BAB II

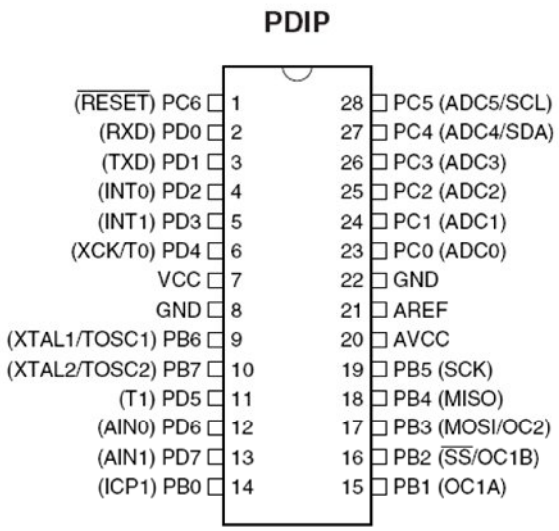
DASAR TEORI

2.1 Mikrokontroler AVR Atmega8

AVR merupakan salah satu jenis mikrokontroler yang di dalamnya terdapat berbagai macam fungsi. Perbedaannya pada mikro yang pada umumnya digunakan seperti MCS51 adalah pada AVR tidak perlu menggunakan *oscillator* eksternal karena di dalamnya sudah terdapat internal oscillator. Selain itu kelebihan dari AVR adalah memiliki *Power-On Reset*, yaitu tidak perlu ada tombol reset dari luar karena cukup hanya dengan mematikan *supply*, maka secara otomatis AVR akan melakukan *reset*. Untuk beberapa jenis AVR terdapat beberapa fungsi khusus seperti ADC, EEPROM sekitar 128 *byte* sampai dengan 512 byte.

AVR ATmega8 adalah mikrokontroler CMOS 8-*bit* berarsitektur AVR RISC yang memiliki 8K *byte in-System Programmable Flash*. Mikrokontroler dengan konsumsi daya rendah ini mampu mengeksekusi instruksi dengan kecepatan maksimum 16MIPS pada frekuensi 16MHz. Jika dibandingkan dengan ATmega8L perbedaannya hanya terletak pada besarnya tegangan yang diperlukan untuk bekerja. Untuk ATmega8 tipe L, mikrokontroler ini dapat bekerja dengan tegangan antara 2,7 - 5,5 V sedangkan untuk ATmega8 hanya dapat bekerja pada tegangan antara 4,5 – 5,5 V.

2.1.1 Konfigurasi Pin Atmega8



Gambar 2.1. Konfigurasi Pin Atmega8

ATmega8 memiliki 28 Pin, yang masing-masing pin nya memiliki fungsi yang berbeda-beda baik sebagai port maupun fungsi yang lainnya. Berikut akan dijelaskan fungsi dari masing-masing kaki ATmega8.

- VCC

  Merupakan *supply* tegangan digital.

- GND

  Merupakan ground untuk semua komponen yang membutuhkan grounding.

- Port B (PB7...PB0)

  Didalam Port B terdapat XTAL1, XTAL2, TOSC1, TOSC2. Jumlah Port B adalah 8 buah pin, mulai dari pin B.0 sampai dengan B.7. Tiap pin dapat digunakan sebagai *input* maupun *output*. Port B merupakan sebuah 8-bit *bi-directional* I/O dengan internal pull-up resistor. Sebagai *input,* pin-pin

yang terdapat pada port B yang secara eksternal diturunkan, maka akan mengeluarkan arus jika *pull-up* resistor diaktifkan. Khusus PB6 dapat digunakan sebagai *input* Kristal (*inverting oscillator amplifier*) dan *input* ke rangkaian *clock* internal, bergantung pada pengaturan *Fuse bit* yang digunakan untuk memilih sumber *clock*. Sedangkan untuk PB7 dapat digunakan sebagai *output* Kristal (*output oscillator amplifier*) bergantung pada pengaturan *Fuse bit* yang digunakan untuk memilih sumber *clock*. Jika sumber clock yang dipilih dari *oscillator internal*, PB7 dan PB6 dapat digunakan sebagai I/O atau jika menggunakan *Asyncronous Timer/Counter2* maka PB6 dan PB7 (TOSC2 dan TOSC1) digunakan untuk saluran *input timer*.

- Port C (PC5…PC0)

Port C merupakan sebuah 7-*bit bi-directional* I/O port yang di dalam masing-masing pin terdapat *pull-up* resistor. Jumlah pin nya hanya 7 buah mulai dari *pin* C.0 sampai dengan *pin* C.6. Sebagai keluaran/*output port* C memiliki karakteristik yang sama dalam hal menyerap arus (*sink*) ataupun mengeluarkan arus *(source*).

- RESET/PC6

Jika RSTDISBL *Fuse* diprogram, maka PC6 akan berfungsi sebagai *pin* I/O. *Pin* ini memiliki karakteristik yang berbeda dengan *pin-pin* yang terdapat pada *port* C lainnya. Namun jika RSTDISBL *Fuse* tidak diprogram, maka pin ini akan berfungsi sebagai input reset. Dan jika *level* tegangan yang masuk ke pin ini rendah dan pulsa yang ada lebih pendek dari pulsa

minimum, maka akan menghasilkan suatu kondisi reset meskipun *clock*-nya tidak bekerja.

- Port D (PD7…PD0)

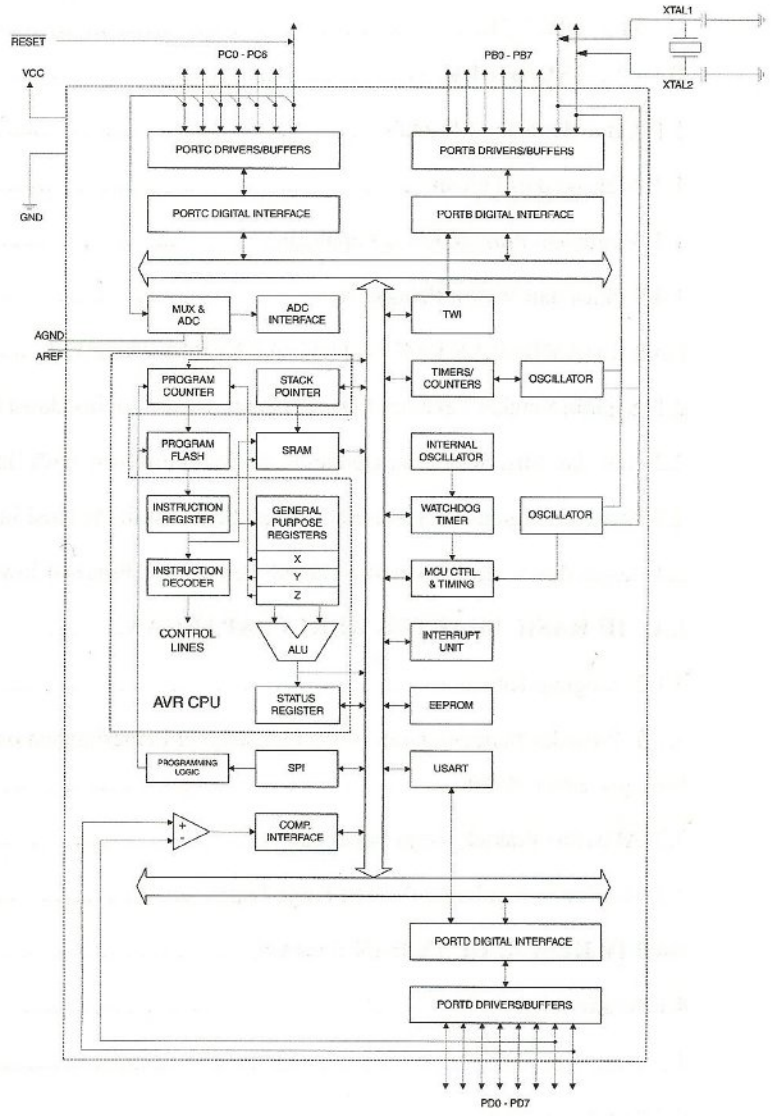
Port D merupakan 8-*bit bi-directional* I/O dengan internal *pull-up* resistor. Fungsi dari port ini sama dengan port-port yang lain. Hanya saja pada port ini tidak terdapat kegunaan-kegunaan yang lain. Pada *port* ini hanya berfungsi sebagai masukan dan keluaran saja atau biasa disebut dengan I/O.

- AVcc

Pin ini berfungsi sebagai *supply* tegangan untuk ADC. Untuk *pin* ini harus dihubungkan secara terpisah dengan VCC karena *pin* ini digunakan untuk analog saja. Bahkan jika ADC pada AVR tidak digunakan tetap saja disarankan untuk menghubungkannya secara terpisah dengan VCC. Jika ADC digunakan, maka AVcc harus dihubungkan ke VCC melalui *low pass filter*.

- AREF

Merupakan pin referensi jika menggunakan ADC.



Gambar 2.2 Blok Diagram ATmega8

Pada AVR status *register* mengandung beberapa informasi mengenai hasil dari kebanyakan hasil eksekusi instruksi aritmatik. Informasi ini digunakan untuk altering arus program sebagai kegunaan untuk meningkatkan performa pengoperasian. Register ini di-*update* setelah operasi ALU (*Arithmetic Logic Unit*) hal tersebut seperti yang tertulis dalam *datasheet* khususnya pada bagian *Instruction Set Reference.* Dalam hal ini untuk beberapa kasus dapat membuang

penggunaan kebutuhan instrukasi perbandingan yang telah didedikasikan serta dapat menghasilkan peningkatan dalam hal kecepatan dan kode yang lebih sederhana dan singkat. *Register* ini tidak secara otomatis tersimpan ketika memasuki sebuah rutin interupsi dan juga ketika menjalankan sebuah perintah setelah kembali dari interupsi. Namun hal tersebut harus dilakukan melalui *software*. Berikut adalah gambar status *register*.

Bit	7	6	5	4	3	2	1	0	
	I	T	H	S	V	N	Z	C	SREG
Read/write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Gambar 2.3 Status Register ATMega8

- Bit 7(I)

  Merupakan *bit Global Interrupt Enable*. *Bit* ini harus di-set agar semua perintah interupsi dapat dijalankan. Untuk perintah interupsi individual akan di jelaskan pada bagian yang lain. Jika bit ini di-*reset*, maka semua perintah interupsi baik yang individual maupun yang secara umum akan di abaikan. Bit ini akan dibersihkan atau *cleared* oleh *hardware* setelah sebuah interupsi di jalankan dan akan di-*set* kembali oleh perintah RETI. *Bit* ini juga dapat di-set dan di-*reset* melalui aplikasi dan intruksi SEI dan CLL.

- Bit 6(T)

  Merupakan *bit Copy Storage*. Instruksi bit *Copy Instructions* BLD (*Bit Load*) and BST (*Bit Store*) menggunakan *bit* ini sebagai asal atau tujuan untuk *bit* yang telah dioperasikan. Sebuah *bit* dari sebuah *register* dalam *Register File*

dapat disalin ke dalam bit ini dengan menggunakan instruksi BST, dan sebuah *bit* di dalam bit ini dapat disalin ke dalam *bit* di dalam *register* pada *Register File* dengan menggunakan perintah BLD.

- Bit 5(H)

Merupakan *bit Half Carry Flag*. *Bit* ini menandakan sebuah *Half Carry* dalam beberapa operasi aritmatika. *Bit* ini berfungsi dalam aritmatika BCD.

- Bit 4(S)

Merupakan *Sign bit*. Bit ini selalu merupakan sebuah ekslusif di antara *Negative Flag* (N) dan *two's Complement Overflow Flag (V).*

- Bit 3(V)

Merupakan *bit Two's Complement Overflow Flag*. Bit ini menyediakan fungsi aritmatika dua komplemen.

- Bit 2(N)

Merupakan *bit Negative Flag*. Bit ini mengindikasikan sebuah hasil *negative* di dalam sebuah fungsi logika atai aritmatika.

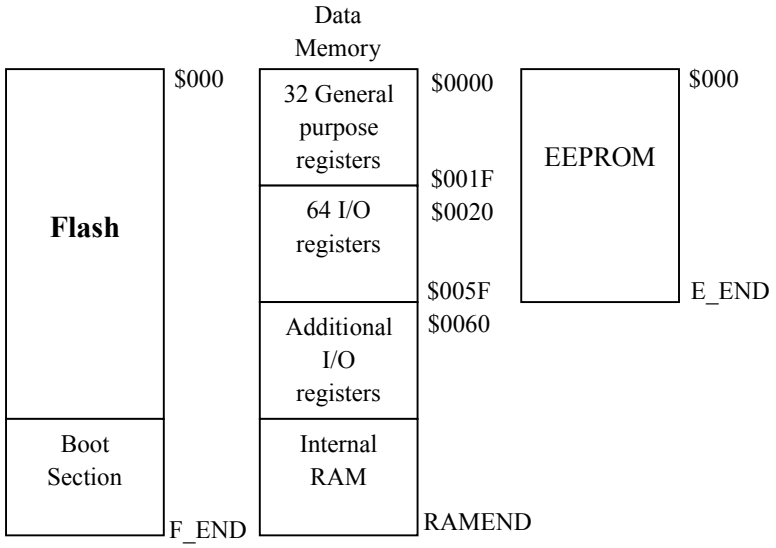
- Bit 1(Z)

Merupakan *bit Zero Flag*. Bit ini mengindikasikan sebuah jasil nol "0" dalan sebuah fungsi aritmatika atau logika.

- Bit 0(C)

Merupakan *bit Carry Flag*. Bit ini mengindikasikan sebuah Carry atau sisa dalam sebuah aritmatika atau logika.

2.1.2 Memori AVR Atmega



Gambar 2.4 Peta Memori Atmega

Memori atmega terbagi menjadi tiga yaitu :

1. Memori *Flash*

   Memori flash adalah memori ROM tempat kode-kode program berada. Kata flash menunjukan jenis ROM yng dapat ditulis dan dihapus secara elektrik. Memori flash terbagi menjadi dua bagian yaitu bagian aplikasi dan bagian *boot*. Bagian aplikasi adalah bagian kode-kode program apikasi berada. Bagian *boot* adalah bagian yang digunakan khusus untuk *booting* awal yang dapat diprogram untuk menulis bagian aplikasi tanpa melalui *programmer/downloader*, misalnya melalui USART.

2. Memori Data

   Memori data adalah memori RAM yang digunakan untuk keperluan program. Memori data terbagi menjadi empat bagian yaitu :

   32 GPR (*General Purphose Register*) adalah register khusus yang bertugas untuk membantu eksekusi program oleh ALU (*Arithmatich Logic Unit*), dalam instruksi assembler setiap instruksi harus melibatkan GPR. Dalam bahasa C biasanya digunakan untuk variabel global atau nilai balik fungsi dan nilai-nilai yang dapat memperingan kerja ALU. Dalam istilah processor komputer sahari-hari GPR dikenal sebagai *"chace memory"*.

   I/O register dan Aditional I/O register adalah *register* yang difungsikan khusus untuk mengendalikan berbagai pheripheral dalam mikrokontroler seperti *pin port, timer/counter, usart* dan lain-lain. Register ini dalam keluarga mikrokontrol MCS51 dikenal sebagi SFR(*Special Function Register*).

3. EEPROM

   EEPROM adalah memori data yang dapat mengendap ketika chip mati (*off*), digunakan untuk keperluan penyimpanan data yang tahan terhadap gangguan catu daya.

2.1.3 *Timer/Counter* 0

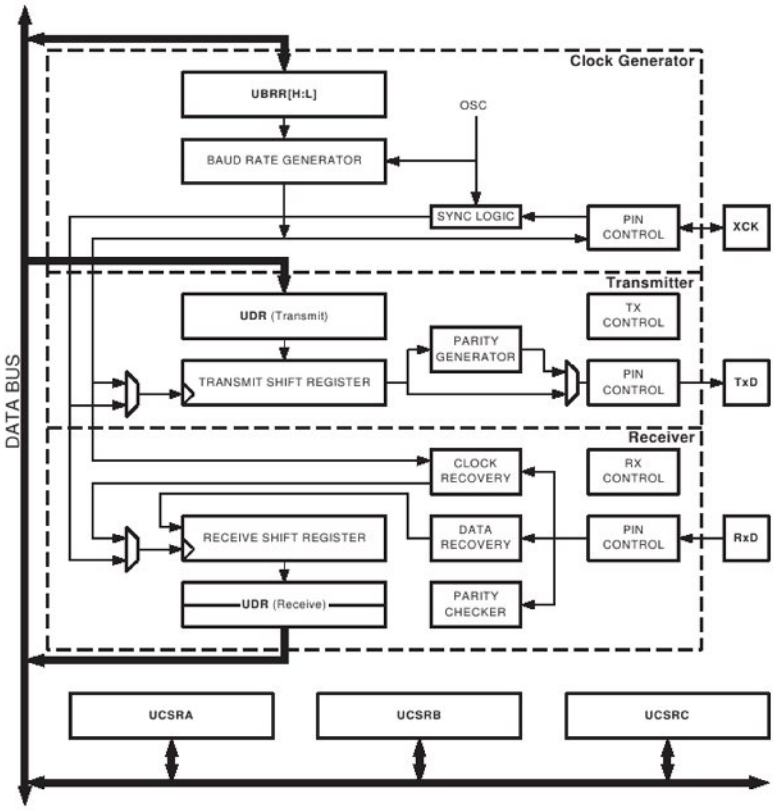
Timer/counter 0 adalah sebuah *timer/counter* yang dapat mencacah sumber pulsa/*clock* baik dari dalam *chip* (*timer*) ataupun dari luar *chip* (*counter*) dengan kapasitas 8-*bit* atau 256 cacahan.

Timer/counter dapat digunakan untuk :

1. *Timer/counter* biasa
2. *Clear Timer on Compare Match* (selain Atmega 8)
3. *Generator* frekuensi (selain Atmega 8)
4. *Counter* pulsa *eksternal*

2.1.4 Komunikasi Serial Pada Atmega 8

Mikrokontroler AVR Atmega 8 memiliki *Port* USART pada Pin 2 dan Pin 3 untuk melakukan komunikasi data antara mikrokontroler dengan mikrokontroler ataupun mikrokontroler dengan komputer. USART dapat difungsikan sebagai transmisi data sinkron, dan asinkron. Sinkron berarti *clock* yang digunakan antara *transmiter* dan *receiver* satu sumber *clock*. Sedangkan asinkron berarti transmiter dab receiver mempunyai sumber clock sendiri-sendiri. USART terdiri dalm tiga blok yaitu *clock generator*, *transmiter*, dan *receiver*.



Gambar 2.5 Blok USART

2.1.4.1 Clock Generator

Clock generator berhubungan dengan kecepatan transfer data (baud rate), register yang bertugas menentukan *baud rate* adalah register pasangan

Tabel 2.1 *Baud Rate*

Operating Mode	Equation for Calculating Baud Rate	Equaltion for Calculating UBRR Value
Asynchronous Normal Mode (U2X = 0)	$= \frac{}{16(\quad + 1)} - 1$	$= \frac{}{16} - 1$
Asynchronous Double Speed Mode (U2X=1)	$= \frac{}{8(\quad + 1)} - 1$	$= \frac{}{8} - 1$
Synchronous Master Mode	$= \frac{}{2(\quad + 1)} - 1$	$= \frac{}{2} - 1$

Dimana :

Fosc adalah frekuensi *ossilator* yang digunakan

BAUD adalah transfer *bit* per detik

2.1.4.2 USART *transmiter*

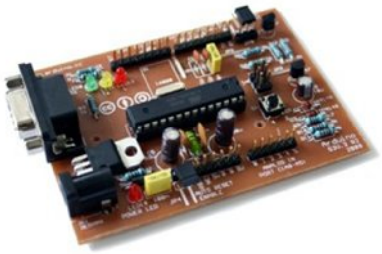
Usart transmiter berhubungan dengan data pada Pin TX. Perangkat yang sering digunakan seperti register UDR sebagi tempat penampungan data yang akan ditransmisikan. *Flag* TXC sebagai akibat dari data yang ditransmisikan telah sukses (*complete*), dan *flag* UDRE sebagai indikator jika UDR kosong dan siap untuk diisi data yang akan ditransmisikan lagi.

2.1.4.3 USART *receiver*

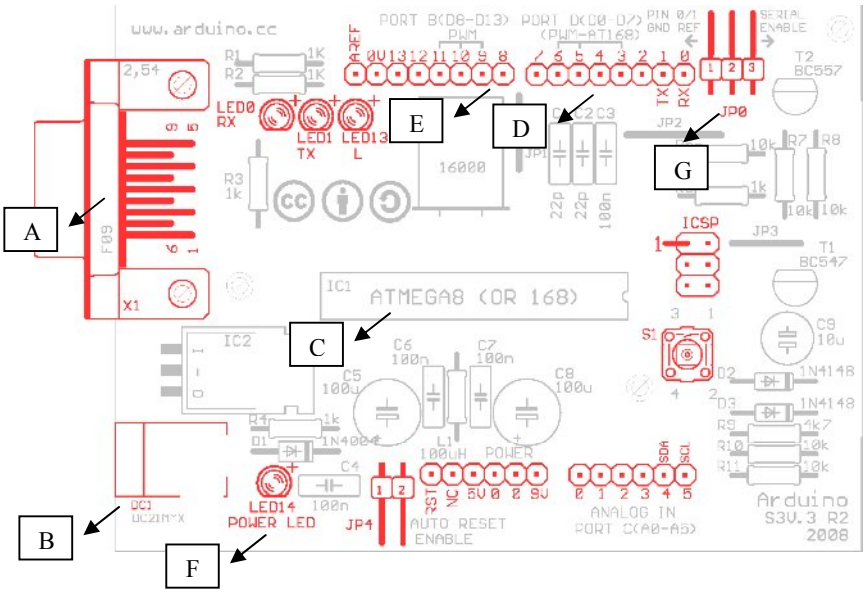
Usart receiver berhubungan dengan penerimaan data dari Pin RX. Perangkat yang sering digunakan seperti register UDR sebagai tempat penampung data yang telah diterima, dan *flag* RXC sebagi indikator bahwa data telah sukses (*complete*) diterima.

2.1.5 Arduino Severino Atmega8

Arduino adalah solusi terpaket yang terdiri dari board hardware dengan dengan mikrokontroler AVR serta *software* yang dikembangkan menggunakan bahasa Java dan menggunakan bahasa C sebagai rujukan dalam pemrograman. Arduino severino adalah board Arduino *single sided* versi lama dengan koneksi Serial RS232. Versi terakhir (S3V3) dirancang oleh Adilson Akashi, dengan penyempurnaan di sejumlah bagian, baik rangkaian maupun layout PCB.



Gambar 2.6 Arduino Severino ATMega8



Gambar 2.7 Bagian-Bagian Arduino yang digunakan

Keterangan :

- A Koneksi Serial RS232 yang digunakan untuk menghubungkan MCU dengan Komputer
- B Power jack dengan tegangan DC sebesar +9v
- C Chip ATMega8

- D Pin-pin digital input/output dengan Pin 0 sebagai Rx dan Pin 1 sebagai Tx
- E Pin-pin digital *input/output* 8 – 13
- F LED *Power*
- G Pin *jumper*, pada posisi 2-3, maka serial dapat digunakan (*serial enable*), selanjutnya pada posisi 1-2, akan menonaktifkan komunikasi serial dan mengaktifkan *external pull-down* resistor pada pin 0 (Rx) dan pin 1 (Tx).

2.1.6 Struktur Pemrograman Arduino

Struktur dasar bahasa pemrograman arduino hanya terdiri dari dua bagian

```
Void setup()
    {
        // Statement;     } di eksekusi satu kali
    }
Void loop()
    {
        // Statement;     } di eksekusi terus menerus
    }
```

1. Setup()

Fungsi setup() hanya dipanggil satu kali ketika program pertama kali di jalankan. Ini digunakan untuk mendifinisikan mode pin atu memulai komunikasi serial. Fungsi setup() harus di ikut sertakan dalam program walaupun tidak ada statement yang di jalankan.

```
void setup()
    {
      pinMode(3,OUTPUT);// men-set “pin” 3 sebagai Output
      pinMode(6, INPUT); // men-set pin  6 sebagai Input
      Serial.begin(9600);
    }
```

- **pinMode()** = berfungsi untuk mengatur fungsi sebuah pin sebagai INPUT atau OUTPUT.
- **Serial**.begin(9600) = digunakan untuk mengaktifkan fitur UART dan menginisialisasinya.

2. Loop()

Setelah melakukan fungsi setup() maka secara langsung akan melakukan fungsi loop() secara berurutan dan melakukan instruksi-instruksi ayang ada dalam fungsi loop().

```
void loop()
    {
      If (digitalRead(6)==HIGH)// membaca input digital pin 6
      {
            xstart = millis();      //aktifkan timer
            digitalWrite (3, HIGH); // nyalakan pin 3
            delay(1000);            // pause selama 1 detik
            digitalWrite(3, LOW);   // matikan pin 3
      }
    }
```

- **digitalWrite()** = Berfungsi untuk memberikan nilai **LOW** atau **HIGH** pada sebuah pin **OUTPUT**

- Fungsi **delay** = Berfungsi untuk memberikan jeda dalam satuan milidetik
- **digitalRead()** = Berfungsi untuk membaca nilai digital **LOW** atau **HIGH** dari sebuah pin **INPUT**
- Fungsi **milis()** akan memberikan posisi "Stopwatch" terakhir yang bekerja dalam system

Stopwatch bekerja dalam satuan milidetik, menggunakan **TIMER0** dan mencatatnya dalam variabel *unsigned long timer0_overflow_count*. Hitungan akan *overflow* setelah mencapai 4,294,967,295 (2^32-1)

2.2 Jarak

Jarak merupakan panjang lintasan yang ditempuh oleh suatu benda dalam selang waktu tertentu. Jarak juga bisa menyatakan posisi suatu benda terhadap titik acuan tertentu. Jarak termasuk besaran skalar, dimana tidak bergantung pada arah dan nilainya selalu positif.

2.3 Perpindahan

Berbeda dengan jarak, perpindahan adalah besaran vektor sehingga arah juga berpengaruh.

2.4 Kelajuan (V)

Dalam ilmu fisika, istilah laju/kelajuan menyatakan seberapa jauh sebuah benda berpindah selama selang waktu tertentu. Kaluan merupakan salah satu

besaran turunan yang tidak bergantung pada arah, sehingga kelajuan termasuk skalar.

2.4.1 Kecepatan Rata-Rata

Kecepatan rata-rata suatu benda yang bergerak didefinisikan sebagai perpindahan yang ditempuh benda dibagi waktu tempuh.

$$- \quad = \frac{h}{h}$$

$$= \frac{-}{-}$$

$$= \frac{\Delta}{\Delta}$$

2.4.2 Kecepatan Sesaat

Kecepatan sesaat adalah kecepatan rata-rata selama selang waktu yang sangat kecil. Secara matematis, kecepatan sesaat adalah perbandingan perpindahan dengan selang waktu, jika selang waktu mendekati nol. Secara matematis, kecepatan sesaat adalah limit perbandingan perpindahan dengan selang waktu jika selang waktu mendekati nol.

$$\lim_{\Delta \to} \frac{\Delta}{\Delta}$$

Limit ini dinamakan turunan x terhadap t, dimana dalam kondisi kalkulus (*diferensial/integral*) ditulis *dx/dt* :

$$= \lim_{\rightarrow} \text{—} - \text{—}$$

$$= \text{—}$$

2.4.3 Percepatan Rata-Rata

Percepatan rata-rata didefinisikan sebagai perbandingan perubahan kecepatan dengan selang waktu yang diperlukan untuk berubah tersebut. Secara metematis ditulis :

$$- \quad = \frac{h}{h}$$

$$- \quad = \frac{2 - 1}{2 - 1}$$

$$\bar{a} = \frac{\Delta}{\Delta}$$

Percepatan merupakan besaran vektor, tetapi untuk gerakan satu dimensi hanya perlu menggunakan tanda (+) dan (-) untuk menunjukkan arah sistem koordinat yang dipakai.

2.4.4 Percepatan Sesaat

Percepatan sesaat adalah limit perbandingan perubahan kecepatan dengan selang waktu perubahan, dengan selang waktu mendekati nol. Jika digambarkan grafik kecepatan dengan waktu, kecepatan sesaat pada t didefinisikan sebagai kemiringan garis yang menyinggung kurva pada saat itu.

$$= \lim_{\Delta \rightarrow} \frac{\Delta}{\Delta}$$

Percepatan merupakan turunan kecepatan terhadap waktu, dan biasa ditulis dengan notasi *dx/dt.*

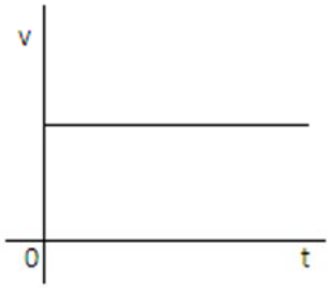
$$= \lim_{\Delta \to} \frac{\Delta}{\Delta} = —$$

Karena kecepatan merupakan turunan dari posisi terhadap t, maka percepatan merupakan turunan kedua x terhadap t, secara matematis ditulis :

$$= — = \frac{(—)}{——} = ——$$

2.5 Gerak Lurus Beraturan (GLB)

Suatu benda dikatakan melakukan gerak lurus beraturan jika kecepatan selalu konstan. Kecepatan konstan artinya besar kecepatan/kelajuan dan arah kecepatan selalu konstan. Karena besar kecepatan/kelajuan dan arah kecepatan selalu konstan maka bisa dikatakan bahwa benda bergerak dengan lintasan lurus dengan kelajuan konstan.



Gambar 2.8 Grafik Kecepatan Terhadap Waktu

Berdasarkan grafik diatas, tampak bahwa besar kecepatan bernilai tetap pada setiap satuan waktu. Besar kecepatan tetap ditandai oleh garis lurus, berawal dari t = 0 hingga t akhir.

$$- \quad = \frac{h}{h}$$

$$= \frac{2 - 1}{2 - 1}$$

$$\nabla \frac{\Delta}{\Delta}$$

2.6 Java

2.6.1 Sejarah Java

Sejarah Java berawal pada tahun 1991 ketika perusahaan Sun Microsystem memulai *Green Project*, yakni proyek penelitian untuk membuat bahasa yang akan digunakan pada *chip-chip embedded* untuk *device intellegent consumer electronic*. Bahasa tersebut haruslah bersifat multiplatform, tidak tergantung kepada vendor yang memanufaktur chip tersebut.

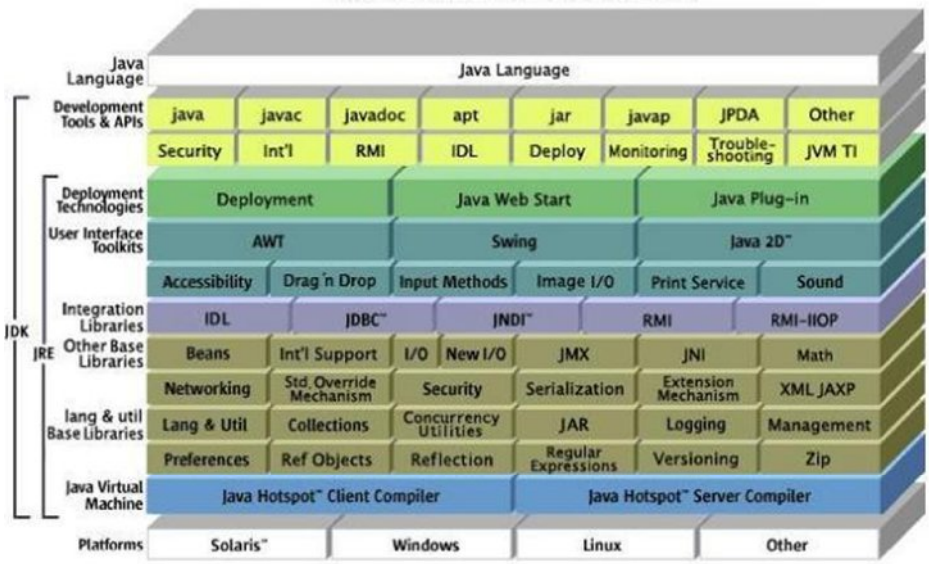
Dalam penelitian, Projek Green berhasil membuat prototype semacam PDA (Personal Data Assistance) yang dapat berkomunikasi antara satu dengan yang lain dan diberi nama Star 7. Ide berawal untuk membuat sistem operasi bagi Star 7 berbasis C dan C++. Setelah berjalan beberapa lama, James Gosling, salah satu seorang anggota team, merasa kurang puas dengan beberapa karakteristik dari kedua bahasa tersebut kemudian dinamakan Oak. Diinspirasi ketika dia melihat pohon di seberang kaca ruang kantornya. Belakangan Oak beralih nama menjadi *java*.

Karena pada awalnya ditunjuk untuk pemrograman *device* kecil, j*ava* memiliki karakteristik berukuran kecil, efisien, dan *portable* untuk berbagai *hardware*. Projek Green sempat terancam terhenti karena dalam perkembangannya, device ini belum memiliki pasar seperti yang diramalkan

semula. Selanjutnya *java* diarahkan untuk pemrograman internet. Secara kebetulan fitur-fitur *java* yang telah disebutkan sebelumnya sangat sesuai bagi pengembangan internet sehingga dalam beberapa tahun belakangan ini *java* telah menjadi primadona untuk pemrograman yang berbasis internet.

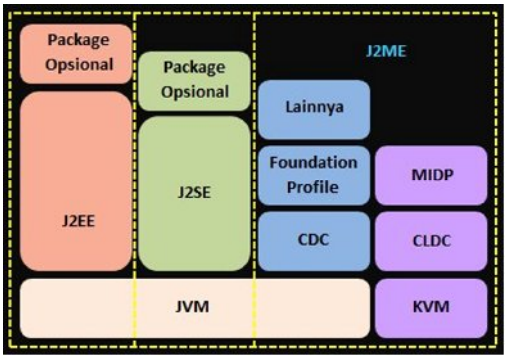
2.6.2 Definisi Java

Java menurut definisi dari Sun adalah nama untuk sekumpulan teknologi untuk membuat dan menjalankan perangkat lunak pada komputer *standalone* ataupun pada lingkungan jaringan. Java2 adalah generasi kedua dari *java platform* (generasi awalnya adalah *Java Devlopment Kit*). Java berdiri di atas sebuah mesin interpreter yang diberi nama *Java Virtual Machine* (JVM). JVM inilah yang akan membaca *bytecode* dalam *file .class* dari suatu program sebagai representasi program yang berisi bahasa mesin. Oleh karena itu, bahasa java di sebut sebagai bahasa pemrograman yang *portable* karena dapat di jalankan pada semua system operasi, asalkan pada system operasi tersebut terdapat JVM. *Platform* java terdiri dari kumpulan *library*, JVM, kelas-kelas *loader* yang dipaket dalam sebuah lingkungan rutin java, dan sebuah *compiler*, *debugger* dan kakas lain yang dipaket dalam *Java Development Kit* (JDK). Java2 adalah generasi yang sekarang sedang berkembang dari platform java. Agar sebuah program java di jalankan, maka file dengan ektensi .java harus di kompilasi menjadi file *bytecode.* Untuk menjalankan *bytecode* tersebut dibutuhkan JRE (*Java Runtime Envirotment*) yang memungkinkan pemakai untuk menjalankan program java, hanya menjalankan, tidak untuk membuat kode baru lagi. JRE berisi JVM dan *library* java yang digunakan, untuk lebih jelasnya dapat dilihat pada gambar 2.3 dibawah ini.



Gambar 2.9 J2SE 5.0

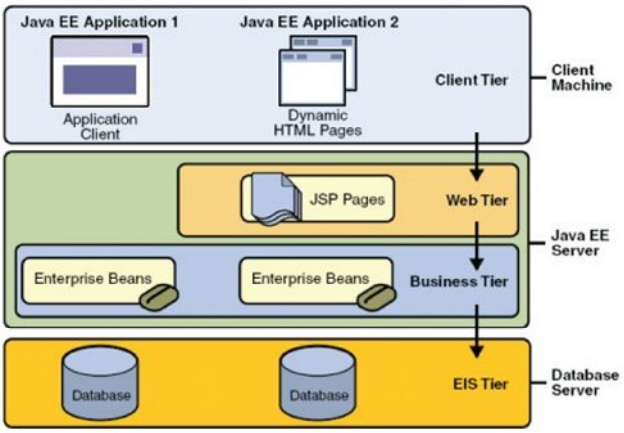
Platform java memiliki tiga buah edisi yang berbeda, yaitu J2EE (*Java Enterprise Edition),* J2ME (*Java2 Mikro Edition*) dan J2SE (*Java2 Scond Edition*). J2EE adalah kelompok dari beberapa API dari Java dan teknologi selain Java. J2EE dibuat untuk membuat aplikasi yang kompleks. J2EE sering dianggap sebagai *middleware* atau teknologi yang berjalan di *server*, namun sebenarnya J2EE tidak hanya terbatas untuk itu. Faktanya J2EE juga mencakup teknologi yang dapat digunakan di semua lapisan dari sebuah sistem informasi. Implementasi J2EE menyediakan kelas dasar dan API dari Java yang mendukung pengembangan dan rutin standar untuk aplikasi klien maupun *server*, termasuk aplikasi yang berjalan di *web browser.* J2SE adalah lingkungan dasar dari Java. Ruang lingkup keterhubungan J2EE, J2SE dan J2ME dapat di ihat pada gambar 2.4 dibawah ini.



Gambar 2.10 Ruang Lingkup Keterhubungan J2EE, J2SE dan J2ME

2.6.3 Java2 Enterprise Edition (J2EE)

J2ee (Java 2 Enterprise Edition) adalah sebuah solusi java untuk sebuah software yang ditunjukan untuk sebuah perusahaan dengan skala cukup besar. Hal-hal yang dibutuhkan dalam skala *interprise* sangat berbeda dengan hal-hal yang dibutuhkan oleh software biasa. J2ee mengandung lebih banyak API (*Application Programming Interface*) serta dipergunakan dalam arsitektur yang lebih besar daripada J2SE. Dari segi arsitektur n-*tier*, dimana *software* tersebut dibagi menjadi beberapa tingkatan/tier yang terpisah tergantung kegunaanya.



Gambar 2.11 Arsitektur J2EE

Dari gambar diatas terlihat bahwa J2EE mencakup 3 bagian utama yaitu : client, java ee server, dan database.

- *User Interface*

  Bagian ini adalah bagian yang berinteraksi langsung dengan user. Java menyediakan 2 solusi untuk bagian ini, yang pertama dengan menggunakan aplikasi java yang ditaruh di client dan bisa terhubung ke server atau menggunakan halaman web HMTL dinamik.

- Java EE Server

  Pada bagian ini dibagi lagi menjadi 2 bagian, yaitu web tier dan business tier. Pada web tier terdapat aplikasi java JSP (*Java Server Pages*). Web tier ini lebih ke arah web dinamik (dapat menggantikan web HMTL dinamik dikomputer *client*). Terdapat juga business *tier*, yaitu sebuah abstraksi dari kegiatan-kegiatan yang ada dalam proses bisnis diperusahaan. Di bagian ini terdapat program java seperti *beans*, dan *servlet,* dan masing-masing dari program tersebut adalah sebuah fungsi dari proses bisnis. Ha ini sesuai dengan fungsi j2ee seagai sebuah IT sistem yang mensupport bisnis perusahaan.

- *Database Server*

  Dibagian ini adalah tempat data-data perusahaan disimpan. Bisa berupa database seperti oracle atau merupakan EIS (*Enterprise Information System*) internal perusahaan. Java disini lebih ke arah menyediakan konektivitas dari aplikasi java kepada database atau sistem internal perusahaan.

2.6.4 *Java2 Standard Edition*

J2SE adalah inti dari bahasa pemrograman Java. JDK (*Java Devlopment Kit*) adalah salah satu *tool* dari J2SE untuk mengkompilasi dan menjalankan program Java. Di dalamnya terdapat *tool* untuk mengkompilasi program Java dan JRE. Tool J2SE yang salah satunya adalah jdk1.5 merupakan *tool open source* dari Sun.

Java 2 Standard Edition (J2SE), digunakan untuk mengembangkan aplikasi-aplikasi *desktop* dan *Applet* (aplikasi java yang dapat dijalankan di dalan *browser* Web). Pada pemrograman Java terdapat kelebihan lain yang tidak dimiliki bahasa pemrograman lainnya yaitu *Applet. Java Applet* adalah sebuah program kecil yang dibuat dengan menggunakan bahasa perograman Java, yang diakses melalui halaman Web dan dapat *di-download* ke dalam mesin *Client* yang kemudian menjalankannya didalam jendela web-browser. Java applet dapat secara dinamis menambahkan beberapa fungsi kepada halaman-halaman *web* yang bersifat statis. Akan tetapi, untuk menjalankannya, sebuah komputer harus memiliki program penjelajah web yang dapat menjalankan Java, seperti *Microsoft Internet Explore*, *Netscape Navigator, Mozilla Firefox,* dan *Opera.*

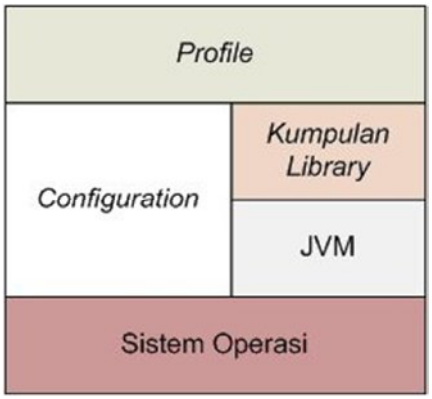
Ketika sebuah *Java Applet* dibuat, semua pernyataan Java yang terkandung di dalam kode sumbernya akan di kompilasi menjadi bytecode, yakni sebuah bahasa mesin semu (*virtual engine/machine laguage*) yang dibentuk oleh Java. Berkas yang berisi Java bytecode ini akan disimpan sebagai sebuah berkas kelas Java (Java class file) didalam sebuah *Web server*, seperti halnya *Apache HTTP Server* atau *Microsoft Internet Information Service* (IIS). Sebuah halaman Web yang hendak menggunakan Applet tersebut harus menggunakan *tag*

<APPLET>...</APPLET> di dalam kode sumbernya. Ketika sebuah penjelajah Web milik klien melakukan request kepada halaman Web tersebut menemukan bahwa di dalamnya terdapat *tag* <APPLET>...</APPLET>, *bytecode* di dalam *java class file* akan dieksekusi oleh mesin semu di dalam jendela penjelajah Web, yang dapat berupa *Microsoft Java Virtual Machine* atau *Java Runtime Engine*.

2.6.5 *Java2 Micro Edition* (J2ME)

Java2 Micro Edition atau yang sering disebut J2ME adalah lingkungan pengembangan yang telah didisain untuk meletakkan perangkat lunak Java pada barang elektronik beserta perangkat pendukungnya. Pada J2ME, jika perangkat lunak berfungsi baik pada sebuah perangkat maka belum tentu juga berfungsi baik pada perangkat yang lainnya. J2ME membawa Java ke dunia informasi, komunikasi, dan perangkat komputasi selain perangkat komputer *desktop*. J2ME biasa digunakan pada telepone *seluler, pager*, *personal digital assistants* (PDA's) dan sejenisnya.

J2ME adalah bagian dari J2SE, karena itu tidak semua *library* yang ada pada J2SE dapat digunakan pada J2ME. Tetapi J2ME mempunyai beberapa library khusus yang tidak dimiliki J2SE. Arsitektur J2ME dapat dilihat pada gambar 2.5 dibawah ini.

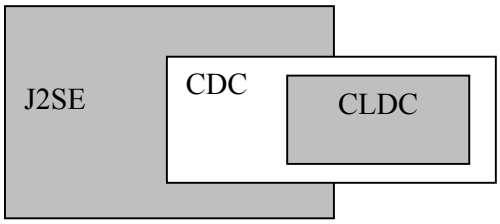


Gambar 2.12 Arsitektur J2ME

Teknologi J2ME juga memiliki keterbatasan, terutama jika diaplikasikan pada ponsel. J2ME sangat tergantung pada perangkat (*device*) yang digunakan, bisa dari segi merk ponsel, maupun kemampuan ponsel, dan dukungannya terhadap teknologi J2ME. Misalnya, jika sebuah ponsel tidak memiliki kamera maka jelas J2ME pada ponsel tersebut tidak dapat mengakses kamera. Keterbatasan lainnya adalah pada ukuran aplikasi, karena memori pada ponsel sangat terbatas. Sebagian besar ponsel tidak mengijinkan aplikasi J2ME menulis pada *file* karena alasan keamanan.

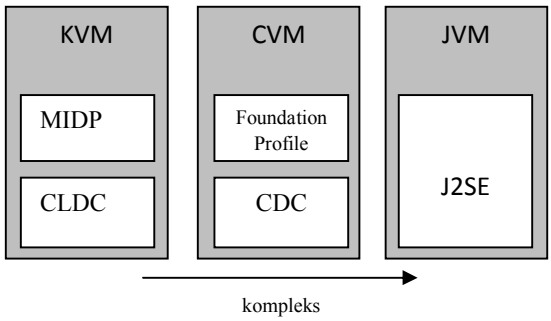
Configuration merupakan Java *library* minimum dan kapabilitas yang di punya oleh para pengembang J2ME, yang maksudnya sebuah *mobile device* dengan kemampuan Java akan dioptimalkan untuk menjadi sesuai. *Configuration* hanyalah mengatur hal-hal tentang kesamaan sehingga dapat dijadikan ukuran kesesuaian antar-*device.* Misalnya sebuah lampu sepeda dirancang sedemikan rupa sehingga didefinisikan dua buah konfigurasi yaitu CLDC (*Connected Limited Device Configuration*) untuk perangkat kecil dan CDC (*Connected*

Device Configuration) untuk perangkat yang lebih besar. Lingkup CLDC dan CDC dapat dilihat pada gambar 2.6 dibawah ini.

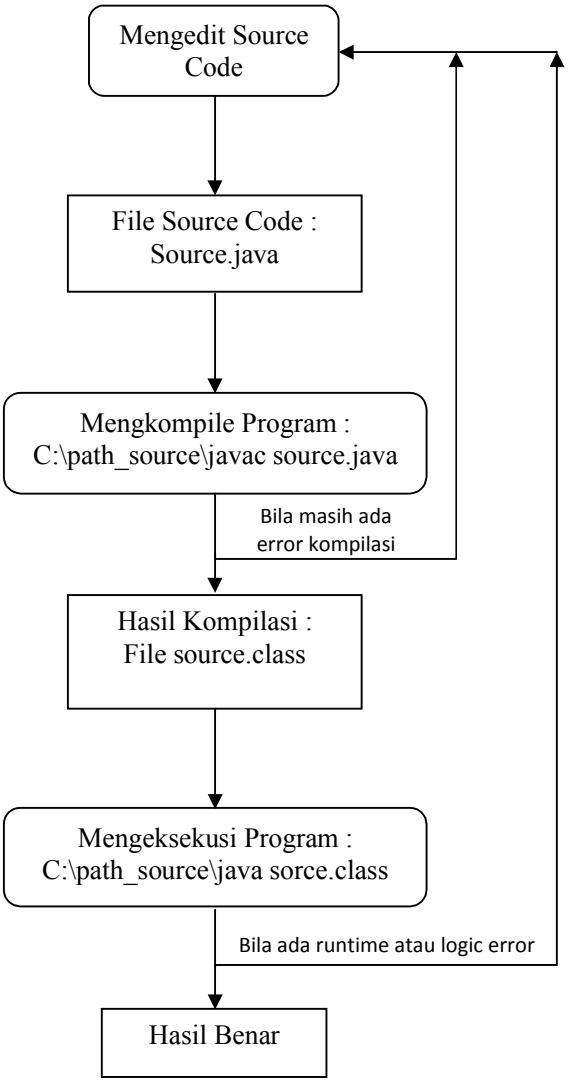


Gambar 2.13 Lingkup *Configuration*

Profile berbeda dengan configuration, profile membahas sesuatu yang spesifik untuk sebuah perangkat. Dalam J2ME terdapat dua buah profile yaitu MIDP dan Foundation Profile. Keterhubungan pada J2ME beserta mesin virtualnya dapat di lihat pada gambar 2.7 dibawah ini.



Gambar 2.14 Hubungan J2ME dan J2SE



Gambar 2.15 Diagram Alur Java

2.6.6 Struktur Pemrograman Java

```
package speedometer;
import gnu.io.*;
/**
*Komentar dalam java
*/
Modifier class namaKelas{
        //deklarasi variable
        Modifier methodConstructor(){
        //statement
        }
Public static void main(){
        /*
        *statement1
                *statement2
        */
}
```

- **Package**

Package atau paket adalah cara pengelompokan dan pengorganisasian class-class dalam satu library. Package bekerja dengan membuat directonari dan folder varu sesuai dengan penamaan package, kemudian menyimpan file class pada folder tersebut. Deklarasi package ditulis pada baris paling atas sebelum perintah import, sebagaimana terlihat pada struktur program java diatas. Deklarasi package adalah sbb:

*Package **speedometer;** // nama paketnya speedometer*

Package adalah perintah java yang digunakan untuk memberitahukan bahwa suatu class adalah anggota dari package, sedangkan namaPackage dapat berupa susunan direktori tempat dimana file class disimpan atau

nama folder. Penulisan nama package dalam java biasanya diawali dengan huruf kecil.

- **Import**

  Perintah import digunakan untuk memberitahukan kepada program untuk mengacu pada class-class yang terdapat pada *package* tersebut dan buka menjalankan class-class. Deklarasi :

```
Import namaKelas.namaKelas
Import gnu.io.*; // import kelas io
```

  Tanda arsitek (*) digunakan untuk mengimport semua *class.* Sedangkan untuk mengimpor *class* nama *class* dituliskan setelah nama *package.*

- **Komentar Dalam java**

  Komentar adalah catatan yang ditulis bersama kode program yang berguna sebagai catatan/keterangan dari satu kode, sehingga dapat digunakan sebagai referensi dalam pembuatan dokumentasi. Komentar merupakan bagian dari program yang tidak mempengaruhi jalanya program saat program dikompilasi atau dieksekusi.

  Dalam penuliasan komentar, *java* mengizinkan penulisan komentar versi java sendiri yang dapat digunakan sebagai pembuatan dokumentasi html. Selain itu *java* juga mengadopsi penuliasn versi C dan C++ akan tetapi dapat dibuat dokumentasi html.

  Penulisan komentar versi java diawali dengan /**. Didalam tanda tersebut anda dapat menuiskan lebih dari satu baris komentar. Contoh :

```
/**
*ini adalah contoh komentar dalam versi java
*menguasai pemrograman java
*/
```

- ***Identifier JAVA***

  Identifier adalah suatu tanda yang berupa huruf, latar, atau gabungan yang digunakan sebagai nama *variable, methode, class*, dan sebagainya.

- **Dekalarasi *class***

  Dalam deklarasi *class*, pertimbangan dimana dan bagaimana *class* tersebut digunakan (menentukan *modifier*). Kemudian berilah nama (*identifier*) yang sesuai dengan informasi yang dikandung dan dituliskan deklarasi *property* secara lengkap beserta *method* secara urut. Pilihlah *modifier* yang sesuai untuk menentukan hubungan dengan *class* lain.

```
(modifier_1) namaClass(modufier_2)
{
        <deklarasi variabel>
      <deklarasi konstruktor>
        <deklarasi methode>
}
```

- ***Modifier***

  Modifier adalah *keyword* java yang berfungsi mengatur hubungan antar class, methode, dan variable. Hubungan yang dimaksud yakni apakah suatu variable, method dan class dalam satu class dapat diakses oleh class dan methode lainnya.

- **Blok *Statement Java***

  Blok statement adalah himpunan pernyataan yang terdapat diantara kuraung kurawal buka dan kurawal tutup({....}). Blok sendiri terdapat 3 macam, yaitu :

  1. Blok *class*

     Blok class dimulai dari tanda "{" setelah indentifier dan di akhiri dengan tanda "}". Disini terlihat blok class sebagi hierarkhi tertinggi yang didalamnya terdapat data, blok methode.

  2. Blok methode

     Blok methode terdapat dalam blok *class* dan tidak dapat berdiri sendiri. Blok *methode* terdiri atas data dan struktur control.

  3. Blok kontrol

     Blok kontrol merupakan representasi struktur control.

- **Deklarasi *methode***

  Methode adalah bagian pemrograman yang menjelaskan tingkah laku dari objek yang akan diinstan. Methode tidak dapat berdiri sendiri sebagaimana *class*, dimana letak penulisan berada dalam badan class. Deklarasi *methode:*

```
(modifier) tipeKembalian namaMethode (parameter input)
Throw Exception {
  // Statement
}
```

2.6.7 Keunggulan Java

Java memiliki beberapa keunggulan bila dibandingkan dengan bahasa pemrograman lainnya. Antara lain :

1. **Java bersifat sederhana dan relatif mudah**

   Java dimodelkan sebagian dari bahasa C++, namun dengan memperbaiki beberapa karakteristik C++, seperti mengurangi kompleksitas beberapa fitur, penambahan fungsionalitas, serta penghilangan beberapa aspek pemicu ketidakstabilan sistem pada C++.

   Sebagai contoh, Java menggantikan konsep pewarisan lebih dari satu (multiple inheritance) dengan *interface,* menghilangkan konsep pointer yang sering membingungkan, otomatisasi sistem alokasi memory, dan sebagainya. Ini membuat Java relatif sederhana dan mudah untuk dipelajari dibandingkan bahasa pemrograman lainnya.

2. **Java berorientasi pada objek (Object Oriented)**

   Java adalah bahasa pemrograman yang berorientasi objek (OOP), bukan seperti Pascal, Basic atau C yang berbasis prosedural. Dalam memecahkan masalah, Java membagi program menjadi objek-objek,

kemudian memodelkan sifat dan tingkah laku masing-masing. Selanjutnya, Java menentukan dan mengatur interaksi antara objek yang satu dengan lainnya.

3. **Java bersifat terdistribusi**

   Pada dekade awal perkembangan PC (*Personal Computer*), komputer hanya bersifat workstation tunggal, tdiak terhubung satu sama lain. Saat ini, sistem komputerisasi cenderung terdistribusi, mulai dari *workstation client*, *e-mail server*, *database server*, *web server, proxy server,* dan sebagainya.

4. **Java bersifat *Multiplatform***

   Java bersifat multiplatform, yakni dapat di-"terjemahkan" oleh Java Interpreter sebagi sistem operasi.

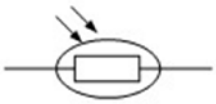
5. **Java bersifat *MultiThread***

   Thread adalah proses yang dapat dikerjakan oleh program dalam satu waktu. Java bersifat *Multithreaded*, artinya dapat mengerjakan beberapa proses dalam waktu yang hampir bersamaan.

2.7 Sensor Cahaya

2.7.1 LDR *(Light Dependent Resistor)*

LDR adalah suatu bentuk komponen yang mempunyai perubahan resistansi yang besarnya tergantung pada cahaya. Resistor peka cahaya (Light Dependent Resistor) memanfaatkan bahan semikonduktor yang karakteristik listriknya berubah-ubah sesuai dengan cahaya yang diterima. Bahan yang digunakan adalah Kadmium Sulfida (CdS) dan Kadmuim Selenida (CdSe). Bahan-bahan ini sangat sensitif terhadap cahaya dalam spektrum tampak, dengan puncaknya sekitar 0,6 μm untuk CdS dan 0,75 μm untuk CdSe. Sebuah LDR CdS yang tipikal memiliki resistansi sekitar 1MΩ dalam kondisi gelap gulita dan kurang dari 1KΩ ketika ditempkan di bawah sumber cahaya terang.



Gambar 2.16 Lambang LDR

Karakteristik LDR terdiri dari dua macam yaitu Laju Recovery dan Respon Spektral :

- **Laju *Recovery***

  Bila sebuah LDR dibawa dari sebuah ruangan dengan level kekuatan cahaya tertentu ke dalam suatu ruangan yang gelap maka nilai resistansi dari LDR tidak akan segera berubah resistansinya pada geadaan ruangan gelap tersebut. Namun LDR tersebut hanya akan bisa mencapai harga di kegelapan setelah mengalami selang waktu tertentu. Laju Recovery

merupakan suatu ukuran praktis dan suatu kenaikan nilai resistansi dalam waktu tertentu. Harga ini ditulis dalam K/detik, untuk LDR tipe arus harganya lebih besar dari 200 K/detik (selama 20 menit pertama mulaidari level cahaya 100 lux), kecepatan tersebut akan lebih tinggi dari arah sebaliknya, yaitu pindah dari tempat gelap ke tempat terang yang memerlukan waktu kurang dari 10 ms untuk mencapai resistansi yang sesuai dengan level cahaya 400 lux.

- **Respon Spektral**

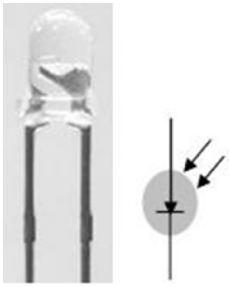
  LDR tidak memiliki sensitivitas yang sama untuk setiap panjang gelombang cahaya yang jatuh padanya (yaitu warna). Bahan yang biasa digunakan sebagai penghantar arus listrik yaitu tembaga, alumunium, baja, emas, perak. Dari kelima bahan tersebut tembaga merupakan yang paling banyak digunakan karena mempunyai daya hantar yang baik.

2.7.2 Photodioda

Photodioda adalah jenis dioda yang berfungsi untuk mendeteksi cahaya. Berbeda dengan dioda biasa, komponen elektronik ini akan mengubah cahaya menjadi arus listrik. Cahaya yang dapat di deteksi oleh dioda ini, mulai dari infra red, sinar ultra violet, sampai dengan sinar X.

Photodioda dibuat dari semikonduktor dengan bahan yang populer adalah silicon (Si) arau galium arsenida (GaAs), dan yang lain meliputi InSb, InAs, PbSe. Material ini menyerap cahay dengan karakteristik panjang gelombang mencakup : 2500 Å – 11000 Å untuk silicon, 8000 Å - 20000 Å untuk GaAs. Ketika sebuah

photon (satu satuan energi dalam cahaya) dari sumber cahaya diserap, hal tersebut membangkitkan suatu elektron dan menghasilkan sepasang pembawa muatan tunggal sebuah elektron dan sebuah hole, dimana suatu hole adalah bagian dari ksisi-kisi semikonduktor yang kehilangan elektron. Arah arus yang melalui sebuah semikonduktor adalah kebalikan dari gerakan muatan pembawa. Cara tersebut didalam sebuah photodiode digunakan untuk mengumpulkan photon - menyebabkan pembawa muatan (seperti arus atau tegangan) mengalir/terbentuk di bagian-bagian elektroda.



Gambar 2.17 Photodioda

2.7.3 *PhotoTrasnsistor*

Phototransistor adalah sebuah transistor yang apabila dikenai cahaya akan mengalirkan elektron sehingga akan terjadi penguatan arus seperti pada transistor.



Gambar 2.18 Photo Transistor

2.7.4 *Optocoupler*

Optocoupler merupakan salah satu jenis komponen yang memanfaatkan sinar sebagai pemicu on/off-nya. Opto berarti *optic* dan *coupler* berarti pemicu. Sehingga bisa diartikan bahwa optocoupler merupakan suatu komponen yang bekerja berdasarkan picu cahaya optik *opto-coupler* termasuk dalam sensor, dimana terdiri dari dua bagian yaitu *transmiter* dan *receiver.*



Gambar 2.19 Optocoupler

Bagian pemancar atau *transmiter* dibangun dari sebuah led infra merah untuk mendapatkan ketahanan yang lebih baik daripada menggunakan *led* biasa. Sensor ini bisa juga digunakan sebagai isolator dari rangkaian tegangan rendah ke rangkaian tegangan tinggi. Selain itu juga bisa digunakan sebagai pendeteksi adanya penghalang adanya *transmiter* dan *receiver* dengan memberi ruang uji dibagian tengah antara *led* dan *photo* transistor. Penggunaan ini bisa diterapkan untuk mendeteksi putaran motor atau mendeteksi lubang penanda disket pada disk *drive* komputer.

2.8 Bluetooth

2.8.1 Pengertian Bluetooth

Bluetooth adalah spesifikasi industri untuk kawasan pribadi (*Personal Area Network* atau PAN) tanpa kabel. Bluetooth menghubungkan dan dapat dipakai untuk tukar menukar informasi diantara peralatan-peralatan. Spesifikasi dari peralatan bluetooth dikembangkan dan didistribusikan oleh kelompok *Bluetooth Special Interest Group.* Bluetooth beroperasi dalam pita frekuensi 2,4 Ghz dengan menggunakan sebuah *frequency hopping traceiver* yang mampu menyediakan layanan komunikasi data dan suara secara *real time* antara *host-host* bluetooth dengan jarak terbatas. Kelemahan teknologi ini adalah jangkauanya yang pendek dan kemampuan transfer data yang rendah.

2.8.2 Sejarah Bluetooth

Awal mula dari bluetooth adalah sebagai teknologi komunikasi wireless (tanpa kabel) yang beroperasi dalam pita frekuensi 2,4 Ghz unlicensed ISM (*Industrial Scientific and Medical*) dengan menggunakan sebuah *frequency hopping traceiver* yang mampu menyediakan layanan komunikasi data dan suara secara *real time* antara *host-host* bluetooth dengan jarak jangkau layanan yang terbatas (sekitar 10 meter). Bluetooth berupa *card* yang menggunakan frekusensi radio standar IEEE 802.11 dengan jarak layanan terbatas dan kemampuan data transfer lebih rendah dari card untuk *Wireless Local Area Network* (WLAN).

Pembentukan bluetooth dipromotori oleh 5 perusahaan besar Ericsson, IBM, Intel, Nokia dan Toshiba membentuk sebuah *special Interest Group* (SIG)

yang meluncurkan proyek ini. Pada bulan Juli 1999 dokumen spesifikasi bleutooth versi 1.0 mulai diluncurkan. Pada bulan Desember 1999 dimulai lagi pembuatan dokumen spesifikasi bluetooth versi 2.0 dengan tambahan 4 promotor baru yaitu 3Com, *Lucent Technologies*, *Microsoft* dan *Motorolla*. Saat ini lebih dari 1800 perusahaan diberbagai bidang bergabung dalam sebuah konsorsium sebagi adopter teknologi bluetooth. Walaupun standar bluetooth SIG saat ini dimiliki oleh group promotor tetapi ia diharapkan akan menjadi standar IEEE (802.15)

2.8.3 Asal Nama Bluetooth dan Lambangnya

Nama bluetooth berasal dari nama raja di akhir abad sepuluh, Harald Blatand yang di Inggris juga dijuluki *Harald* Bluetooth kemungkinan karena memang giginya berwarna gelap. Ia adalah raja Denmark yang telah berhasil menyatukan suku-suku yang sebelumnya berperang, termasuk suku dari wilayah yang sekarang bernama Norwegia dan Swedia. Bahkan wilayah Scania di Swedia, tempat teknologi bluetooth ini ditemukan juga termasuk daerah kekuasaannya. Kemampuan raja itu sebagai pemersatu juga mirip dengan teknologi bluetooth sekarang yang bisa menghubungkan berbagai peralatan seperti komputer peronal dan telepon genggam. Sedangkan logo bluetooth berasal dari penyatuan huruf Jerman yang analoh dengan huruf H dan B (singkatan dari Harald Bluetooth), yaitu ᚼ (Hagall) dan ᛒ (Blatand) yang kemudian digabungkan.



Gambar 2.20 Logo Bluetooth

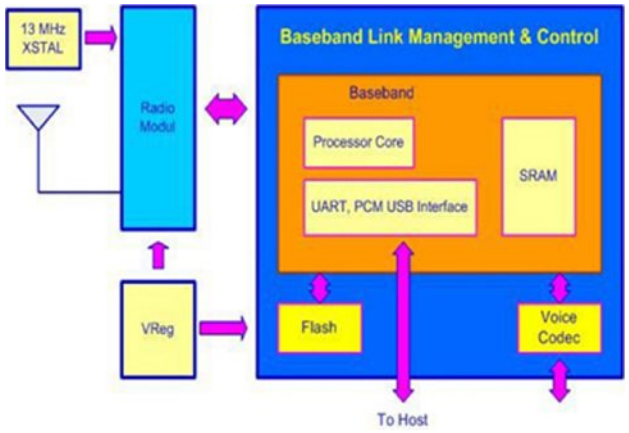
Tabel 2.2 Karakteristik Radio Bluetooth

Parameter	**Spesifikasi**
Frekuesnsi	ISM band, 2400-2483.5 MHz (mayoritas), untuk beberapa negara mempunyai batasan frekuensi sendiri, spasi kanal 1 MHz
Maksimum Output Power	Power class 1 : 100 mW (20 dBm) Power class 2 : 2.5 mW (4dBm) Power class 3 : 1 mW (0dBm)
Modulasi	GFSK(Gaussian Frequency Shift Keying), Bandwidth Time : 0.5; Modulation Index : 0.28 sampai dengan 0.35
Out of band Spurious Semission	30 MHz - 1 GHz : -36 dBm (operation mode), -57 dBm (idle mode)1 GHz – 12.75 GHz: -30 dBm (operation mode), -47 dBm (idle mode)1.8 GHz – 1.9 GHz: -47 dBm (operation mode), -47 dBm (idle mode)5.15 GHz –5.3 GHz: -47 dBm (operation mode), -47 dBm (idle mode)
Receiver	
Actual Sensitivity Level	-70 dBm pada BER 0,1%.
Spurious Emission	30 MHz - 1 GHz : -57 dBm1 GHz – 12.75 GHz : -47 dBm
Max. usable level	-20 dBm, BER : 0,1%

2.8.4 Cara Kerja Bluetooth

Sistem bluetooth terdiri dari sebuah ***radio transceiver, baseband link Management dan Control, Baseband (processor core, SRAM, UART, PCM USB Interface), flash*** dan ***voice codec***. Baseband link controller menghubungkan perangkat keras radio ke baseband processing dan layer protokol fisik. Link

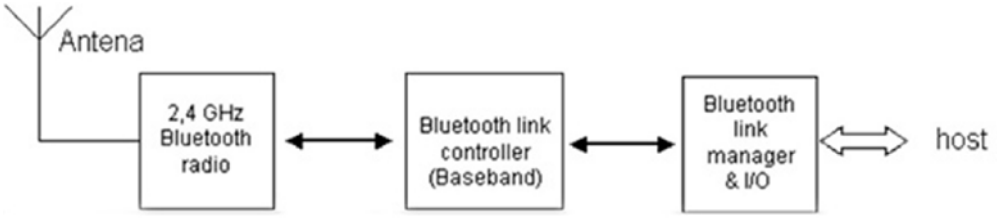
manager melakukan aktivitas-aktivitas protokol tingkat tinggi seperti melakukan link setup, autentikasi dan konfigurasi. Secara umum blok fungsional pada sistem bluetooth secara umum dapat dilihat pada Gambar 2.19 dibawah ini.



Gambar 2.21 Blok Fungsi Sistem Bluetooth

Protokol bluetooth menggunakan sebuah kombinasi antara circuit switching dan packet *switching*. Sebuah perangkat yang memiliki teknologi *wireless* akan mempunyai kemampuan untuk melakukan pertukaran informasi dengan jarak jangkauan sampai dengan 10 meter (~30 feet), bahkan untuk daya kelas 1 bisa sampai pada jarak 100 meter.

Bluetooth merupakan chip radio yang dimasukkan ke dalam komputer, printer, handphone dan peralatan lainnya. *Chip* bluetooth ini dirancang untuk menggantikan kabel. Informasi yang biasanya dibawa oleh kabel dengan Bluetooth ditransmisikan pada frekuensi tertentu kemudian diterima oleh chip Bluetooth kemudian informasi tersebut diterima oleh komputer, handphone dan peralatan lainnya.



Gambar 2.22 Proses Distribusi Aliran Data Dari Host sampai Antena pada Teknologi Bluetooth

Tiga buah lapisan fisik yang sangat penting dalam protokol arsitektur Bluetooth ini adalah :

1. ***Bluetooth radio***, adalah lapisan terendah dari spesifikasi Bluetooth. Lapisan ini mendefinisikan persyaratan yang harus dipenuhi oleh perangkat tranceiver yang beroperasi pada frekuensi 2,4 GHz ISM.
2. ***Baseband***, lapisan yang memungkinkan hubungan *Radio Frequency* (RF) terjadi antara beberapa unit Bluetooth membentuk piconet. Sistem RF dari bluetooth ini menggunakan *frekuensi-hopping-spread spectrum* yang mengirimkan data dalam bentuk paket pada time slot dan frekuensi yang telah ditentukan, lapisan ini melakukan prosedur pemeriksaan dan paging untuk sinkronisasi transmisi frekuensi *hopping* dan *clock* dari perangkat bluetooth yang berbeda.
3. ***LMP (Link Manager Protocol)****,* bertanggung jawab terhadap link set-up antar perangkat Bluetooth. Hal ini termasuk aspek security seperti autentifikasi dan enkripsi dengan pembangkitan, penukaran dan pemeriksaan ukuran paket dari lapis *baseband.*

2.8.5 Kelebihan dan Kekurangan

- Kelebihan
    1. Bluetooth dapat menembus dinding, kotak, dan berbagai rintangan lain walaupun jarak transmisinya hanya sekitar 30 kaki atau 10 meter
    2. Bluetooth tidak memerlukan kabel ataupun kawat
    3. Bluetooth dapat mensinkronisasi basis data dari telepon genggam ke komputer
    4. Dapat digunakan sebagai perantara modem
- Kekurangan
    1. Sistem ini menggunakan frekuensi yang sama dengan gelombang LAN standar
    2. Apabila dalam suatu ruangan terlalu banyak koneksi bleutooth yang digunakan, akan menyulitkan pengguna untuk menemukan penerima yang diharapkan
    3. Banyak mekanisme keamanan bleutooth yang harus diperhatikan untuk mencegah kegagalan pengiriman atau penerimaan informasi
    4. Di Indonesia, sudah banyak beredar virus-virus yang disebarkan melalui bluetooth dari handpone.