternal sama dengan frekuensi gaya sistem, maka akan menimbulkan resonansi. Resonansi ini dapat membahayakan suatu sistem dan menyebabkan kerusakan struktur dari bangunan.

- **Getaran Teredam dan Tidak Teredam**

Getaran tidak teredam adalah getaran dimana jika tidak adanya energi dalam sebuah getaran yang hilang atau terdisipasi akibat adanya getaran atau hambatan lainnya. Sedangkan yang dimaksud dengan **getaran teredam** adalah sebuah getaran dimana mengalami pengurangan energi secara bertahap.

- **Getaran Mekanis dan Nonmekanis**

Getaran mekanis adalah getaran suatu benda yang getarannya mengalami suatu pergeseran linear atau pergeseran sudut. Contoh getaran mekanis adalah getaran senar gitar pada saat dipetik, getaran pada bandul dan getaran atom pada zat padat. Sedangkan **getaran nonmekanis** adalah suatu gerakan yang juga melibatkan adanya perubahan pada besaran-besaran fisika. Contoh dari getaran nonmekanis adalah medan listrik serta medan magnet.

- **Getaran Deterministik dan Acak**

Getaran deterministik adalah sebuah getaran dimana besarnya eksitasi (gaya atau gerakan) yang bekerja pada suatu sistem getaran diketahui pada waktu tertentu. **Getaran acak** adalah getaran dimana getaran atau gaya yang bekerja pada suatu sistem dihasilkan secara acak pada waktu tertentu. Contoh dari getaran acak adalah kecepatan angin, gerakan tanah selama gempa bumi.

2.3 *Wireless Sensor Network*

Wireless Sensor Network (WSN) merupakan jaringan nirkabel yang terdiri dari sekumpulan node sensor yang saling terhubung yang diletakkan pada suatu tempat dan memiliki kemampuan untuk mengukur kondisi lingkungan sekitar (*sensing*), melakukan komputasi dan dilengkapi dengan alat komunikasi *wireless* untuk komunikasi antara node sensor. *Wireless Sensor Network* juga berguna untuk memantau hasil *sensing* kondisi suatu lingkungan yang dilakukan oleh sensor. Selain itu WSN juga dapat mengatur data yang akan dikirimkan dan dikumpulkan di *base station* atau juga dapat diteruskan ke node sensor tetangganya hingga sampai ke *base station* sebagai pusat menggunakan *radio transceiver* untuk dilakukan pengelolaan data (Firdaus, 2014). Bentuk radio transceiver untuk setiap node menggunakan antena internal ataupun antena eksternal.

2.3.1 Penerapan *Wireless Sensor Network* di berbagai bidang

Wireless Sensor Network pada awalnya hanya digunakan untuk melakukan ilmu komputasi saja. Semakin lama dan majunya teknologi, *Wireless Sensor Network* telah digunakan untuk melakukan

pemantauan ataupun pengukuran seperti di bidang militer, *Wireless Sensor Network* digunakan untuk mendeteksi musuh yang ada dilautan ataupun darat. Kemudian pemanfaatannya dikembangkan untuk membantu berbagai bidang kegiatan manusia dalam kehidupan. Penerapan *Wireless Sensor Network* untuk kehidupan manusia dapat dilihat pada contoh ilustrasi (Gambar 2.1)². Berikut adalah beberapa penerapan *Wireless Sensor Network*:

- Bidang Pemantauan Lingkungan

Wireless Sensor Network dibidang pemantauan lingkungan memiliki banyak pengaplikasian seperti untuk memantau polusi udara, mendeteksi kebakaran hutan, mendeteksi adanya bencana alam khususnya gempa, dan lain-lain.

- Bidang Kesehatan

Wireless Sensor Network dapat digunakan pada aplikasi kesehatan seperti mendeteksi kondisi manusia yang memiliki kekurangan seperti disabilitas, monitoring penggunaan obat, serta bisa juga untuk mendiagnosis sebuah penyakit.

- Bidang Transportasi

Pada bidang transportasi, *Wireless Sensor Network* digunakan untuk mendeteksi arus lalu lintas secara aktual yang akan dikirimkan kepada pengendara. Selain itu, juga bisa digunakan untuk mendeteksi kecepatan kendaraan yang melaju apakah melewati batas kecepatan atau tidak.

- Bidang Militer

Pemanfaatan *Wireless Sensor Network* di bidang militer adalah Wide Area Tracking System (WATS). WATS merupakan prototipe jaringan yang menggunakan teknologi *Wireless Sensor Network* untuk mendeteksi ancaman nuklir. Pemanfaatan lainnya untuk bidang militer dapat juga digunakan untuk mendeteksi serangan dari musuh.

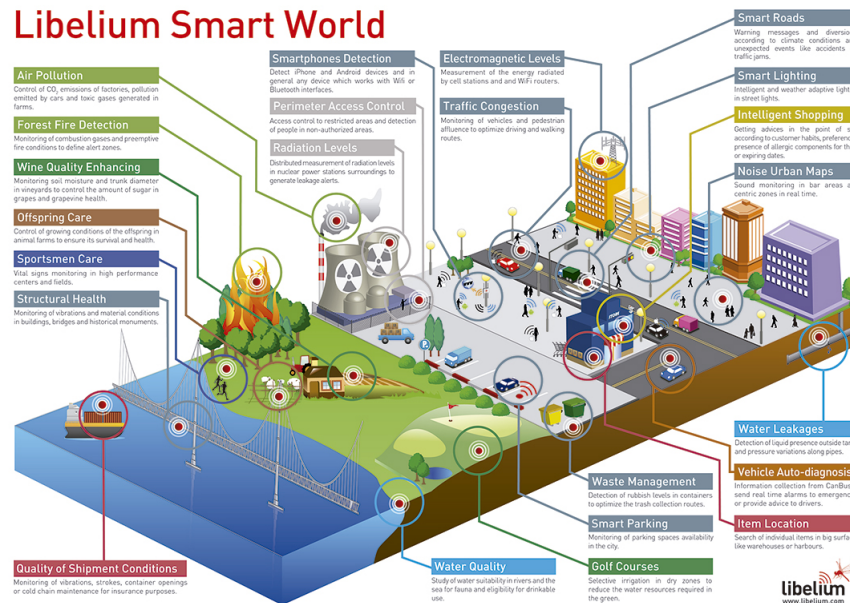
- Bidang Pertanian

Pada bidang pertanian, *Wireless Sensor Network* dapat digunakan untuk membantu pengelola pertanian untuk penggunaan air, kelembaban tanah yang digunakan, pH tanah yang digunakan untuk pertanian, serta mengelola pembuangan pertanian mereka.

- Bidang Infrastruktur

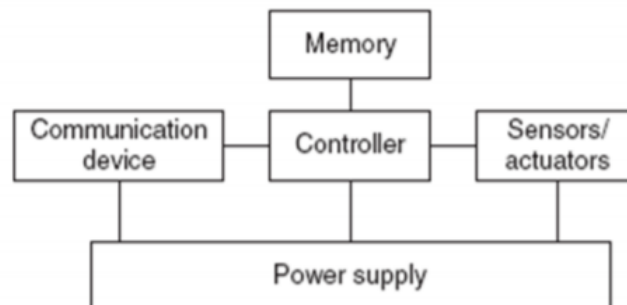
Wireless sensor network juga digunakan dalam pembangunan infrastruktur. Pemanfaatan *Wireless Sensor Network* biasa digunakan untuk melakukan pengamatan kondisi pembangunan infrastruktur, baik dari segi bangunan maupun geografis seperti mengukur getaran sebuah pembangunan. Penerapan WSN untuk membantu pembangunan infrastruktur dilakukan secara *daily* dan *real-time* untuk menjaga kondisi lingkungan selama proses pemabangunan selesai dilakukan.

²<http://eprints.polsri.ac.id/4497/3/File/20III.pdf>

Gambar 2.1: Ilustrasi penerapan *Wireless Sensor Network*

2.3.2 Struktur Node Sensor

Node adalah salah satu titik sambungan, titik redistribusi, atau titik akhir komunikasi³. Komponen utama node sensor antara lain *controller*, *transceiver*, *memory*, *power source*, dan sensor. (Gambar 2.2).



Gambar 2.2: Struktur Node Sensor

- **Controller**

Controller adalah inti utama yang ada pada node sensor. *Controller* juga bertindak sebagai pengatur fungsi dari komponen-komponen lain. *Controller* juga mengumpulkan data dari sensor lain dan memproses data. Pada *controller* terdapat *microcontroller* yang mengatur dan melakukan komputasi data. *Microcontroller* lebih sering menjadi alternatif karena mengurangi penggunaan energi dan adanya sleep states yang berarti hanya bagian dari *controller* saja yang aktif.

- **Communication Device**

Communication Device digunakan untuk menerima atau mengirim data antar node sensor. Communication Device membuat node sensor dapat terhubung dalam jaringan dan

³<http://gilang777.blogspot.com/2015/06/pengertian-node-jaringan.html>

berkomunikasi dengan node sensor lainnya.

- Sensor / Actuator

Sensor merupakan bagian yang digunakan untuk melakukan *sensing* atau pengukuran terhadap suatu keadaan lingkungan yang diamati. *Actuator* berfungsi sebagai pengubah sinyal dari lingkungan menjadi besaran-besaran fisik.

- Memory

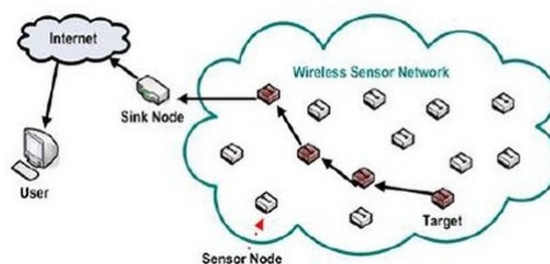
Random Access Memory (RAM) digunakan untuk menyimpan hasil sementara yang didapat dari *sensing* dari sensor. RAM ini memiliki sifat *volatile* yang berarti jika node sensor mati atau energi habis maka data-data yang ada pada RAM akan hilang.

- Power Supply

Power Supply digunakan sebagai sumber energi yang dibutuhkan untuk mengoperasikan fungsionalitas dari keseluruhan komponen node sensor lainnya. Penyediaan energi ini dapat memiliki dua jenis metode, yaitu *storing energy* dan *energy scavenging*. Metode *storing energy*, sensor node menggunakan baterai sebagai energi utamanya. Jenis baterai yang digunakan dapat yang diisi ulang maupun yang tidak dapat diisi ulang. Sedangkan metode *energy scavenging* digunakan saat membuat *Wireless Sensor Network* yang akan digunakan dalam waktu yang lama. Pada metode *energy scavenging*, node sensor menggunakan perubahan energi sebagai sumber energi utamanya. Energi yang didapat dengan metode *energy scavenging* didapatkan setelah mengonversi pemanfaatan energi alam seperti cahaya matahari, air atau angin untuk dijadikan energi listrik untuk menjalankan node sensor.

2.3.3 Arsitektur *Wireless Sensor Network*

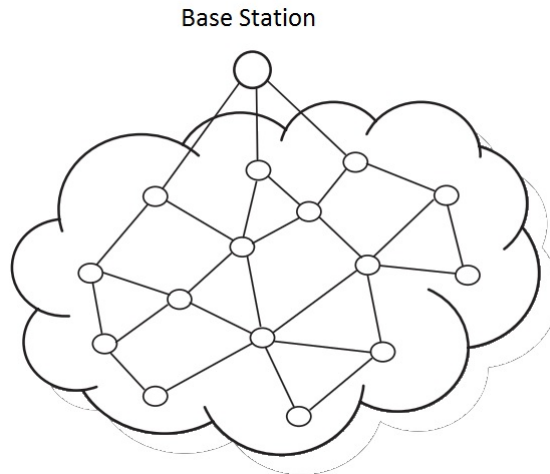
Pada *Wireless Sensor Network*, arsitektur yang biasanya dipakai adalah **arsitektur flat atau peer-to-peer** dan *arsitektur hierarki*. Perbedaan antara kedua jenis arsitektur adalah cara node sensor dalam berkomunikasi. Pada arsitektur *flat*, node yang tersebar dapat langsung mengirimkan data hasil sensing ke *base station*. Sedangkan pada arsitektur *hierarkikal*, node yang tersebar harus mengirimkan data ke *cluster head* terlebih dahulu, sebelum diteruskan ke *base station*.



Gambar 2.3: Ilustrasi penerapan *Wireless Sensor Network*

(i) **Arsitektur *Flat / Peer-to-Peer***

Pada arsitektur *flat*, setiap node sensor yang tersebar memiliki tugas dan peran yang sama dalam melakukan *sensing* dan mengirimkan hasilnya ke *base station*. Data hasil *sensing* ini langsung dikirimkan ke *base station* tanpa butuh sebuah perantara.

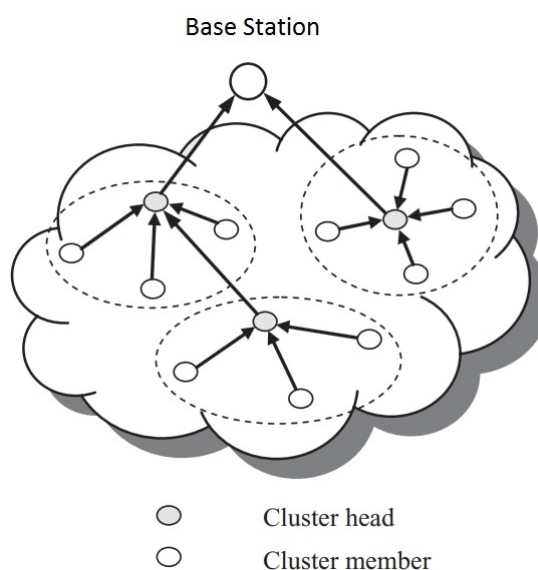
Gambar 2.4: Arsitektur *flat* pada *Wireless Sensor Network*

(ii) **Arsitektur Hierarki**

Pada arsitektur hierarki, setiap hasil *sensing* dari sensor tidak langsung dikirimkan ke *base station*. Melainkan setiap node sensor akan membentuk sebuah grup yang disebut dengan *cluster*. Tiap *cluster* terdiri dari sebuah *cluster head* dan *cluster member*. *Cluster head* bertindak sebagai penerima data hasil *sensing* dari *cluster member*, untuk diteruskan ke *base station*. Jenis arsitektur hierarki juga dapat dibedakan berdasarkan jarak antara *cluster head* dengan *cluster member*. Perbedaan dari jenis ini adalah jumlah *hop* yang dibutuhkan untuk mencapai tujuan yaitu *base station*.

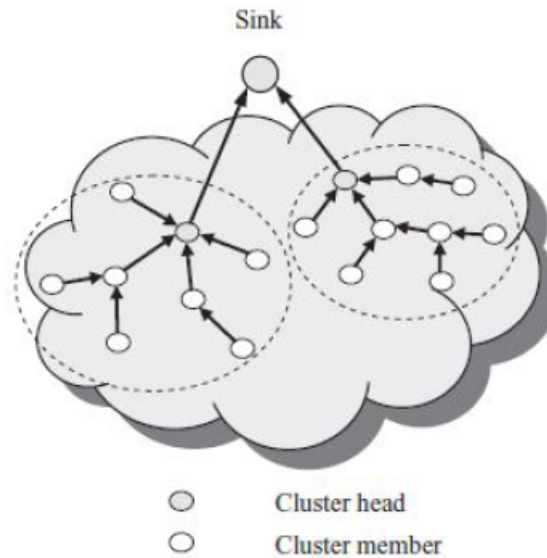
- *Single hop*

Pada arsitektur hierarki *single hop*, *cluster member* hanya membutuhkan satu lompatan untuk mencapai ke *cluster head*. Dapat dikatakan, data hasil *sensing* yang diterima oleh *cluster member* dapat langsung diterima oleh *cluster head* tanpa adanya perantara dan dapat diteruskan ke *base station*.

Gambar 2.5: Arsitektur *single hop* pada *Wireless Sensor Network*

- *Multi hop*

Pada arsitektur hierarki *multi hop*, *cluster member* membutuhkan lebih dari satu lompatan untuk mencapai *base station*. *Cluster member* akan mengirimkan data ke *cluster member* yang jaraknya lebih dekat dengan *cluster head* dan ini akan terus dilakukan sampai data yang dikirimkan sampai di *cluster head* kemudian akan diteruskan ke *base station*.



Gambar 2.6: Arsitektur *multi hop* pada *Wireless Sensor Network*

2.3.4 Topologi *Wireless Sensor Network*

Topologi pada *wireless sensor network* memiliki beberapa jenis. Tiap jenis topologi dibedakan berdasarkan tujuan, skala jaringan dan kondisi lingkungan. Beberapa jenis topologi pada *wireless sensor network* adalah topologi *point-to-point*, *bus*, *tree*, *star*, *ring* dan *mesh*.

- Topologi *Point-to-Point*

Topologi *point-to-point* adalah topologi yang menghubungkan dua titik (Gambar 2.7). Pada topologi ini dibagi menjadi dua yaitu *permanent point-to-point* dan *switched point-to-point*. *Permanent point-to-point* adalah koneksi antara dua titik dan bersifat tidak dapat diubah (permanen). Sedangkan *switched point-to-point* adalah koneksi *point-to-point* yang dapat dipindahkan antara node yang berbeda ⁴.

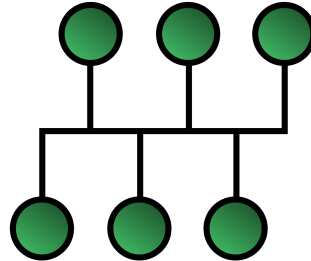


Gambar 2.7: Topologi *point-to-point*

- Topologi *Bus*

⁴<http://link.springer.com/chapter/10.1007/978-1-4302-6014-14.html>

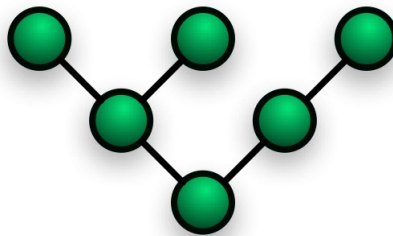
Topologi bus akan terdiri dari node-node yang terhubung pada sebuah jalur. Jalur ini digunakan oleh node-node untuk saling berkomunikasi (Gambar 2.8). Sifat komunikasi pada jalur ini adalah satu arah, dimana komunikasi node dilakukan secara bergantian. Topologi ini sederhana dan mudah untuk diimplementasikan. Kekurangan dari topologi ini adalah apabila jalur mengalami kerusakan maka setiap node tidak dapat melakukan komunikasi lagi.



Gambar 2.8: Topologi *bus*

- Topologi *Tree*

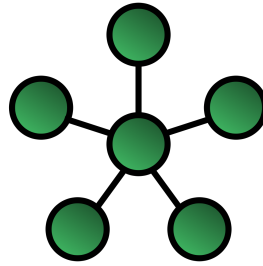
Pada topologi *tree*, node-node akan disusun secara hierarki dengan sebuah node yang berada pada level paling atas disebut sebagai *root node*. *Root node* akan terhubung dengan satu atau lebih node yang levelnya dibawah sehingga *root node* akan bertindak sebagai komunikasi utama (Gambar 2.9). Penggunaan topologi *tree* lebih mudah untuk melakukan identifikasi dan meminimalisir kesalahan. Kelemahan dari topologi ini adalah topologi ini akan semakin sulit dikonfigurasi seiring jika ukuran *tree* yang sangat besar dan banyaknya *level tree* yang ada.



Gambar 2.9: Topologi *tree*

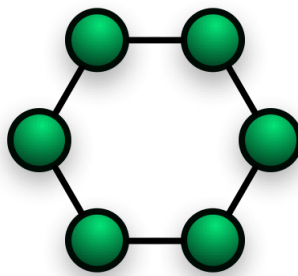
- Topologi *Star*

Topologi *star* memiliki sebuah node yang berada di tengah sebagai *hub* atau *switch* yang biasa disebut *central node* (Gambar 2.10). Setiap node-node akan terhubung dengan *central node* ini. *Central node* yang telah mendapatkan pesan dari node pengirim akan meneruskan pesannya ke node tujuan. Apabila *central node* mengalami kerusakan, maka tidak akan terjadinya komunikasi antar node karena segala komunikasi harus dilakukan melalui *central node*.

Gambar 2.10: Topologi *star*

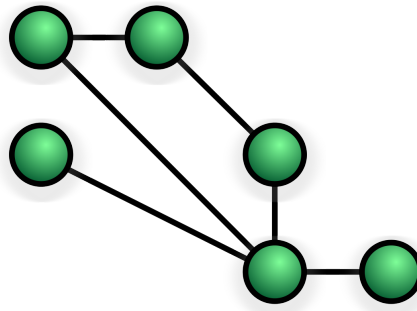
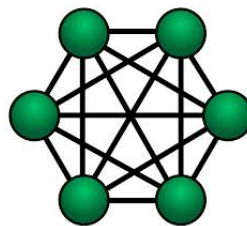
- Topologi *ring*

Jaringan topologi *ring* berbentuk rangkaian node yang saling terhubung dengan dua node terdekat lainnya sehingga akan berbentuk seperti lingkaran atau cincin (Gambar 2.11). Pada topologi *ring*, node akan mengirimkan pesan kemudian akan diteruskan ke node tetangganya sampai menemukan node tujuan yang akan dikirimkan pesan. Kekurangannya adalah salah satu node mati maka komunikasi jaringannya akan mati. Masalah ini dapat diatasi dengan membuat sebuah node tidak hanya dapat melakukan komunikasi satu arah melainkan ke arah sebaliknya.

Gambar 2.11: Topologi *ring*

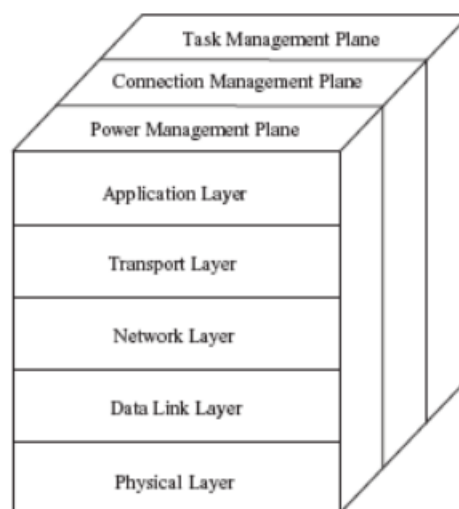
- Topologi *Mesh*

Topologi mesh dibagi menjadi dua jenis yaitu partially connected mesh dan fully connected mesh. Perbedaan keduanya adalah hubungan antar node, pada partially connected mesh, node dapat terhubung dengan satu atau lebih node lainnya (Gambar 2.12). Sedangkan fully connected mesh, setiap node harus terhubung dengan semua node-node lain yang ada pada sebuah jaringan (Gambar 2.13).

Gambar 2.12: Topologi *partially connected mesh*Gambar 2.13: Topologi *fully conneted mesh*

2.3.5 Protokol *Wireless Sensor Network*

Wireless Sensor Network memiliki lima layer protokol yaitu *physical layer*, *data link layer*, *network layer*, *transport layer* dan *application layer* (Gambar 2.14). Selain itu, protokol *Wireless Sensor Network* juga dibagi menjadi 3 grup manajemen yaitu *power management plane*, *connection management plane* dan *task management plane*. *Power management plane* bertanggung jawab untuk mengatur tingkat kekuatan sebuah sensor untuk melakukan *sensing*, memproses data dan melakukan komunikasi. *Connection management plane* bertanggung jawab untuk melakukan konfigurasi terhadap node sensor yang terkait dengan koneksi antar node sensor. *Task management plane* bertanggung jawab untuk pembagian tugas diantara node sensor dalam melakukan *sensing* agar energi yang digunakan efektif dan efisien.

Gambar 2.14: Layer pada *Wireless Sensor Network*

- *Physical layer*

Physical layer bertanggung jawab untuk mengubah *bit stream* dari *data link layer* menjadi sinyal agar bisa melakukan transmisi melalui media komunikasi yang tersedia. Pengubahan ini dilakukan agar transmisi dapat dilakukan melalui *transceiver*. Salah satu cara yang bisa digunakan adalah menggunakan *Radio Frequency* (RF). *Radio Frequency* sering dipakai karena biaya yang dikeluarkan murah dan ukuran perangkat yang kecil.

- *Data Link layer*

Data Link layer bertanggung jawab untuk melakukan *multiplexing* pada aliran data, membentuk *data frame*, mendeteksi *data frame*, *medium access*, and *error control*. *Data Link layer* juga memiliki proses yang penting yaitu *Medium Access Control* (MAC). Protokol MAC ini bertindak untuk melakukan kontrol terhadap akses media yang dilakukan oleh node sensor dan mencegah terjadinya adanya paket yang bertabrakan.

- *Network layer*

Network layer bertanggung jawab untuk melakukan *routing* dari node sensor ke *sink node*. *Network layer* juga merupakan lapisan yang menyediakan jalur komunikasi jaringan sehingga *network layer* bertugas untuk menentukan jenis komunikasi antar node yang digunakan apakah jenis yang digunakan *single hop* atau *multi hop*.

- *Transport layer*

Transport layer bertanggung jawab untuk pengiriman data yang *reliable* antar node sensor ke node sensor lainnya atau ke *sink node*. Pengiriman data pada *Wireless Sensor Network* terbagi menjadi dua jenis, yaitu ***upstream*** dan ***downstream***. Kedua jenis ini dibedakan berdasarkan dari asal pengirim dan penerima data. ***Upstream*** merupakan node sensor mengirimkan data *sensing* ke *sink node*. Sedangkan ***downstream***, *sink node* akan mengirimkan perintah-perintah ataupun *query* ke setiap node sensor yang ada. Setiap jenis pengiriman, kebutuhan *reliability* nya berbeda-beda. Pada *upstream*, *reliable* data dapat ditoleransi karena node sensor mengirimkan data ke sink node secara berulang-ulang sehingga data yang hilang dapat dikoreksi. Pada *downstream*, tidak dapat ditoleransi karena jika data yang dikirimkan tidak *reliable*, maka aplikasi tidak dapat dijalankan.

- *Application layer*

Application layer bertanggung jawab untuk manajemen lalu lintas dan menyediakan antar muka perangkat lunak untuk berbagai aplikasi yang menerjemahkan data dalam bentuk yang dapat dimengerti atau mengirim *query* untuk mendapatkan informasi tertentu.

2.4 Akselerometer

Salah satu sensor yang terdapat pada node sensor adalah sensor akselerometer. Akselerometer adalah alat yang mengukur *linear acceleration*. Accelerometer memiliki kemampuan untuk mengukur akselerasi, kemiringan, dan getaran statis atau dinamis pada 1 sumbu saja $x/y/z$ yang biasa disebut *single axis* atau pada 2 sumbu contohnya (x,y) atau (x,z) atau (y,z) yang biasa disebut dengan *two axis* ataupun 3 sumbu yaitu (x,y,z) yang disebut *three axis*. Sensor Akselerometer dibagi menjadi 2 jenis yaitu:

- *Absolute accelerometer*

Akselerometer ini terpasang langsung pada objek yang akan diukur.

- *Relative accelerometer*

Akselerometer ini mengukur jarak antara objek yang diukur dan titik acuan yang stabil atau bergerak dengan konstan. Akselerometer ini sering digunakan untuk mengukur getaran dari jarak tertentu.

2.5 PreonVM

PreonVM adalah *virtual machine* (VM) yang berasal dari VIRTENIO untuk digunakan dalam sistem *embedded* dengan sumber daya yang sangat rendah⁵. *Virtual machine* PreonVM ini sudah sangat optimal dan tidak memerlukan sistem operasi tambahan dan berjalan langsung pada mikrokontroler. Dengan PreonVM, *developer* dapat membuat aplikasi sensor dengan mudah menggunakan bahasa Pemrograman Java yang mengumpulkan data-data (hasil *sensing*) dari sensor. API pada PreonVM mendukung antarmuka radio sesuai dengan IEEE 802.15.4.

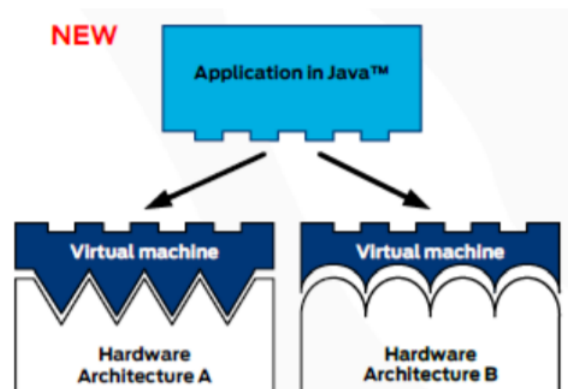
2.5.1 Fitur PreonVM

Fitur-fitur yang dimiliki oleh PreonVM sebagai berikut:

- Aplikasi dibangun dengan Bahasa Pemrograman Java
- Mendukung tipe data pada Java seperti *char*, *byte*, *int*, *long*, *float*, atau *double*
- Jumlah *thread* yang tidak terbatas
- *System properties* untuk konfigurasi aplikasi
- *Garbage collection* dengan *memory defragmentation*
- *Exception handling* (*try*, *catch*, *Exception*, atau *Runtime Exception*)

2.5.2 Kelebihan PreonVM

Kelebihan PreonVM adalah PreonVM menggunakan *object-oriented programming* menggunakan bahasa pemrograman Java pada *virtual machine* nya untuk *embed-system*. PreonVM juga dioptimasi agar aplikasi dapat dijalankan 8-bit sampai 32-bit *microcontroller* dengan 8KB RAM dan 128KB Flash minimum. *Virtual Machine* pada PreonVM dapat membuat aplikasi dapat berjalan secara sendiri pada arsitektur yang digunakan. Sehingga aplikasi Java yang dibuat dapat dijalankan pada arsitektur yang berbeda-beda tanpa harus dijalankan (Gambar 2.15).



Gambar 2.15: *Virtual Machine* yang membuat aplikasi dapat dijalankan secara independen

⁵<https://www.virtenio.com/en/portfolio-items/preonvm/>

2.5.3 Class Library PreonVM

PreonVM memiliki 2 *package* yaitu *package* dari Virtenio itu sendiri dan *package* dari Java. *Package* yang disediakan oleh Virtenio dibagi lagi menjadi beberapa bagian diantaranya adalah *Route Packages*, *Radio Packages* dan *Other Virtenio Packages*. Sedangkan *package* yang dari Java terdiri dari *Java Related Packages*. Berikut ada tabel-tabel *package* yang disediakan:⁶

Tabel 2.1: Tabel Route Packages

Package	Deskripsi
com.virtenio.route.aodv	Paket yang berisi kelas terkait AODV (Ad Hoc On Demand Vector) Routing

Tabel 2.2: Tabel Radio Packages

Package	Deskripsi
com.virtenio.radio	Paket yang berisi kelas terkait radio
ESRTcom.virtenio.radio.ieee_802_15_4	Paket yang berisi kelas terkait IEEE 802.15.4

Tabel 2.3: Tabel Other Virtenio Packages

Package	Deskripsi
com.virtenio.crypt	Paket yang berisi untuk enkripsi dan dekripsi
com.virtenio.driver	Paket yang berisi drivers untuk berbagai perangkat
com.virtenio.driver.adc	Paket yang berisi kelas ADC <i>driver</i>
com.virtenio.driver.atmodem	Paket yang berisi kelas ATModem <i>driver</i>
com.virtenio.driver.button	Paket yang berisi kelas <i>button driver</i>
com.virtenio.driver.can	Paket yang berisi kelas CAN <i>driver</i>
com.virtenio.driver.cpu	Paket yang berisi kelas CPU <i>driver</i>
com.virtenio.driver.device	Paket yang berisi kelas <i>device driver</i>
com.virtenio.driver.device.at86rf212	Paket yang berisi <i>driver</i> untuk perangkat AT86RF212
com.virtenio.driver.device.at86rf231	Paket yang berisi <i>driver</i> untuk perangkat AT86RF231
com.virtenio.driver.flash	Paket yang berisi kelas <i>Flash driver</i>
com.virtenio.driver.gpio	Paket yang berisi kelas GPIO <i>device driver</i>
com.virtenio.driver.i2c	Paket yang berisi kelas I2C <i>device driver</i>
com.virtenio.driver irq	Paket yang berisi kelas IRQ <i>device driver</i>
com.virtenio.driver.led	Paket yang berisi kelas LED <i>device driver</i>
com.virtenio.driver.lin	Paket yang berisi kelas LIN <i>device driver</i>
com.virtenio.driver.onewire	Paket yang berisi kelas OneWire <i>device driver</i>
com.virtenio.driver.pwm	Paket yang berisi kelas PWM (<i>pulse-width modulation</i>) <i>device driver</i>
com.virtenio.driver.ram	Paket yang berisi kelas FRAM <i>device driver</i>
com.virtenio.driver.rtc	Paket yang berisi kelas untuk pengaturan jam secara <i>real-time</i> , dan <i>real-counter device driver</i>
com.virtenio.driver.spi	Paket yang berisi kelas SPI (<i>Serial Peripheral Interface</i>) <i>device driver</i>
com.virtenio.driver.sw	Paket yang berisi kelas <i>switch device driver</i>
com.virtenio.driver.timer	Paket yang berisi kelas <i>hardware timer device driver</i>
com.virtenio.driver.usart	Paket yang berisi kelas USART <i>device driver</i>
com.virtenio.driver.watchdog	Paket yang berisi WatchDog <i>device driver</i>
com.virtenio.io	Paket Virtenio VM yang berisi IO
com.virtenio.lib	Paket Virtenio VM yang berisi pengaturan classlib
com.virtenio.misc	Paket tambahan Virtenio VM
com.virtenio.net	
com.virtenio.vm	
com.virtenio.vm.event	Sistem event pada Virtenio VM untuk menangani <i>event asynchronous</i> dan <i>synchronous</i>

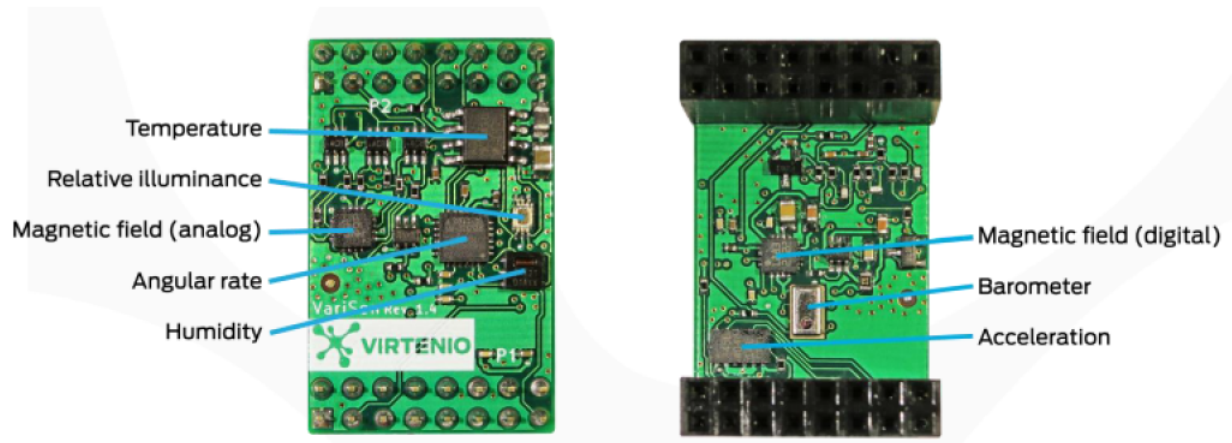
⁶<https://virtenio/com/assets/vm/javadoc/overview-summary.html>

Tabel 2.4: Tabel Java Related Packages

Package	Deskripsi
java.io	Paket Java IO
java.lang	Paket Java lang
java.lang.annotation	Paket <i>Annotation</i> pada Java
java.lang.ref	
java.nio	
java.nio.channels	
java.text	Paket Java Text
java.util	Paket Java <i>Utility</i> yang berisi <i>collection</i>
java.util.regex	Paket Java <i>Regular Expression</i>

2.5.4 Preon32

Preon32 merupakan salah satu sensor node buatan VIRTENIO. Preon32 menggunakan PreonVM digunakan sebagai operating software untuk sensor node ini. Pada umumnya, Preon32 memiliki 5 jenis sensor pada sebuah board. Sensor yang ada antara lain adalah sensor suhu (*temperature sensor*), sensor cahaya (*light intensity sensor*), sensor tekanan udara (*air pressure sensor*), sensor getaran (*acceleration sensor*) dan sensor kelembaban udara (*relative humidity sensor*). Preon32 juga terdapat versi tambahannya dilengkapi dengan sensor untuk mendeteksi *gyroscope* dan medan magnet (Gambar 2.16).



Gambar 2.16: Preon32 Board

2.5.5 Spesifikasi Sensor-Sensor Preon32

Berikut spesifikasi sensor-sensor yang terdapat di Preon32:

- Sensor suhu (*temperature sensor*)
 - Manufacture : Analog Devices
 - Model : ADT7410
 - Interface : digital, I2C
 - Resolution : 16-Bit
 - Range : -40°C sampai +105°C
 - Accuracy : $\pm 0.5^\circ\text{C}$

- Sensor cahaya (*light intensity* sensor)
 - Manufacture : Rohm
 - Model : BH1715FVC
 - Interface : digital, I2C
 - Resolution : 16-Bit
 - Range : 1 lx to 65355 lx
- Sensor tekanan udara (*air pressure* sensor)
 - Manufacture : Freescale
 - Model : MPL115A2
 - Interface : digital, I2C
 - Resolution : 0,15 kPa
 - Range : 50 kPa sampai 115 kPa
 - Accuracy : ± 1.0 kPa
- Sensor getaran (*acceleration* sensor)
 - Manufacture : Analog Devices
 - Model : ADXL345
 - Interface : digital, SPI
 - Resolution : 13 Bit per axis
 - Range : ± 16 g, 3 axis
 - Accuracy : 3,9 mg/LSB
- Sensor kelembaban udara (*relative humidity* sensor)
 - Manufacture : Sensirion
 - Model : SHT21
 - Interface : digital, I2C
 - Resolution : 12-Bit
 - Range : 0 %RH sampai 100 %RH
 - Accuracy : $\pm 2,0$ %RH (typ.)

BAB 3

ANALISIS

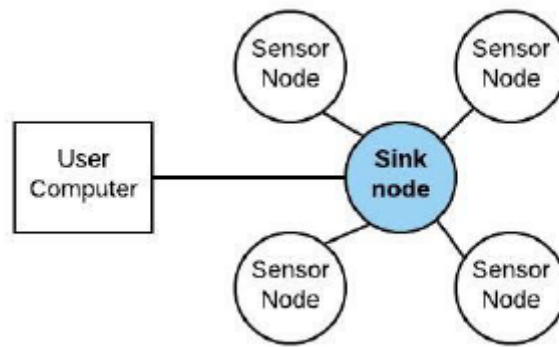
Pada bab ini akan dijelaskan tentang analisis sistem yang merujuk pada hasil studi yang telah dilakukan dan ditulis pada bab sebelumnya. Perangkat lunak dibangun merupakan sebuah perangkat lunak yang berfungsi untuk memantau getaran gedung menggunakan sensor-sensor yang terhubung dalam sebuah jaringan. Sensor-sensor tersebut akan disebar di sekitar area gedung-gedung dan melakukan sensing terhadap getaran yang dihasilkan oleh gedung tersebut. Kemudian sensor-sensor tersebut akan saling komunikasi untuk mengirimkan hasil sensor (*sensing*) ke *base station*. Base station akan menerima semua hasil sensing yang diteruskan dan ditampilkan ke layar komputer. Bab ini juga akan membahas bagaimana perancangan sistem secara umum yang ditunjukkan dalam bentuk flowchart dan usecase diagram.

3.1 Analisis Aplikasi Pemantauan Getaran Gedung Berbasis *Wireless Sensor Network*

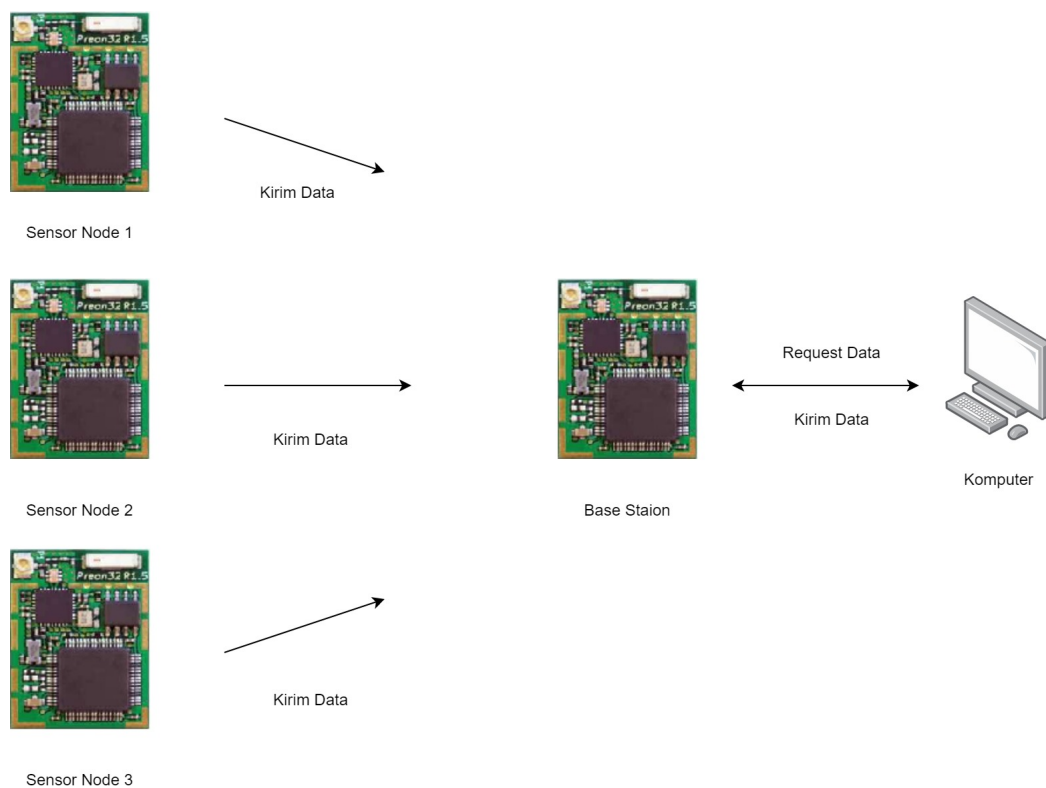
Perangkat lunak yang dibangun tidak dibatasi oleh jenis arsitektur WSN yang dapat dipilih. Jenis arsitektur Wireless Sensor Network flat ataupun hirakikal, dapat digunakan dalam perangkat lunak yang dibangun. Jenis komunikasi *single-hop* ataupun *multi-hop* dapat digunakan pada arsitektur hirakikal seperti yang dibahas pada subbab 2.3.3.

3.1.1 Analisis Arsitektur dan Topologi *Wireless Sensor Network*

Aplikasi yang dibangun untuk melakukan *sensing* terhadap gedung yang diteliti dan melakukan komunikasi agar data yang ada dapat dikirimkan. Berdasarkan pembahasan tentang Arsitektur *Wireless Sensor Network* yang dibahas pada subbab 2.3.3. Aplikasi ini akan menggunakan arsitektur bertipe flat untuk implementasi. Pemilihan arsitektur flat karena wilayah pengujian gedung yang tidak terlalu luas dan tugas dari setiap node yang disebar memiliki tugas yang sama. Sedangkan berdasarkan pembahasan tentang topologi pada subbab 2.3.4, topologi yang diuji untuk membangun aplikasi ini adalah topologi *star* ataupun *linear*. Topologi *linear* dipilih untuk dilakukan pengujian karena topologi *linear* adalah dasar dari komunikasi antar node. Pada topologi *star*, central node akan berfungsi sebagai *sink node* atau dapat dikatakan sebagai *base station* sedangkan node lainnya akan memiliki tugas untuk melakukan *sensing* (Gambar 3.1).



Gambar 3.1: *Wireless Sensor Network* dengan Topologi *Star*



Gambar 3.2: Arsitektur dan Topologi WSN

3.1.2 Analisis Akselerometer

Akselerometer yang akan digunakan dalam penelitian ini adalah akselerometer jenis ADXL345 dari sensor node yang berasal dari Preon32 yang ada pada subbab 2.5. Jenis akselerometer dari Preon32 adalah *three-axis* dan *absolute accelerometer* yang telah dibahas pada subbab 2.4. Hasil *sensing* dari sensor akselerometer akan dikonversi ke dalam satuan gravitasi (g) dan proses konversi sudah ditangani oleh *class library* yang dimiliki oleh PreonVM yang dimana daftar class librarynya dapat dilihat pada subbab 2.5.4.

3.1.3 Analisis Fungsi Aplikasi

Aplikasi yang akan dibangun memiliki fungsi-fungsi utama sebagai berikut:

1. Memeriksa status pada setiap node sensor
2. Melakukan sinkronisasi waktu pada setiap node sensor
3. Mengirimkan perintah yang bertujuan untuk melakukan *sensing*(pengukuran) pada setiap node
4. Mengirimkan perintah yang bertujuan untuk memberhentikan *sensing*(pengukuran) pada setiap node
5. Mengubah data hasil *sensing* dari sinyal analog menjadi digital
6. Mengirimkan data hasil *sensing* ke *base station*
7. Menyimpan data hasil *sensing* yang diterima oleh *base station*
8. Menampilkan data hasil *sensing* yang disimpan oleh *base station*

Fungsi memeriksa status pada setiap node sensor bertujuan untuk memeriksa apakah setiap node aktif atau tidak sebelum melakukan *sensing*. Jika ada sensor node yang statusnya tidak aktif, maka aplikasi akan memberikan informasi berupa adanya node yang tidak aktif dan pengguna dapat melakukan perbaikan atau *maintenance*.

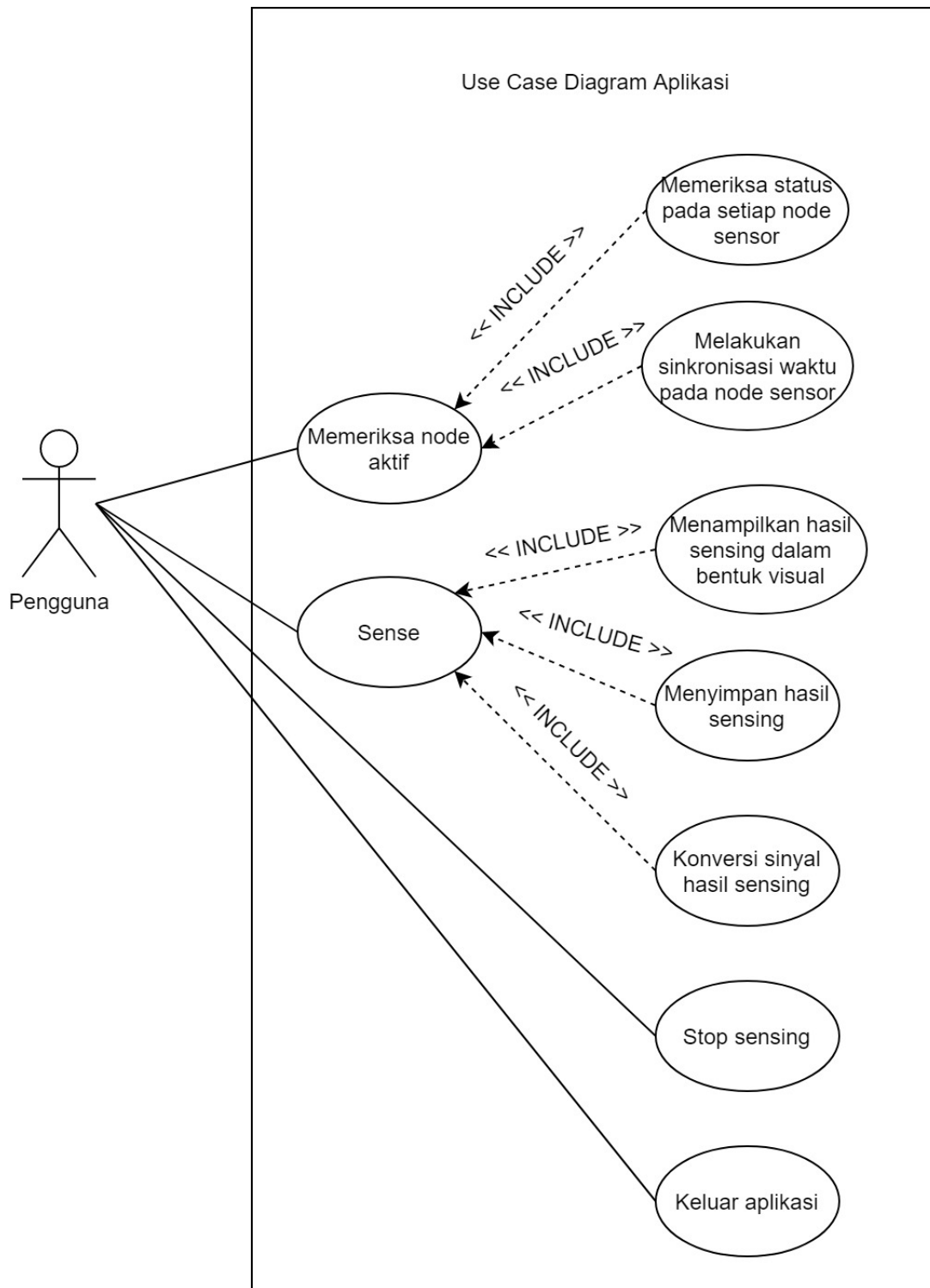
Fungsi melakukan sinkronisasi waktu pada setiap node sensor diperlukan agar setiap node mengirimkan data hasil *sensing* secara bersamaan atau pada waktu yang sama. Selain itu, *memory* yang ada di setiap node sensor memiliki sifat sementara atau *volatile* (subbab 2.3.2). Hal ini menyebabkan adanya kebutuhan untuk menyamakan waktu di setiap node sensor. Karena aplikasi mencatat waktu kapan sensor node itu mati sehingga perlu adanya penyamaan waktu.

Fungsi mengirimkan perintah yang bertujuan untuk melakukan *sensing* pada setiap node digunakan untuk memulai proses *sensing* getaran pada setiap node dengan sensor akselerometer. Setiap hasil pengukuran tersebut akan langsung diekstraksi dan dikirimkan ke *base station*.

Fungsi mengirimkan perintah yang bertujuan untuk memberhentikan *sensing* pada setiap node digunakan untuk memberhentikan proses *sensing* getaran pada sensor node.

Fungsi mengubah data hasil *sensing* dari analog ke digital digunakan untuk mengubah sinyal hasil *sensing* yang berupa analog menjadi digital (dapat dilihat dengan bentuk angka). Hasil konversi ini pun akan dikirimkan langsung ke *base station* dan akan disimpan ke server atau localhost. Kemudian *base station* akan meneruskan hasilnya ke komputer.

Fungsi yang terakhir merupakan fungsi yang bertujuan untuk menampilkan hasil *sensing* nya. Fungsi ini dapat dikatakan sebagai fungsi antarmuka dan antarmuka pada aplikasi ini akan menampilkan data hasil *sensing* pada layar komputer pengguna, selain itu juga, antarmuka digunakan untuk melakukan komunikasi antara *base station* ataupun *node sensor*



Gambar 3.3: Diagram Use Case Aplikasi

Tabel 3.1: Tabel Skenario Memeriksa node aktif

Nama	Memeriksa node aktif
Deskripsi	Memeriksa status node sensor aktif atau tidak dan melakukan penyesuaian waktu sensor node dengan komputer pengguna
Aktor	Pengguna
Pre-kondisi	Aplikasi baru dibuka oleh pengguna
Alur Skenario Utama	<ol style="list-style-type: none"> 1. Sistem menjalankan aplikasi. 2. Pengguna memasukkan banyak node yang dipakai. 3. Sistem menampilkan pilihan fungsi yang dapat digunakan. 4. Pengguna memilih fungsi "Memeriksa node aktif". 5. Sistem akan menampilkan <i>sensor node</i> dan <i>base station</i> yang memiliki status online dan waktu pada setiap <i>sensor node</i>.

Tabel 3.2: Tabel Skenario Memberi perintah sense ke sensor node

Nama	Memberikan perintah sense ke sensor node / Sense
Deskripsi	Memberikan perintah sense ke setiap node sensor dan hasil dari setiap node sensor yang aktif kemudian akan disimpan dan ditampilkan dalam bentuk digital
Aktor	Pengguna
Pre-kondisi	Aplikasi sudah berjalan dan fungsi "Memeriksa node aktif" sudah dijalankan
Alur Skenario Utama	<ol style="list-style-type: none"> 1. Sistem menampilkan pilihan-pilihan fungsi yang dapat dipilih oleh pengguna. 2. Pengguna memilih fungsi "Sense". 3. Sistem akan menampilkan hasil pengukuran (sense) dari setiap node sensor dan menyimpan hasilnya dalam hasil konversi menjadi digital dari setiap node.

Tabel 3.3: Tabel Skenario Stop Sensing

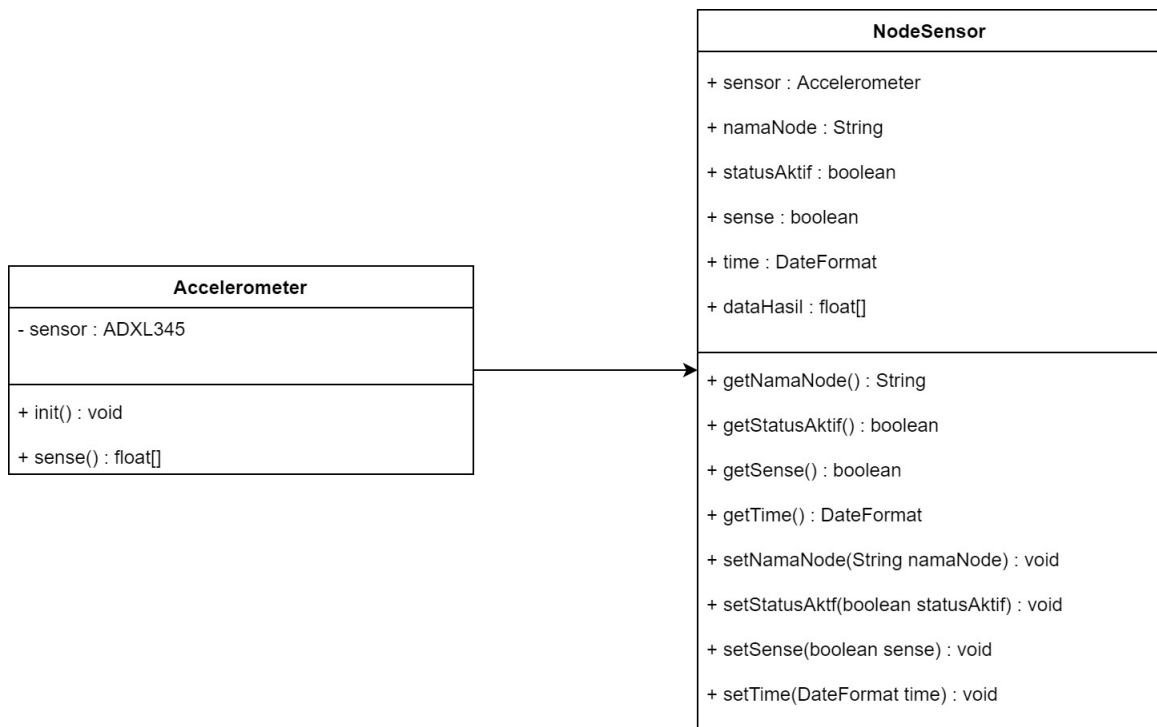
Nama	Memberikan perintah untuk berhenti melakukan sensing (Stop Sensing)
Deskripsi	Memberikan perintah berhenti sensing ke setiap node sensor
Aktor	Pengguna
Pre-kondisi	Aplikasi sudah berjalan dan fungsi aplikasi "Sense" dijalankan
Alur Skenario Utama	<ol style="list-style-type: none"> 1. Sistem menampilkan pilihan-pilihan fungsi yang dapat dipilih oleh pengguna. 2. Pengguna memilih fungsi "Stop Sensing". 3. Sistem akan menjalankan fungsi dan menampilkan pesan tunggu sampai fungsi selesai dijalankan. 4. Semua sensor node akan berhenti melakukan sensing dan tampilan hasil sense akan tertutup dan menampilkan pesan "Sense has stopped".

Tabel 3.4: Tabel Skenario Keluar aplikasi

Nama	Keluar aplikasi
Deskripsi	Memberikan perintah keluar aplikasi untuk semua node sensor dan base station
Aktor	Pengguna
Pre-kondisi	Aplikasi sedang berjalan
Alur Skenario Utama	<ol style="list-style-type: none"> 1. Sistem menampilkan pilihan-pilihan fungsi yang dapat dipilih oleh pengguna. 2. Pengguna memilih fungsi "Keluar aplikasi". 3. Sistem menampilkan pesan "aplikasi telah dihentikan". 4. Aplikasi akan berhenti.

3.1.4 Analisis Kelas

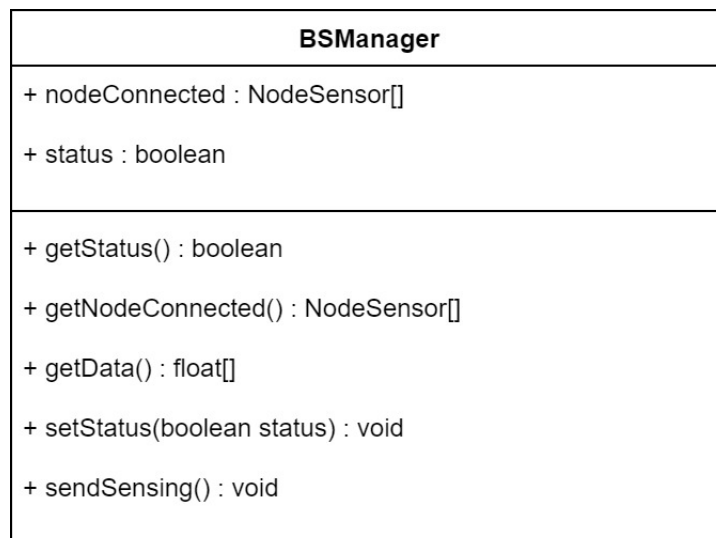
Pembuatan aplikasi pemantauan getaran gedung menggunakan IDE yaitu Eclipse dan SandBox untuk sensor node Preon32 yaitu salah satu jenis PreonVM yang berasal dari Virtenio. Berdasarkan subbab 3.1.3 yang membahas tentang analisis fungsi aplikasi, maka dibuat diagram kelas untuk pembuatan aplikasi ini. Berikut diagram kelas yang dibutuhkan untuk aplikasi ini.



Gambar 3.4: Diagram Kelas NodeSensor

Penjelasan Kelas Accelerometer dan NodeSensor:

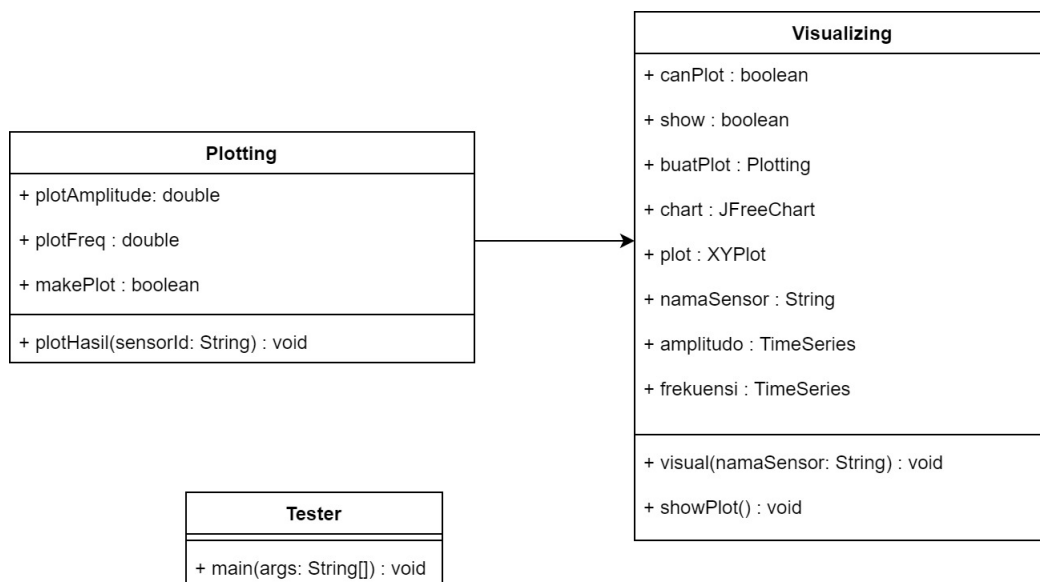
- **Kelas Accelerometer** Pada kelas Accelerometer terdapat sebuah atribut yaitu sensor dengan tipe data ADXL345 dimana fungsi atribut ini adalah mempresentasikan jenis sensor yang akan digunakan. Sedangkan pada methodnya terdapat dua method yaitu `init()` dan `sense()`. Method `init()` digunakan untuk menginisialisasi kelas Accelerometer dan method `sense()` digunakan untuk melakukan sensing (pengukuran).
- **Kelas NodeSensor** Pada Kelas NodeSensor terdapat beberapa atribut berupa sensor dimana atribut ini akan merujuk ke kelas Accelerometer, kemudian atribut `namaNode` untuk memberikan identitas terhadap node, `statusAktif` digunakan untuk memberikan kondisi node sensor sedang aktif atau tidak, atribut `sense` ada untuk memberikan kondisi apakah node sensor sedang melakukan *sensing* (pengukuran) atau tidak. Atribut `time` ada untuk mencatat waktu agar node sensor dapat sama ketika melakukan *sensing*. Atribut `dataHasil` digunakan untuk mencatat hasil pengukuran yang telah dilakukan oleh node sensor. Terdapat juga beberapa method-method untuk kelas NodeSensor berupa *setter* dan *getter* untuk atribut-atribut yang ada pada kelas NodeSensor.



Gambar 3.5: Diagram Kelas BaseStation

Penjelasan Kelas BSManager:

- Kelas BSManager Pada Kelas BaseStation, terdapat dua atribut yaitu nodeConnected dimana ada untuk mengetahui Base Station terhubung ke node yang mana saja kemudian ada atribut status dimana ada untuk mengetahui kondisi Base Station apakah aktif atau tidak. Terdapat beberapa method juga pada kelas BSManager yaitu *setter* dan *getter* untuk atributnya kemudian ada method getData() dimana ada untuk mengambil data hasil *sensing* dari node-node yang terhubung. Method sendSensing() digunakan untuk mengirimkan hasil dari method getData() untuk dikirimkan ke komputer pengguna.



Gambar 3.6: Diagram Kelas Tester

Penjelasan Kelas Plotting, Visualizing dan Tester:

- Kelas Plotting Pada Kelas Plotting, terdapat tiga atribut yaitu plotAmplitude dimana ada untuk mencatat nilai amplitudo dari plot, kemudian plotFreq ada untuk mencatat nilai frequency dari plot sedangkan atribut makePlot ada untuk penanda pembuatan plot.

Untuk method terdapat 1 method yaitu `plotHasil(sensorId)` dimana method digunakan untuk membuat plot pada visualisasi yang sedang berjalan

- Kelas Visualizing Pada Kelas Visualizing, terdapat beberapa atribut yang ada antaranya `canPlot` digunakan untuk penanda bahwa dapat dilakukan plot atau tidak, kemudian atribut `show` digunakan untuk penanda apakah sudah ditampilkan atau belum, atribut `buatPlot` digunakan untuk membuat objek Plotting, atribut `chart` digunakan untuk membuat objek `JFreeChart`, atribut `plot` digunakan untuk mengatur plot yang akan divisualisasikan, atribut `namaSensor` dibuat untuk menyimpan nama sensor, amplitudo digunakan untuk membuat plot time series dari amplitudo dan atribut frekuensi digunakan untuk membuat plot time series dari frekuensi. Method yang ada pada kelas Visualizing adalah `visual(namaSensor)`, method ini digunakan untuk membuat konstruktor dari kelas Visualizing dan method `showPlot()` digunakan untuk menampilkan plot pada aplikasi.
- Kelas Tester Pada kelas Tester, terdapat hanya 1 method yaitu `main(args: String[])` dimana method ini digunakan untuk menjalankan aplikasi secara keseluruhan.

3.1.5 Analisis Paket/Pesan

Hal pertama yang akan dilakukan adalah sensor node akan disebarkan di sekitar bagian gedung, dan melakukan konfigurasi waktu pada setiap node agar waktu mulai sensing dan pengiriman data juga dilakukan bersama. Setelah itu, sensor node akan dinyalakan dan akan dilakukan pengecekan node yang aktif. Cara pengecekan node yang aktif dengan cara mengirimkan pesan yang dikirimkan oleh *base station* ke setiap node bahwa node telah aktif dan sistem akan menampilkan pesan bahwa status node sensor ke sekian yang sebelumnya belum aktif sekarang menjadi aktif. Lalu akan diketahui tujuan data hasil sensing yang akan dikirimkan.

Ketika pengguna memberikan perintah sensing, maka sensor node akan melakukan pengukuran (sensing) dan melakukan pengambilan data. *Base Station* akan mengirimkan pesan berupa perintah untuk melakukan sensing ke sensor node yang ada. Data-data tersebut merupakan data getaran-getaran yang terjadi pada gedung tersebut, kemudian akan dikirimkan ke *base station*. *Base station* akan menerima data hasil sensing dari sensor node secara terus-menerus kemudian akan mengirimkan hasilnya ke aplikasi dan akan ditampilkan oleh aplikasi tersebut.