

# BAB 1

## PENDAHULUAN

### 1.1 Latar Belakang

*Wireless Sensor Network* (WSN) adalah suatu jaringan nirkabel yang terdiri dari kumpulan *sensor node* dengan kemampuan deteksi (*sensing*), komputasi, dan komunikasi yang tersebar pada suatu tempat. Setiap sensor akan mengumpulkan data dari area yang dideteksi seperti temperatur, suara, getaran, tekanan, gerakan, kelembaban udara dan deteksi lainnya tergantung kemampuan sensor tersebut. Data yang diterima ini kemudian akan diteruskan ke *base station* untuk diolah sehingga memberikan suatu informasi. *Wireless Sensor Network* dapat diimplementasikan pada berbagai bidang kehidupan manusia diantaranya bidang militer untuk deteksi musuh, bidang pertanian untuk pemantauan pertumbuhan tanaman, bidang kesehatan, deteksi bahaya dan bencana alam, bidang pembangunan dan tata kota, dan bidang pendidikan.

Terdapat dua macam arsitektur *Wireless Sensor Network*, yaitu kluster dan flat. Pada arsitektur kluster, *node-node* akan disusun secara hierarki dan memiliki peran sebagai *cluster head*, *child node*, atau *parent node*. *Cluster head* berfungsi sebagai pengatur beberapa *child node*. Sedangkan pada arsitektur flat hanya terdapat dua macam *node* secara fungsional, yaitu *source node* dan *sink node*. Semua *sensor node* (*source node*) akan mengirim data ke satu tujuan akhir yaitu *sink node*.

Dalam praktiknya pengiriman data merupakan suatu hal yang penting pada *Wireless Sensor Network*. Data yang didapat dari sensor harus dapat sampai ke *base station* dengan utuh dan akurat (*reliable*). Data yang *reliable* ini sangat penting karena hasil pengukuran dan tindakan selanjutnya yang akan diambil akan bergantung pada data-data tersebut. Terdapat beberapa cara untuk memastikan transfer data *reliable* yaitu dengan *Link-Layer Retransmission*, *End-to-End Retransmission*, dan *Erasure Code*.

Pada skripsi ini dibangun aplikasi untuk transfer data pada *Wireless Sensor Network*. *Wireless Sensor Network* yang dibuat juga dapat melakukan transfer data ke *sensor node* tetangganya hingga sampai ke *sensor node* yang berperan sebagai *base station*. Karena *sensor node* memiliki kapasitas penyimpanan yang kecil dan data yang *reliable* sangat dibutuhkan untuk menentukan tindakan selanjutnya, maka akan dibangun juga *Wireless Sensor Network* yang memiliki sifat *reliable* tersebut.

### 1.2 Rumusan Masalah

- Bagaimana cara membangun aplikasi transfer data dari setiap *sensor node* pada *Wireless Sensor Network*?
- Bagaimana cara membangun aplikasi transfer data yang *reliable* pada *Wireless Sensor Network*?

### 1.3 Tujuan

- Membangun aplikasi transfer data yang *reliable* pada *Wireless Sensor Network*.

## 1.4 Batasan Masalah

Penelitian ini dibuat berdasarkan batasan-batasan sebagai berikut:

1. Sensor yang digunakan sebagai penelitian hanya sensor untuk mengukur temperatur, kelembapan, getaran, dan tekanan udara.
2. Arsitektur yang digunakan untuk membangun *Wireless Sensor Network* ini adalah flat dan kluster.

## 1.5 Metodologi

Berikut adalah metode penelitian yang digunakan dalam penelitian ini:

1. Melakukan studi literatur mengenai *Wireless Sensor Network*.
2. Mempelajari protokol transfer data yang biasa pada *Wireless Sensor Network*.
3. Mempelajari prinsip *Reliable Data Transfer* pada *Wireless Sensor Network*.
4. Mempelajari pemrograman pada *Wireless Sensor Network* dengan Bahasa Pemrograman JAVA.
5. Melakukan perancangan perangkat lunak.
6. Mengimplementasi rancangan perangkat lunak pada *Wireless Sensor Network*.

## 1.6 Sistematika Pembahasan

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

1. Bab 1: Pendahuluan yaitu membahas mengenai gambaran umum penelitian ini. Berisi tentang latar belakang, rumusan masalah, tujuan, batasan masalah, metode penelitian, dan sistematika penulisan
2. Bab 2: Dasar Teori, yaitu membahas teori-teori yang mendukung berjalannya penelitian ini. Berisi tentang *Wireless Sensor Network*,
3. Bab 3: Analisis,
4. Bab 4: Perancangan,
5. Bab 5: Implementasi dan Pengujian,
6. Bab 6: Kesimpulan,

## BAB 2

### LANDASAN TEORI

Pada bab ini dijelaskan dasar-dasar teori mengenai *Wireless Sensor Network*, transfer data pada wireless sensor network, prinsip *reliable data transfer* pada Wireless Sensor Network, dan pengembangan pemrograman pada WSN.

#### 2.1 Wireless Sensor Network

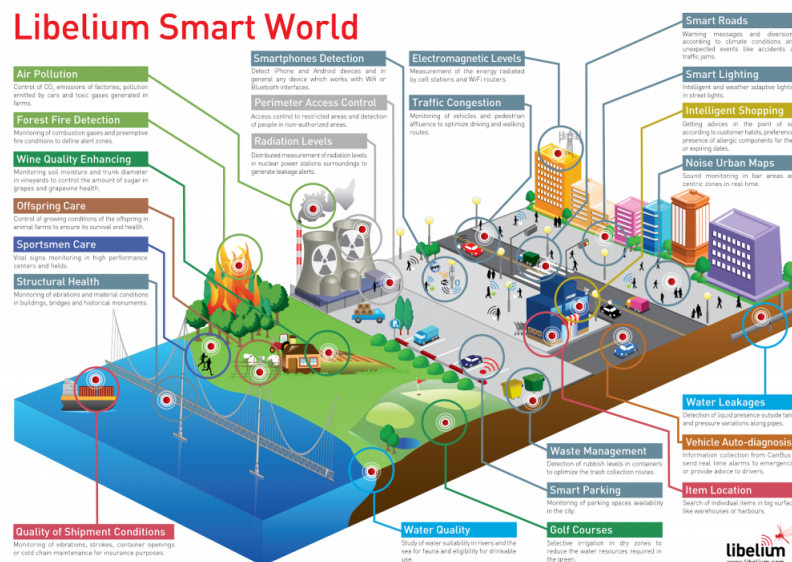
*Wireless Sensor Network* merupakan jaringan nirkabel yang terdiri dari sekumpulan *sensor node* yang diletakan pada suatu tempat dan memiliki kemampuan untuk mengukur kondisi lingkungan sekitar (*sensing*), komputasi dan dilengkapi dengan alat komunikasi *wireless* untuk komunikasi antar *sensor node*. Sensor ini akan mengumpulkan data dari kondisi lingkungannya, seperti: cahaya, suara, kelembapan, getaran, gerakan, temperatur, tekanan udara, kualitas air, komposisi tanah, dan lain-lain. Data ini kemudian dikirimkan ke *sensor node* tetangganya hingga sampai ke *base station* sebagai pusat untuk dikelola.

##### 2.1.1 Penerapan Wireless Sensor Network

Pada awalnya *sensor network* (jaringan sensor) digunakan dalam teknologi militer untuk mendeteksi musuh di laut dan di darat. Semakin lama sensor node ini banyak dikembangkan untuk membantu berbagai bidang kehidupan manusia. Pemanfaatan Wireless Sensor Network pada kehidupan manusia dapat dilihat pada ilustrasi Gambar 2.1. Berikut adalah beberapa penerapan wireless sensor network pada berbagai bidang kehidupan manusia:

- Bidang militer  
*Wireless Sensor Network* digunakan sebagai bagian dari komunikasi pada bidang militer dan deteksi target atau musuh.
- Monitoring area  
Pada *monitoring area*, *sensor node* akan disebar pada suatu tempat yang akan di monitoring. Saat *sensor node* mendeteksi kejadian (panas, tekanan, dan lain-lain) pada suatu tempat, data akan dikirimkan ke *base station* untuk ditentukan tindakan selanjutnya.
- Transportasi  
Pada bidang transportasi *Wireless Sensor Network* digunakan untuk mendeteksi lalu lintas secara aktual yang nantinya akan disampaikan kepada pengendara tentang kejadian di depan mereka seperti kemacetan lalu lintas.
- Kesehatan  
Beberapa aplikasi kesehatan seperti membantu pada disabilitas, monitoring pasien, diagnosis, pengaturan penggunaan obat, dan pelacakan dokter dan pasien di rumah sakit.
- Deteksi lingkungan  
Deteksi lingkungan yang dapat dilakukan antara lain deteksi gunung berapi, polusi udara, kebakaran hutan, efek rumah kaca, dan deteksi longsor.

- Monitoring struktur  
*Wireless Sensor Network* dapat melakukan deteksi pergerakan bangunan dan infrastruktur seperti jembatan, *flyover*, terowongan dan fasilitas lain tanpa mengeluarkan biaya untuk melakukan dektesi manual dengan mendatangi tempatnya secara langsung.
- Bidang pertanian  
 Pada bidang pertanian dapat membantu pengelola pertanian untuk melihat penggunaan air dalam irigasi dan mengelola buangan pertanian mereka.

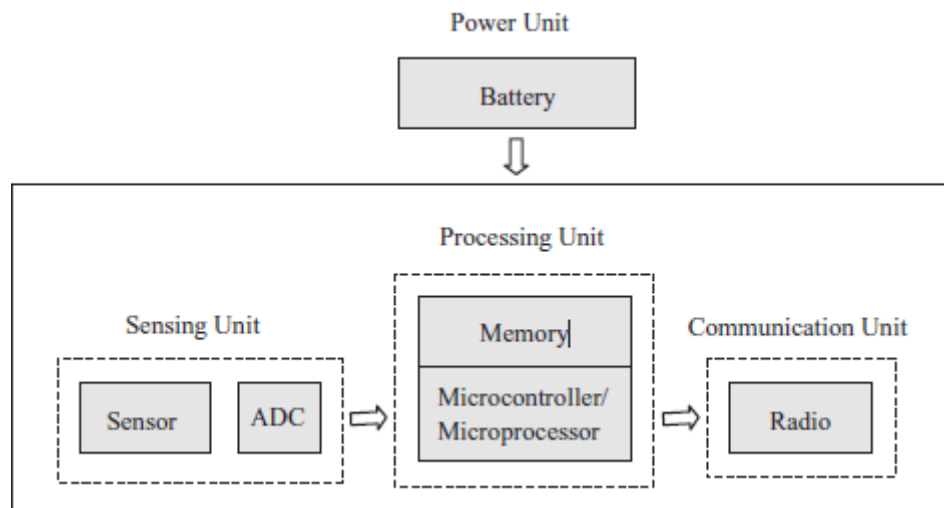


Gambar 2.1: Ilustrasi Pemanfaatan *Wireless Sensor Network*

## 2.1.2 Sensor Node

### Struktur Sensor Node

Setiap sensor node memiliki kemampuan deteksi, komputasi dan komunikasi. Sensor node memiliki lima komponen utama yaitu *controller*, *memory*, *sensor and actuator*, *communication device*, dan *power supply*, Gambar 2.2. Semua komponen akan bekerja secara seimbang dalam melakukan sensing, komputasi, komunikasi, dan menjaga penggunaan energi seminimal mungkin. *Sensing unit* adalah sensor yang terdapat pada sensor node, sensor unit akan melakukan *analog-to-digital converter*. Sensing unit akan melakukan deteksi lingkungan dan data yang didapat merupakan sinyal analog, sinyal ini akan diubah menjadi sinyal digital dan dikirimkan ke *processing unit*. *Power unit* pada sensor node biasanya adalah baterai sebagai sumber energi.



Gambar 2.2: Struktur Sensor Node

## Controller

Controller adalah inti utama pada sensor node. Controller mengumpulkan data dari sensor dan memproses data tersebut hingga menentukan kapan dan kemana data tersebut dikirim. Controller menerima data dari sensor node lain. Pada controller biasanya terdapat microcontroller atau microprocessor yang mengatur dan melakukan komputasi data dari sensor node. Microcontroller ini juga dapat mengurangi penggunaan energi dengan masuk ke sleep states yang berarti hanya bagian dari controller saja yang aktif.

Beberapa microcontroller yang digunakan dalam Wireless Sensor Node:

- Intel StrongARM (32-bit RISC, up to 206 MHz)
- Texas Instrument MSP 430 (16-bit RISC, up to 4 MHz, RAM 2-10 kB)
- Atmel Atmega 128L (8-bit)

## Memory

Random Access Memory (RAM) digunakan untuk menyimpan sementara hasil yang didapat dari sensor. RAM juga menyimpan sementara paket dari sensor node lain. Saat power supply terjadi masalah maka data pada RAM ini akan hilang. Ini merupakan salah satu kekurangan dari penggunaan RAM. Maka untuk menyimpan kode program digunakanlah Read Only Memory (ROM). ROM ini biasa disebut Electrically Erasable Programmable Read-Only Memory (EEPROM) atau Flash Memory. Flash memory dapat menyimpan data jika suatu saat data pada RAM hilang atau energi sudah habis (intermediate storage).

## Communication Device

*Communication Device* digunakan untuk bertukar data antar sensor node. Pada aplikasi Wireless Sensor Network, Radio Frequency (RF)-based adalah media komunikasi yang paling relevan saat ini. RF-based mendukung jangkauan yang jauh, data rate yang tinggi, dapat menerima error rate pada penggunaan energi, dan tidak perlu saling mengetahui posisi antara penerima dan pengirim.

Pada sensor node dibutuhkan transmitter dan receiver untuk menerima dan mengirim data. Kedua hal ini dapat digabung dan disebut dengan transceiver. Tugas transceiver adalah mengubah stream bit dari microcontroller dan mengubahnya menjadi gelombang radio. Selain itu transceiver juga dapat mengubah gelombang radio menjadi stream bit.

## 1 Sensor dan Actuator

Sensor dan Actuator adalah hal yang penting pada Wireless Sensor Network. Tanpa sensor dan actuator maka sensor node tidak berguna dan tidak dapat digunakan. Tabel 2.1 adalah jenis - jenis sensor yang dapat dimiliki *sensor node*. Sensor dikategorikan menjadi tiga:

1. **Passive, omnidirectional sensors** Sensor ini dapat mengukur kualitas dari lingkungan fisik tempat sensor node tersebut tanpa mengubah lingkungannya. Beberapa sensor dikategori ini *self-powered*, sensor mendapatkan energi yang mereka butuhkan dari lingkungannya. Energi ini digunakan untuk memperkuat sinyal analog. *Omnidirectional* berarti tidak ada arah pada sensor ini. Sensor akan memancarkan sinyalnya ke segala arah. Contoh sensor ini adalah termometer, sensor cahaya, sensor getaran, mikrofon, sensor kelembapan, sensor tekanan udara, sensor deteksi asap, dan lain-lain.
2. **Passive, narrow-beam sensors** Sensor ini memiliki sifat yang sama yaitu pasif, tidak mengubah lingkungannya. Sensor ini dapat melakukan gerakan dan memiliki arah atau daerah pengukuran. Contoh dari sensor ini adalah kamera yang bisa mengukur sesuai dengan arah yang dituju.
3. **Active Sensor** Sensor ini aktif dalam memeriksa lingkungannya. Contoh dari sensor ini adalah sonar, radar atau sensor seismik. Sensor ini menghasilkan gelombang dengan ledakan kecil untuk melakukan deteksi.

Actuator jumlahnya beragam seperti sensor. Actuator adalah penerima sinyal dan yang mengubahnya ke aksi fisik. Salah satu aktuatur adalah LED, yang mengubah listrik menjadi cahaya. Motor (electrical motor) juga mengubah listrik menjadi gerakan.

Tabel 2.1: Jenis - jenis sensor yang dapat dimiliki sensor node

Sensor	Sense Event	Remark
Accelerometer		
Acoustic emission sensor		
Acoustic sensor		
Capacitance sensor		
ECG		
EEG		
EMG		
Electrical/electromagnetic sensor		
Gyroscope		
Humidity Sensor		
Infrasonic sensor		
Magnetic sensor		
Oximeter		
pH sensor		
Photo acoustic spectroscopy		
Piezoelectric cylinder		
Soil moisture sensor		
Temperature sensor		
Barometer sensor		
Passive infrared sensor		
Seismic sensor		
Oxygen sensor		
Blood flow sensor		

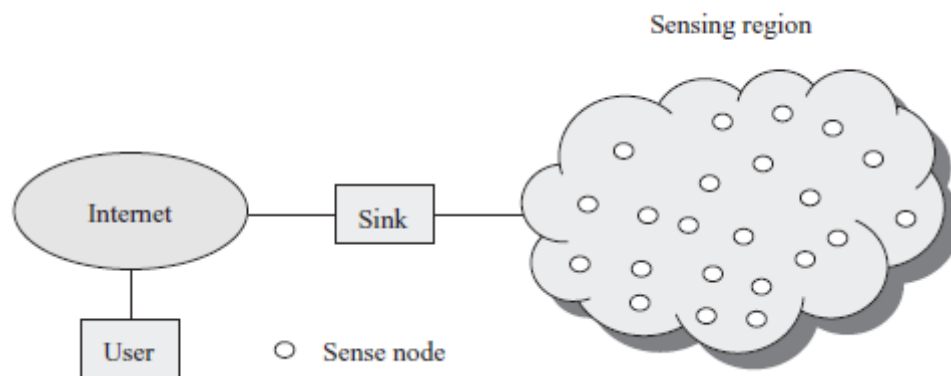
## 1 Power Supply

2 Power supply atau sumber energi pada Wireless Sensor Network bisa berasal dari dua cara yaitu  
 3 **storing energy** dan **energy scavenging**. *Storing energy* adalah dengan menggunakan baterai  
 4 sebagai sumber energinya. Baterai yang digunakan dapat diisi ulang maupun yang tidak dapat  
 5 diisi ulang. *Energy scavenging* digunakan saat membuat Wireless Sensor Network yang akan  
 6 digunakan dalam waktu yang lama. Dibutuhkan energi yang bisa dikatakan tidak terbatas. Salah  
 7 satu cara *energy scavenging* adalah *photovoltaics*. *photovoltaics* biasa disebut juga solar cell yang  
 8 memanfaatkan cahaya matahari dan mengubahnya menjadi energi sebagai pembangkit daya. Cara  
 9 lain yang bisa digunakan adalah pemanfaatan angin dan air untuk menggerakkan kincir atau turbin  
 10 yang akan menghasilkan listrik dan digunakan sebagai sumber energi pada sensor node.

### 11 2.1.3 Topologi Jaringan

### 12 2.1.4 Arsitektur Wireless Sensor Network

13 Pada *Wireless Sensor Network* biasanya akan terdapat banyak *sensor node* yang disebar pada  
 14 suatu tempat. Terdapat satu atau lebih *sink node* atau *base station* dalam area sensing tersebut  
 15 (Gambar 2.3). *sink node* atau *base station* adalah sensor node yang bertugas untuk mendapatkan  
 16 data dari *sensor node* lain. Dalam membuat *Wireless Sensor Network* perlu diperhatikan topologi  
 17 yang akan digunakan. Tidak semua topologi jaringan komputer dapat digunakan untuk *Wireless*  
 18 *Sensor Network*. Topologi yang biasanya dipakai pada *Wireless Sensor Network* adalah **topologi**  
 19 **flat** dan **topologi hierarki**. Selain itu dalam membangun *Wireless Sensor Network* perlu juga  
 20 diperhatikan jalur komunikasi yang digunakan untuk menghubungkan antar *sensor node* saat  
 21 transfer data. Untuk area *sensing* yang tidak terlalu luas dan hanya menggunakan sedikit *sensor*  
 22 *node* dapat menggunakan cara komunikasi **single hop**. Sedangkan untuk daerah yang luas dan  
 23 memerlukan banyak *sensor node* dapat menggunakan cara komunikasi **multi hop**.



Gambar 2.3: Arsitektur Wireless Sensor Network

### 24 Single-Hop dan Multi-Hop

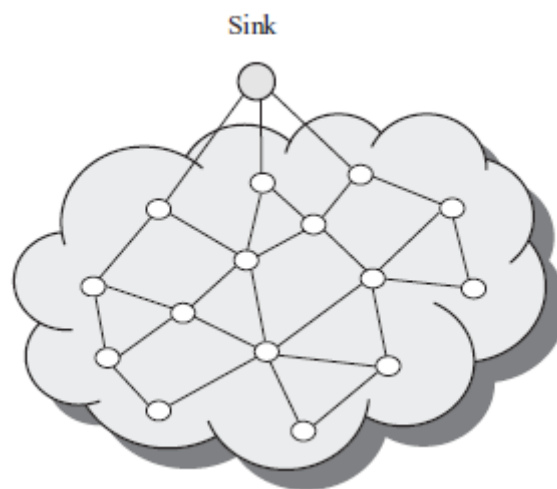
25 Untuk mengirim data ke sink node setiap sensor node dapat menggunakan single-hop long-distance  
 26 transmission. Single-hop long-distance ini berarti setiap sensor node akan mengirimkan data ke  
 27 sink node hanya satu kali lompatan walaupun jarak antara sink node dengan sensor node itu sangat  
 28 jauh. Dalam jaringan sensor, penggunaan energi paling besar adalah saat melakukan komunikasi  
 29 dibandingkan saat sensing. Penggunaan energi akan semakin bertambah jika jarak sink dan sensor  
 30 node semakin jauh. Untuk menangani masalah tersebut muncul protokol multi-hop.

31 Pada protokol multi-hop sensor node akan disusun saling berdekatan dan terhubung dengan  
 32 yang lain. Jadi saat akan berkomunikasi dengan sink node, sensor node harus mengirimkan data  
 33 tersebut ke sensor node tetangganya dan diteruskan hingga sampai ke sink node. Karena jarak

yang saling berdekatan maka penggunaan energi dapat efektif. Single-hop dan multi-hop ini dapat digunakan dengan topologi flat maupun hierarki sesuai dengan kebutuhan sistem.

### Topologi Flat

Pada topologi flat, setiap *sensor node* memiliki peran atau *role* yang sama dalam melakukan *sensing*. Secara fungsional hanya terdapat dua macam *sensor node* pada topologi flat, yaitu *source node* dan *sink node*. Karena jumlah *sensor node* yang banyak maka tidak mungkin menentukan *global identified* untuk setiap *sensor node* pada jaringan ini. Untuk mendapatkan data dilakukan dengan cara *sink node* melakukan pengiriman data ke semua *sensor node* pada area *sensing* dengan cara *flooding* dan hanya *sensor node* yang sesuai yang akan merespon *sink node*. Setiap *sensor node* mengirimkan data ke *sink node* dengan *multi hop* dan melalui node tetangganya yang terhubung dengannya untuk meneruskan data (Gambar 2.4).



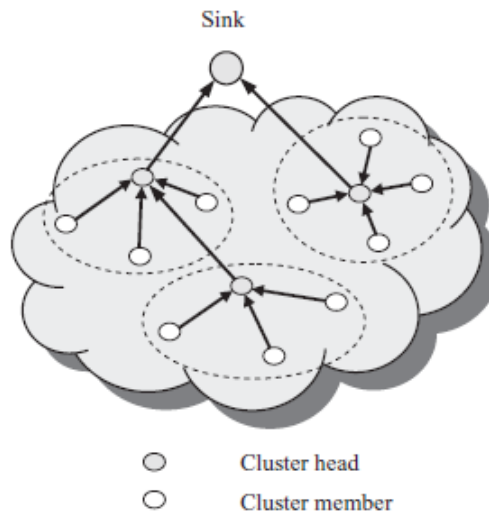
Gambar 2.4: Topologi flat pada *Wireless Sensor Network*

### Topologi Hierarki

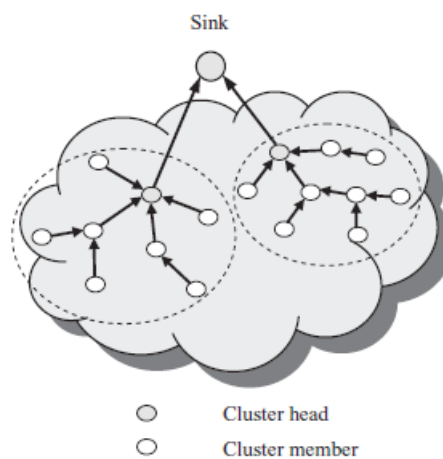
Pada topologi hierarki, semua sensor node dikelompokkan ke dalam cluster-cluster. Terdapat cluster head pada setiap cluster. Cluster head ini yang mengumpulkan data dari setiap sensor node di bawahnya dan meneruskan data yang telah diterima ke base station atau sink. Hal yang perlu diperhatikan pada arsitektur hierarki adalah pemilihan sensor node sebagai cluster head dan sensor node yang melakukan *sensing*. Penggunaan energi yang paling besar dalam *Wireless Sensor Network* ini adalah saat melakukan komunikasi yaitu saat mengirimkan data ke sensor node lain. Maka untuk sensor node yang memiliki energi kecil dapat digunakan untuk *sensing*, karena sensor node *sensing* ini hanya melakukan komunikasi ke cluster head. Cluster head harus memiliki energi atau daya yang lebih banyak, karena cluster head akan bertugas menerima hasil *sensing* sensor node di bawahnya dan meneruskan data ke sink node.

Masalah yang utama pada clustering ini adalah pemilihan cluster head dan bagaimana cara mengatur setiap cluster. Terdapat beberapa cara untuk membuat clustering ini. Berdasarkan jarak antara cluster head dengan cluster member, dapat dibuat clustering dengan single hop atau multi hop seperti pada Gambar 2.5 dan Gambar 2.6. Sedangkan jika berdasarkan jumlah tier dapat dibangun clustering single tier atau multi tier (Gambar 2.7).

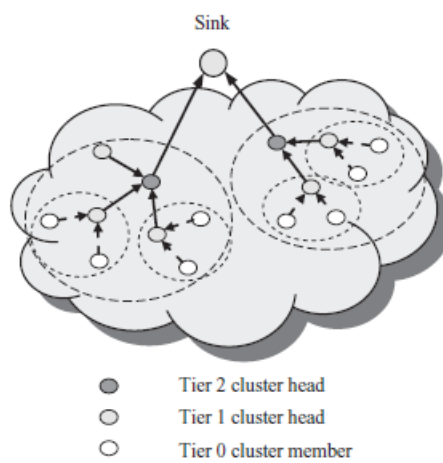




Gambar 2.5: Topologi hierarki pada *Wireless Sensor Network* dengan *single hop* terhadap *Cluster Head*



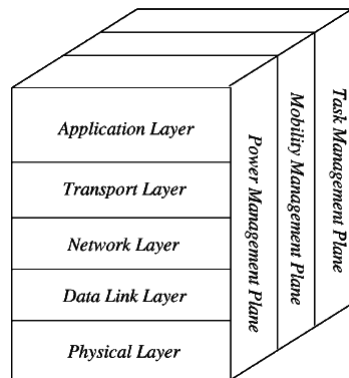
Gambar 2.6: Topologi hierarki pada *Wireless Sensor Network* dengan *multi hop*



Gambar 2.7: Clustering dengan multi tier

### 2.1.5 Layer pada Wireless Sensor Network

Wireless Sensor Network memiliki lima layer protokol: physical layer, data link layer, network layer, transport layer, dan application layer, seperti pada Gambar 2.8.



Gambar 2.8: Layer pada *Wireless Sensor Network*

### 2.1.6 Sistem Operasi

## 2.2 Data Transfer pada Wireless Sensor Network

## 2.3 Prinsip Reliable Data Transfer pada Wireless Sensor Network

Refrensi dari paper reliable. isinya link level retransmission, end to end retransmission, erasure code, Alternate Route (BVR) , dan route fix.

## 2.4 Pengembangan Pemrograman pada Wireless Sensor Network

### 2.4.1 Tabel

Berikut adalah contoh pembuatan tabel. Penempatan tabel dan gambar secara umum diatur secara otomatis oleh  $\text{\LaTeX}$ , perhatikan contoh di file bab2.tex untuk melihat bagaimana cara memaksa tabel ditempatkan sesuai keinginan kita.

Perhatikan bawa berbeda dengan penempatan judul gambar gambar, keterangan tabel harus diletakkan di atas tabel!! Lihat Tabel 2.2 berikut ini:

Tabel 2.2: Tabel contoh

	$v_{start}$	$\mathcal{S}_1$	$v_{end}$
$\tau_1$	1	12	20
$\tau_2$	1		20
$\tau_3$	1	9	20
$\tau_4$	1		20

Tabel 2.3 dan Tabel 2.4 berikut ini adalah tabel dengan sel yang berwarna dan ada dua tabel yang bersebelahan.

Tabel 2.3: Tabel bewarna(1)

	$v_{start}$	$\mathcal{S}_2$	$\mathcal{S}_1$	$v_{end}$
$\tau_1$	1	5	12	20
$\tau_2$	1	8		20
$\tau_3$	1	2/8/17	9	20
$\tau_4$	1			20

Tabel 2.4: Tabel bewarna(2)

	$v_{start}$	$\mathcal{S}_1$	$\mathcal{S}_2$	$v_{end}$
$\tau_1$	1	12	5	20
$\tau_2$	1		8	20
$\tau_3$	1	9	2/8/17	20
$\tau_4$	1			20

### 2.4.2 Kutipan

Berikut contoh kutipan dari berbagai sumber, untuk keterangan lebih lengkap, silahkan membaca file referensi.bib yang disediakan juga di template ini. Contoh kutipan:

- Buku: [?]
- Bab dalam buku: [?]
- Artikel dari Jurnal: [?]
- Artikel dari prosiding seminar/konferensi: [?]
- Skripsi/Thesis/Disertasi: [?] [?] [?]
- Technical/Scientific Report: [?]
- RFC (Request For Comments): [?]
- Technical Documentation/Technical Manual: [?] [?] [?]
- Paten: [?]
- Tidak dipublikasikan: [?] [?]
- Laman web: [?]
- Lain-lain: [?]

### 2.4.3 Gambar

Pada hampir semua editor, penempatan gambar di dalam dokumen  $\text{\LaTeX}$  tidak dapat dilakukan melalui proses *drag and drop*. Perhatikan contoh pada file bab2.tex untuk melihat bagaimana cara menempatkan gambar. Beberapa hal yang harus diperhatikan pada saat menempatkan gambar:

- Setiap gambar **harus** diacu di dalam teks (gunakan *field* LABEL)
- *Field* CAPTION digunakan untuk teks pengantar pada gambar. Terdapat dua bagian yaitu yang ada di antara tanda [ dan ] dan yang ada di antara tanda { dan }. Yang pertama akan muncul di Daftar Gambar, sedangkan yang kedua akan muncul di teks pengantar gambar. Untuk skripsi ini, samakan isi keduanya.
- Jenis file yang dapat digunakan sebagai gambar cukup banyak, tetapi yang paling populer adalah tipe PNG (lihat Gambar 2.9), tipe JPG (Gambar 2.10) dan tipe PDF (Gambar 2.11)
- Besarnya gambar dapat diatur dengan *field* SCALE.
- Penempatan gambar diatur menggunakan *placement specifier* (di antara tanda [ dan ] setelah deklarasi gambar. Yang umum digunakan adalah **H** untuk menempatkan gambar **sesuai** penempatannya di file .tex atau **h** yang berarti "kira-kira" di sini. Jika tidak menggunakan *placement specifier*,  $\text{\LaTeX}$  akan menempatkan gambar secara otomatis untuk menghindari bagian kosong pada dokumen anda. Walaupun cara ini sangat mudah, hindarkan terjadinya penempatan dua gambar secara berurutan.
  - Gambar 2.9 ditempatkan di bagian atas halaman, walaupun penempatannya dilakukan setelah penulisan 3 paragraf setelah penjelasan ini.
  - Gambar 2.10 dengan skala 0.5 ditempatkan di antara dua buah paragraf. Perhatikan penulisannya di dalam file bab2.tex!
  - Gambar 2.11 ditempatkan menggunakan *specifier* **h**.

Curabitur tellus magna, porttitor a, commodo a, commodo in, tortor. Donec interdum. Praesent scelerisque. Maecenas posuere sodales odio. Vivamus metus lacus, varius quis, imperdiet quis, rhoncus a, turpis. Etiam ligula arcu, elementum a, venenatis quis, sollicitudin sed, metus. Donec nunc pede, tincidunt in, venenatis vitae, faucibus vel, nibh. Pellentesque wisi. Nullam malesuada. Morbi ut tellus ut pede tincidunt porta. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam congue neque id dolor.

Donec et nisl at wisi luctus bibendum. Nam interdum tellus ac libero. Sed sem justo, laoreet vitae, fringilla at, adipiscing ut, nibh. Maecenas non sem quis tortor eleifend fermentum. Etiam id tortor ac mauris porta vulputate. Integer porta neque vitae massa. Maecenas tempus libero a libero posuere dictum. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Aenean quis mauris sed elit commodo placerat. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Vivamus rhoncus tincidunt libero. Etiam elementum pretium justo. Vivamus est. Morbi a tellus eget pede tristique commodo. Nulla nisl. Vestibulum sed nisl eu sapien cursus rutrum.

Nulla non mauris vitae wisi posuere convallis. Sed eu nulla nec eros scelerisque pharetra. Nullam varius. Etiam dignissim elementum metus. Vestibulum faucibus, metus sit amet mattis rhoncus, sapien dui laoreet odio, nec ultricies nibh augue a enim. Fusce in ligula. Quisque at magna et nulla commodo consequat. Proin accumsan imperdiet sem. Nunc porta. Donec feugiat mi at justo. Phasellus facilisis ipsum quis ante. In ac elit eget ipsum pharetra faucibus. Maecenas viverra nulla in massa.

Nulla ac nisl. Nullam urna nulla, ullamcorper in, interdum sit amet, gravida ut, risus. Aenean ac enim. In luctus. Phasellus eu quam vitae turpis viverra pellentesque. Duis feugiat felis ut enim. Phasellus pharetra, sem id porttitor sodales, magna nunc aliquet nibh, nec blandit nisl mauris

Gambar 2.9: Gambar *Serpentes* dalam format png

1 at pede. Suspendisse risus risus, lobortis eget, semper at, imperdiet sit amet, quam. Quisque  
 2 scelerisque dapibus nibh. Nam enim. Lorem ipsum dolor sit amet, consectetur adipiscing elit.  
 3 Nunc ut metus. Ut metus justo, auctor at, ultrices eu, sagittis ut, purus. Aliquam aliquam.



Gambar 2.10: Ular kecil

4 Etiam pede massa, dapibus vitae, rhoncus in, placerat posuere, odio. Vestibulum luctus commodo  
 5 lacus. Morbi lacus dui, tempor sed, euismod eget, condimentum at, tortor. Phasellus aliquet odio ac  
 6 lacus tempor faucibus. Praesent sed sem. Praesent iaculis. Cras rhoncus tellus sed justo ullamcorper  
 7 sagittis. Donec quis orci. Sed ut tortor quis tellus euismod tincidunt. Suspendisse congue nisl eu elit.  
 8 Aliquam tortor diam, tempus id, tristique eget, sodales vel, nulla. Praesent tellus mi, condimentum  
 9 sed, viverra at, consectetur quis, lectus. In auctor vehicula orci. Sed pede sapien, euismod in,  
 10 suscipit in, pharetra placerat, metus. Vivamus commodo dui non odio. Donec et felis.

11 Etiam suscipit aliquam arcu. Aliquam sit amet est ac purus bibendum congue. Sed in eros.  
 12 Morbi non orci. Pellentesque mattis lacinia elit. Fusce molestie velit in ligula. Nullam et orci vitae  
 13 nibh vulputate auctor. Aliquam eget purus. Nulla auctor wisi sed ipsum. Morbi porttitor tellus ac  
 14 enim. Fusce ornare. Proin ipsum enim, tincidunt in, ornare venenatis, molestie a, augue. Donec  
 15 vel pede in lacus sagittis porta. Sed hendrerit ipsum quis nisl. Suspendisse quis massa ac nibh  
 16 pretium cursus. Sed sodales. Nam eu neque quis pede dignissim ornare. Maecenas eu purus ac urna  
 17 tincidunt congue.



Gambar 2.11: *Serpentes* jantan

# LAMPIRAN A

## KODE PROGRAM

Listing A.1: MyCode.c

```

1 // This does not make algorithmic sense,
2 // but it shows off significant programming characters.
3
4
5 #include<stdio.h>
6
7 void myFunction( int input, float* output ) {
8     switch ( array[i] ) {
9         case 1: // This is silly code
10             if ( a >= 0 || b <= 3 && c != x )
11                 *output += 0.005 + 20050;
12             char = 'g';
13             b = 2^n + ~right_size - leftSize * MAX_SIZE;
14             c = (--aaa + &daa) / (bbb++ - ccc % 2 );
15             strcpy(a,"hello_$@?");
16         }
17         count = ~mask | 0x00FF00AA;
18     }
19
20 // Fonts for Displaying Program Code in LATEX
21 // Adrian P. Robson, nepsweb.co.uk
22 // 8 October 2012
23 // http://nepsweb.co.uk/docs/progfonts.pdf

```

Listing A.2: MyCode.java

```

1 import java.util.ArrayList;
2 import java.util.Collections;
3 import java.util.HashSet;
4
5 //class for set of vertices close to furthest edge
6 public class MyFurSet {
7     protected int id; //id of the set
8     protected MyEdge FurthestEdge; //the furthest edge
9     protected HashSet<MyVertex> set; //set of vertices close to furthest edge
10    protected ArrayList<ArrayList<Integer>> ordered; //list of all vertices in the set for each trajectory
11    protected ArrayList<Integer> closeID; //store the ID of all vertices
12    protected ArrayList<Double> closeDist; //store the distance of all vertices
13    protected int totaltrj; //total trajectories in the set
14
15    /*
16     * Constructor
17     * @param id : id of the set
18     * @param totaltrj : total number of trajectories in the set
19     * @param FurthestEdge : the furthest edge
20     */
21    public MyFurSet(int id,int totaltrj,MyEdge FurthestEdge) {
22        this.id = id;
23        this.totaltrj = totaltrj;
24        this.FurthestEdge = FurthestEdge;
25        set = new HashSet<MyVertex>();
26        ordered = new ArrayList<ArrayList<Integer>>();
27        for (int i=0;i<totaltrj;i++) ordered.add(new ArrayList<Integer>());
28        closeID = new ArrayList<Integer>(totaltrj);
29        closeDist = new ArrayList<Double>(totaltrj);
30        for (int i = 0;i <totaltrj;i++) {
31            closeID.add(-1);
32            closeDist.add(Double.MAX_VALUE);
33        }
34    }
35
36 }

```





## LAMPIRAN B

### HASIL EKSPERIMEN

Hasil eksperimen berikut dibuat dengan menggunakan TIKZPICTURE (bukan hasil excel yg diubah ke file bitmap). Sangat berguna jika ingin menampilkan tabel (yang kuantitasnya sangat banyak) yang datanya dihasilkan dari program komputer.



Gambar B.1: Hasil 1



Gambar B.2: Hasil 2



Gambar B.3: Hasil 3



Gambar B.4: Hasil 4