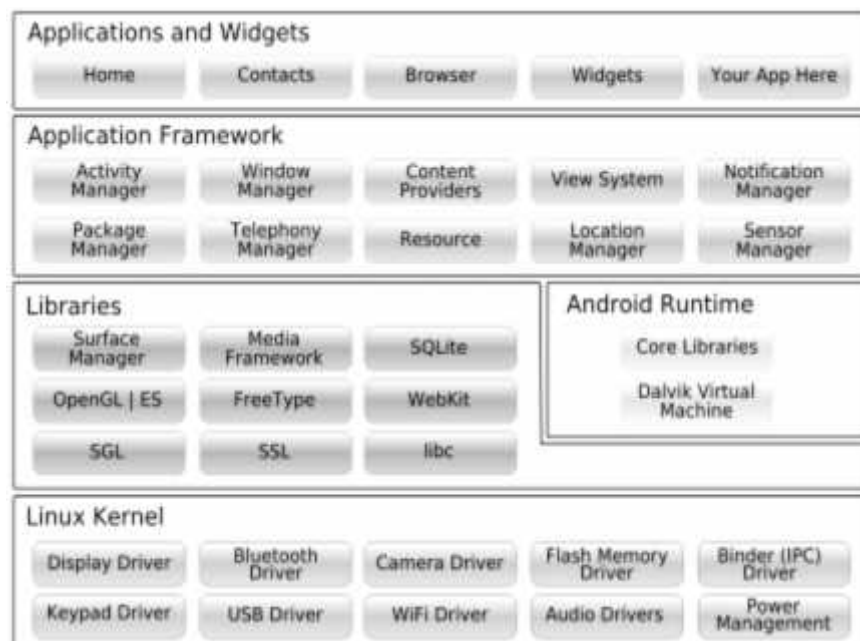


BAB II

TINJAUAN PUSTAKA

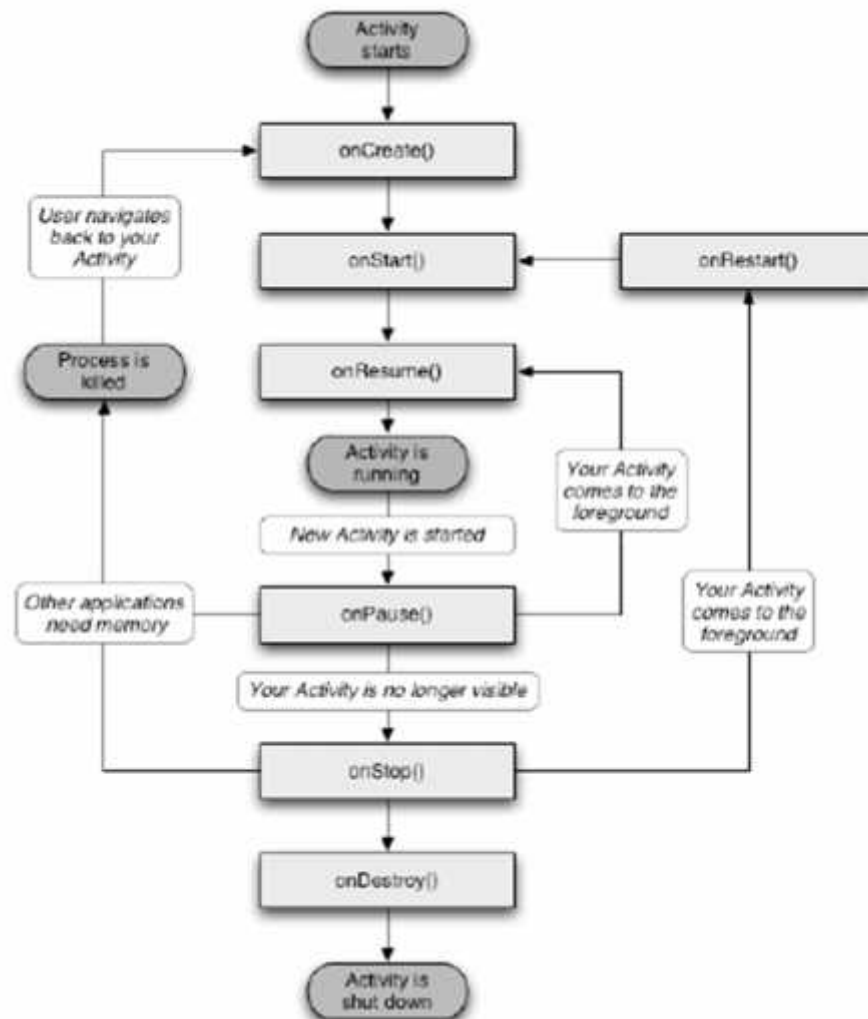
2.1 Android

Android adalah sistem operasi untuk telepon seluler yang berbasis *Linux*. *Android* menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak. Awalnya, *Google Inc.* membeli *Android Inc.*, pendatang baru yang membuat peranti lunak untuk ponsel. Kemudian untuk mengembangkan *Android*, dibentuklah *Open Handset Alliance*, konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk *Google*, *HTC*, *Intel*, *Motorola*, *Qualcomm*, *T-Mobile*, dan *Nvidia*. *Android* memiliki arsitektur sistem sebagai berikut



Gambar 2.1 arsitektur sistem *android*

Sumber : Meier (2009,p 3)



Gambar 2.2 Daur hidup *android*

Sumber : Felker (2011, p. 119)



Berikut penjelasan daur hidup *android*:

a. *onCreate()*

Method ini dipanggil saat *activity* pertama kali dibuat. Di sinilah inisialisasi dilakukan – membuat *UI*, mendapatkan data yang diperlukan, dan lainnya.

b. *onStart()*

Method ini dipanggil sesaat sebelum *activity* ditampilkan ke pengguna.

c. *onResume()*

Method ini dipanggil ketika *activity* berinteraksi dengan pengguna

d. *onPause()*

Method ini berjalan ketika *activity* berada di balik layar (*background*), tidak terlihat oleh pengguna tapi masih berjalan. Biasanya hal ini terjadi saat ada *activity* lainnya yang dijalankan. Di *state* inilah seharusnya data *program* kita disimpan ke *persistent state*.

e. *onStop()*

Method ini berjalan ketika *activity* sudah tidak terlihat lagi oleh pengguna dalam waktu yang cukup lama dan *activity* tidak diperlukan untuk sementara waktu

f. *onRestart()*

jika *method* ini dipanggil, berarti *activity* sedang ditampilkan ulang ke pengguna dari *state* berhenti (*stop*)



Pengembangan sistem operasi dan aplikasinya sendiri mengacu pada empat prinsip (Hermawan S, 2011) yaitu :

a. Terbuka

Android dibangun untuk menjadi benar-benar terbuka. Sebagai contoh, sebuah aplikasi dapat mengambil dan mengakses fungsi-fungsi utama ponsel seperti membuat panggilan, mengirim pesan teks, menggunakan kamera. Hal ini memungkinkan pengembang untuk membuat aplikasi yang lebih baik.

b. Semua Aplikasi Dibuat Sama

Android tidak membedakan antara aplikasi inti ponsel dan aplikasi pihak ketiga. Kedua jenis aplikasi ini dapat dibangun untuk memiliki akses yang sama ke ponsel. Pengguna dapat sepenuhnya mengatur telepon sesuai kepentingan mereka.

c. Mendobrak Batasan-batasan Aplikasi

Android membuang berbagai hambatan untuk membangun aplikasi baru yang *inovatif*. Misalnya, seorang pengembang dapat menggabungkan informasi dari *WEB* dengan data individu dari ponsel. Misalnya data kontak, kalender, atau lokasi geografis. Sehingga memberikan informasi yang lebih *relevan*. Dengan *android*, pengembang juga dapat membangun aplikasi yang memungkinkan pengguna untuk melihat lokasi dan terkoneksi dengan teman-temannya.

d. Pengembangan Aplikasi yang Cepat dan Mudah

Android menyediakan akses ke berbagai *libraries* dan *tools* yang dapat digunakan untuk membangun aplikasi yang kaya.



Dalam pengembangan aplikasi *Android* perlu dipahami beberapa komponen dasarnya (Komatineni et al, 2011) yaitu :

a. *Views*

Views adalah element *user interface (UI)* yang membentuk dasar dari sebuah *user interface*. *View* dapat berupa sebuah tombol, *label*, kolom teks, atau banyak elemen *UI* lainnya.

b. *Activity*

Activity adalah sebuah konsep dari *UI*. Sebuah *activity* biasanya memrepresentasikan sebuah layar dari sebuah aplikasi. Biasanya *activity* memiliki satu atau lebih *view*, tetapi bisa juga tidak memiliki *view*.

c. *Intent*

Sebuah *Intent* biasanya didefinisikan sebagai “*intention*” atau “*niat*” untuk melakukan beberapa pekerjaan. *Intent* dapat digunakan untuk :

1. Mengirim pesan.
2. Memulai *service*.
3. Menjalankan *activity*.
4. Menampilkan halaman *web*, atau daftar kontak ponsel.
5. Memutar nomor telepon atau menerima panggilan telepon.

d. *Content Provider*

Dengan menggunakan *content provider*, kita dapat mengekspos data dan menggunakan data dari aplikasi lain.

e. *Service*

Service pada *android* menyerupai *service* pada *Windows* ataupun *platform* lain. Aplikasi yang berpotensi berjalan lama ditempatkan Pada *background*. Contoh : aplikasi *e-mail*.

f. *Android Manifest. XML*

Android Manifest. XML mendefinisikan *konten* dan *behavior* dari aplikasi yang dibuat. Contoh : daftar *activity* yang dibuat, *permission* dan fitur ponsel yang digunakan oleh aplikasi.

2.2 Bluetooth

Bluetooth adalah salah satu bentuk komunikasi data secara *nirkabel* berbasis *frekuensi* radio. Penggunaan utama dari modul *Bluetooth* ini adalah menggantikan komunikasi *serial* menggunakan kabel. *Bluetooth* terdiri dari dua jenis perangkat, yaitu *Master* (pengirim data) dan *Slave* (penerima). Modul *HC-06* dari produsen koneksi secara *default* di *set* di kecepatan 9,600 bps (bisa dikustomisasi antara 1200 bps hingga 1,35 Mbps). Modul *HC-06* hanya bisa berperan sebagai *slave device*, module selain modul *bluetooth HC-06* ada modul *Bluetooth HC-05*, modul ini dapat berperan juga sebagai *bluetooth master device* ataupun *slave*, secara *default slave*.



Gambar 2.3 modul bluetooth HC-06

Jika kita akan menghubungkan dua sistem mikrokontroler agar bisa berkomunikasi *via serial port* dengan menggunakan media *frekuensi* radio, sebagai pilihan bisa menggunakan modul *bluetooth* ini. Agar dapat berkomunikasi dua sistem mikrokontroller ini maka dipasang sebuah modul *Bluetooth Master* pada satu sistem dan modul *Bluetooth Slave* pada sistem lainnya. Komunikasi dapat langsung dilakukan setelah kedua modul *Bluetooth* melakukan *pairing* koneksi. Koneksi melalui *Bluetooth* ini

menyerupai komunikasi *serial* komunikasi biasa, yaitu adanya pin komunikasi *TXD* dan pin komunikasi *RXD*.

Sistem mikrokontroler dipasang modul *Bluetooth Slave* maka sistem dapat berkomunikasi dengan perangkat lain misal laptop yang dilengkapi *adapter Bluetooth*, perangkat ponsel, *smartphone*, *gps* dan lain-lain. Jadi syarat utama agar dapat terkoneksi antara dua perangkat yang memiliki modul *Bluetooth* adalah yang satu *mode slave* dan yang satu *mode master*. Dan syarat lain yang wajib adalah *password* pada waktu *pairing* yang di minta cocok.

2.3 Mikrokontroler

Mikrokontroler merupakan sebuah sistem komputer yang seluruh atau sebagian besar elemennya dikemas dalam satu *chip (integrated circuit)*, sehingga sering disebut dengan *single chip microcomputer*. Mikrokontroler merupakan sebuah sistem komputer yang mempunyai satu atau beberapa tugas yang sangat *spesifik*, berbeda dengan *PC (Personal Komputer)* yang memiliki beragam fungsi. Perbedaan yang lainnya adalah perbandingan *RAM (Random Access Memory)* dan *ROM (Read Only Memory)* yang sangat berbeda antara mikrokontroler dengan komputer. Dalam mikrokontroler *ROM* jauh lebih besar dibandingkan dengan *RAM* sedangkan dalam komputer atau *PC*, *RAM* jauh lebih besar dibandingkan dengan *ROM*. Sistem mikrokontroler ini berupa modul perangkat keras (*Hardware*) yang dilengkapi dengan perangkat lunak (*software*) serta beberapa komponen pendukung lainnya hingga mikrokontroler dapat berfungsi.

Mikrokontroler merupakan komputer di dalam *chip* yang digunakan untuk mengontrol peralatan elektronik, yang menekankan *efisiensi* dan *efektifitas* biaya. Secara harfiahnya bisa disebut “pengendali kecil” dimana sebuah sistem elektronik yang sebelumnya banyak memerlukan komponen-komponen pendukung seperti *IC TTL* dan *CMOS* dapat *direduksi* / diperkecil



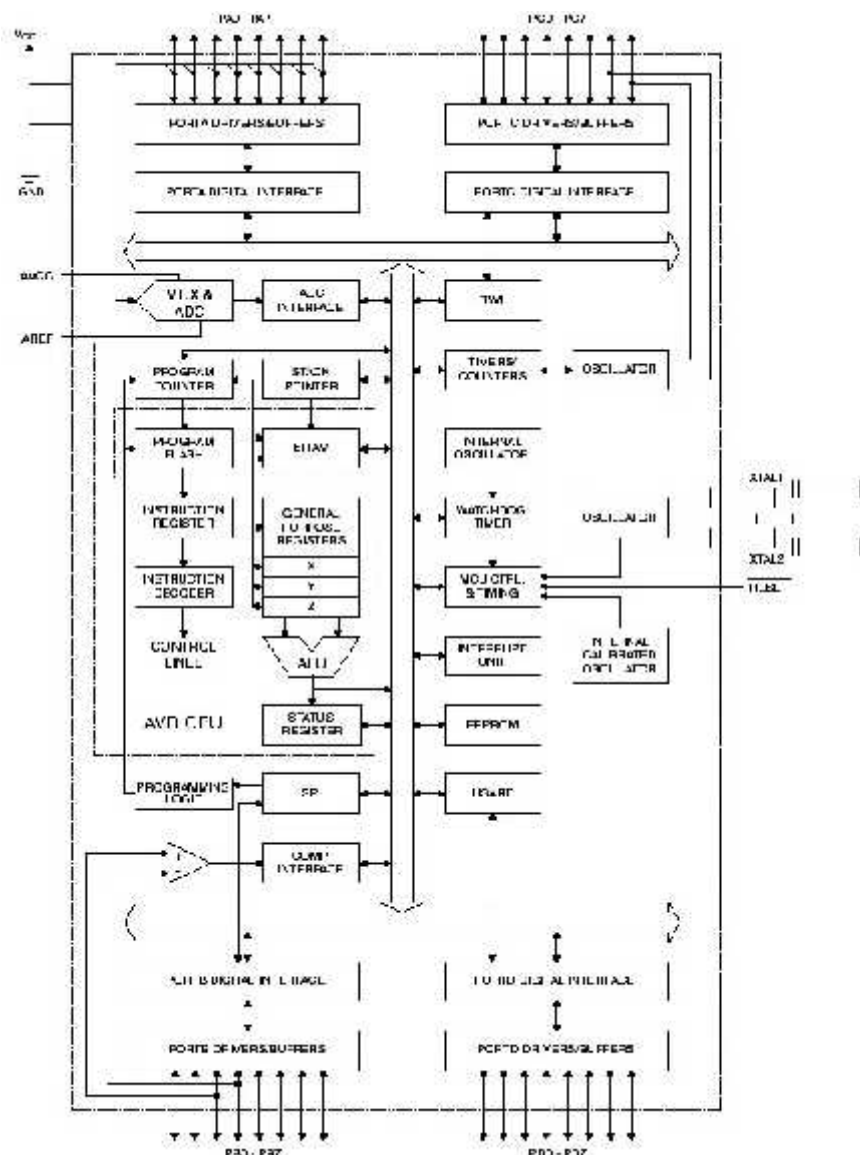
dan akhirnya terpusat serta dikendalikan oleh mikrokontroler ini. Dengan menggunakan mikrokontroler ini maka:

1. Sistem elektronik akan menjadi lebih ringkas.
2. Rancang bangun sistem elektronik akan lebih cepat karena sebagian besar dari sistem adalah perangkat lunak yang mudah dimodifikasi.
3. Pencarian gangguan lebih mudah ditelusuri karena sistemnya yang kompak. Namun demikian tidak sepenuhnya mikrokontroler bisa *mereduksi* komponen *IC TTL* dan *CMOS* yang seringkali masih diperlukan untuk aplikasi kecepatan tinggi atau sekedar menambah jumlah saluran *input* dan *output (I/O)*. dengan kata lain, mikrokontroler adalah *versi mini* atau *mikro* dari sebuah komputer karena mikrokontroler sudah mengandung beberapa bagian yang langsung bisa dimanfaatkan, misalnya *port paralel*, *port serial*, *komparator*, *konversi digital ke analog (DAC)*, *konversi analog ke digital (ADC)*, dan sebagainya hanya menggunakan *Minimum System* yang tidak rumit atau *kompleks*.

Mikrokontroler adalah otak dari suatu sistem elektronika seperti halnya mikroprosesor sebagai otak komputer. Namun mikrokontroler memiliki nilai tambah karena didalamnya sudah terdapat memori dan sistem *input/output* dalam suatu kemasan *IC*. Mikrokontroler *AVR (Alf and Vegard's RISC processor)* standar memiliki *arsitektur 8-bit*, dimana semua instruksi dikemas dalam *kode 16-bit* dan sebagian besar instruksi dieksekusi dalam satu *siklus clock*. Berbeda dengan instruksi *MCS-51* yang membutuhkan 12 *siklus clock* karena memiliki *arsitektur CISC* (seperti komputer).

Secara umum, *AVR* dapat dikelompokkan menjadi 4 kelas, yaitu keluarga *ATTiny*, keluarga *AT90Sxx*, keluarga *ATMega* dan *AT89RFxx*. Pada dasarnya yang membedakan masing-masing kelas adalah memori, *peripheral*, dan fungsinya. Dari segi *arsitektur* dan *instruksi* yang digunakan, mereka bisa dikatakan hampir sama. Oleh karena itu, dipergunakan salah satu *AVR* produk *Atmel*, yaitu *ATMega 8535*. Selain mudah didapatkan dan lebih murah *ATMega 8535* juga memiliki fasilitas yang lengkap. Untuk tipe *AVR* ada 3 jenis yaitu *ATTiny*, *AVR klasik*, dan *ATMega*. Perbedaannya hanya pada

fasilitas dan I/O yang tersedia serta fasilitas lain seperti ADC, EEPROM, dan lain sebagainya. Salah satu contohnya adalah ATmega 8535. Memiliki teknologi RISC dengan kecepatan maksimal 16 MHz membuat ATmega 8535 lebih cepat bila dibandingkan dengan varian MCS51. Dengan fasilitas yang lengkap tersebut menjadikan ATmega 8535 sebagai mikrokontroler yang powerful. Adapun blok diagramnya sebagai berikut.



Gambar 2.4 Blok Diagram ATMega 8535

(Sumber :<http://duniaelektronika.co.id/> diakses tanggal 28 oktober 2014 pukul 14.50)



Dari gambar tersebut dapat dilihat bahwa ATMega 8535 memiliki bagian sebagai berikut :

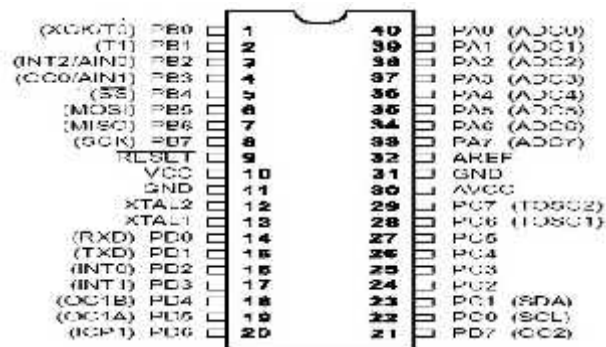
- Saluran *I/O* sebanyak 32 buah, yaitu *Port A*, *Port B*, *Port C*, *Port D*.
- *ADC 10 bit* sebanyak 8 saluran.
- Tiga buah *Timer/Counter* dengan kemampuan perbandingan.
- *CPU* yang terdiri atas 32 buah *register*.
- *Watchdog Timer* dengan *osilator internal*.
- *SRAM* sebesar 512 *byte*.
- *Memori Flash* sebesar 8 kb dengan kemampuan *Read While Write*.
- *Unit interupsi internal* dan *eksternal*.
- *Port* antar muka *SPI*.
- *EEPROM* sebesar 512 *byte* yang dapat diprogram saat operasi.
- Antar muka *komparator analog*.
- *Port USART* untuk komunikasi *serial*.

Kapabilitas detail dari ATMega8535 adalah sebagai berikut :

- Sistem mikroprosesor 8 bit berbasis *RISC* dengan kecepatan maksimal 16 MHz.
 - *Kapabilitas memori flash* 8 Kb, *SRAM* sebesar 512 *byte*, dan *EEPROM* (*Electrically Erasable Programmable Read Only Memory*) sebesar 512 *byte*.
 - *ADC internal* dengan *fidelitas 10 bit* sebanyak 8 *channel*.
 - *Portal komunikasi serial (USART)* dengan kecepatan maksimal 2,5 Mbps.
 - Enam pilihan *mode sleep* menghemat penggunaan daya listrik.
-

2.3.1 Konfigurasi PIN ATmega8535

Mikrokontroler ATmega8535 mempunyai jumlah pin sebanyak 40 buah, dimana 32 pin digunakan untuk keperluan *port I/O* yang dapat menjadi pin *input/output* sesuai konfigurasi. Pada 32 pin tersebut terbagi atas 4 bagian (*port*), yang masing-masingnya terdiri atas 8 *pin*. *Pin-pin* lainnya digunakan untuk keperluan rangkaian *osilator*, *supply* tegangan, *reset*, serta tegangan *referensi* untuk *ADC*. Untuk lebih jelasnya, konfigurasi *pin* ATmega8535 dapat dilihat pada gambar 2.9.



Gambar 2.5 Konfigurasi Pin ATmega8535

(Sumber : <http://duniaelektronika.co.id/>)

Berikut ini adalah susunan pin-pin dari ATmega8535;

- VCC merupakan pin yang berfungsi sebagai pin masukan catu daya
- GND merupakan *pin ground*
- Port A (PA0.PA7) merupakan *pin I/O* dua arah dan *pin* masukan *ADC*
- Port B (PB0.PB7) merupakan *pin I/O* dua arah dan *pin* fungsi khusus, yaitu *Timer/Counter*, *Komparator Analog*, dan *SPI*
- Port C (PC0..PC7) merupakan *pin I/O* dua arah dan *pin* fungsi khusus, yaitu *TWI*, *Komparator Analog*, dan *Timer Oscilator*

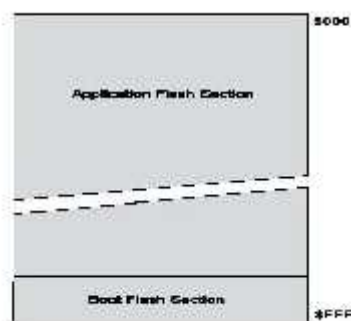
- *Port D (PD0..PD7)* merupakan *pin I/O* dua arah dan *pin* fungsi khusus, yaitu *Komparator Analog*, *Interupsi Eksternal* dan komunikasi serial *USART*
- *Reset* merupakan *pin* yang digunakan untuk mereset mikrokontroler
- *XTAL1* dan *XTAL2* merupakan *pin* masukkan *clock eksternal* (*osilator* menggunakan *kristal*, biasanya dengan *frekuensi 11,0592 MHz*)

2.3.2 Peta Memori ATmega8535

ATmega8535 memiliki dua jenis memori yaitu *Data Memory* dan *Program Memory* ditambah satu fitur tambahan yaitu *EEPROM Memory* untuk menyimpan data.

a. Program Memory

ATMEGA8535 memiliki *On-Chip In-System Reprogrammable Flash Memory* untuk menyimpan program. Untuk alasan keamanan, *program memory* dibagi menjadi dua bagian, yaitu *Boot Flash Section* dan *Application Flash Section*. *Boot Flash Section* digunakan untuk menyimpan program *Boot Loader*, yaitu program yang harus dijalankan pada saat AVR reset atau pertama kali diaktifkan.



Gambar 2.6 Peta Memori Program

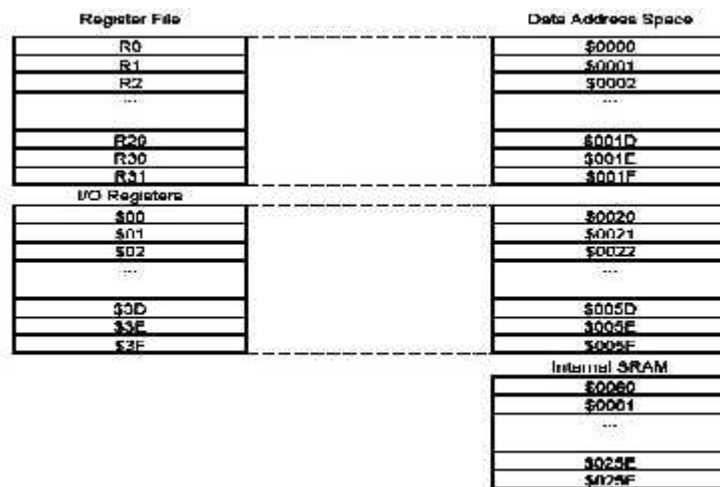
(Sumber : <http://duniaelektronika.co.id/>)

Application Flash Section digunakan untuk menyimpan program aplikasi yang dibuat user. AVR tidak dapat menjalankan program

aplikasi ini sebelum menjalankan program *Boot Loader*. Besarnya memori *Boot Flash Section* dapat diprogram dari 128 word sampai 1024 word tergantung *setting* pada konfigurasi *bit* di register *BOOTSZ*. Jika *Boot Loader* diproteksi, maka program pada *Application Flash Section* juga sudah aman.

b. *Data Memory*

Gambar berikut menunjukkan peta memori *SRAM* pada ATMEGA8535. Terdapat 608 lokasi *address* data memori. 96 lokasi *address* digunakan untuk *Register File* dan *I/O Memory* sementara 512 lokasi *address* lainnya digunakan untuk *internal data SRAM*. *Register file* terdiri dari 32 *general purpose working register*, *I/O register* terdiri dari 64 register.

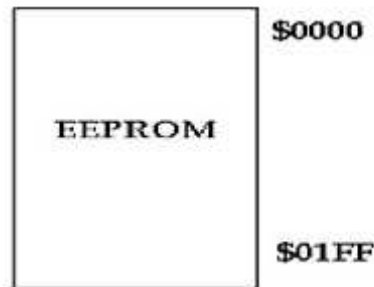


Gambar 2.7 Peta Memori Data

(Sumber : <http://duniaelektronika.co.id>)

2.3.3 EEPROM Data Memory

ATMEGA8535 memiliki *EEPROM* 8 bit sebesar 512 byte untuk menyimpan data. Loaksinya terpisah dengan *system address register*, *data register* dan *control register* yang dibuat khusus untuk *EEPROM*. Alamat *EEPROM* dimulai dari \$0000 sampai \$1FF.



Gambar 2.8 EEPROM Data Memori

(Sumber :<http://duniaelektronika.co.id>)

2.4 Motor Servo

Motor servo adalah sebuah alat yang dapat mengubah energi listrik menjadi energi mekanik. Motor servo sebenarnya digerakkan oleh sebuah motor DC. Karena motor servo digerakkan oleh motor DC maka jenis arus yang dipergunakan adalah jenis arus searah. Selain terdapat komponen motor DC, di dalam motor servo pun terdapat roda gigi yang berfungsi untuk mereduksi putaran dan untuk memperbesar torsi yang dihasilkan oleh motor DC tersebut.

Berbeda dengan motor DC yang akan langsung berputar ketika diberi tegangan, motor servo tidak akan berputar tanpa adanya perintah tertentu walaupun telah diberi tegangan. Motor servo dapat bergerak bila dialiri arus DC dan kaki sinyal diberi isyarat sinyal atau pulsa berupa arus listrik. Dengan diberi pulsa tertentu, motor servo akan berputar pada sudut tertentu.

Jenis motor servo ada 2 yaitu jenis motor servo *continuous* dan motor servo *standard*. Kedua motor servo ini tidak jauh berbeda hanya saja pada putarannya. Berikut ini adalah penjelasan kedua jenis motor servo tersebut.

1. Motor Servo *Standard* 180°

Motor servo jenis ini hanya mampu bergerak dua arah (*CW* dan *CCW*) dengan masing-masing sudut mencapai 90° sehingga total sudut dari kanan – tengah – kiri adalah 180°. Motor servo *standard* lebih mudah *dikontrol* jika dibandingkan dengan motor servo *continuous* karena motor servo *standard* dapat diatur sudutnya sesuai dengan yang diinginkan (tidak berputar secara kontinyu).

Gambar dibawah ini adalah jenis Motor servo standar futaba s3003 yang digunakan pada perancangan Laporan Akhir ini . untuk Spesifikasi yang lebih jelas dapat dilihat pada lampiran Laporan Akhir ini .



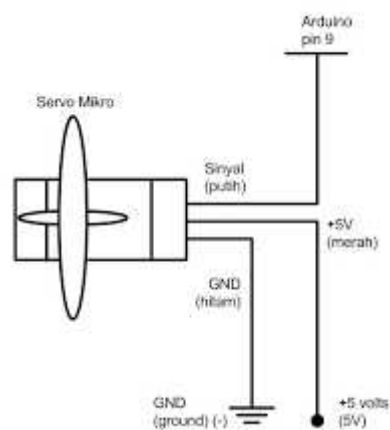
Gambar 2.9 Motor Servo

(sumber : www.google.com+servos.html)

2. Motor Servo *Continuous*

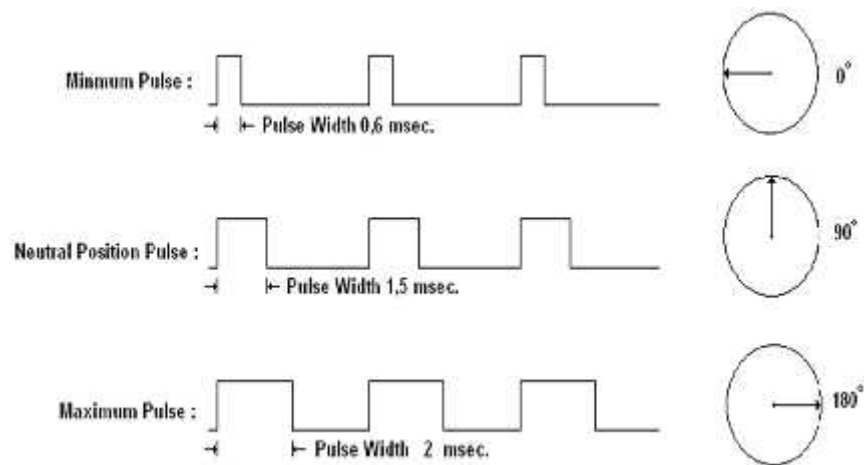
Motor servo jenis ini mampu bergerak dua arah (CW dan CCW) tanpa batasan sudut putar (dapat berputar secara kontiu).

Motor servo hanya memiliki 3 kabel yang mana masing-masing fungsinya terdiri dari positif (*Vcc*), negatif (*Ground*) dan *control* (*Signal*). Motor servo mampu bergerak searah jarum jam ataupun berlawanan arah jarum jam tanpa membalik *pin connector* pada motor servo, hal ini disebabkan bahwa pada motor servo telah terdapat *driver* untuk membalik *polaritas* motor DC yang ada pada motor servo. *Konfigurasi pin* pada motor servo dapat dilihat pada gambar 2.10



Gambar 2.10 Rangkaian Motor Servo

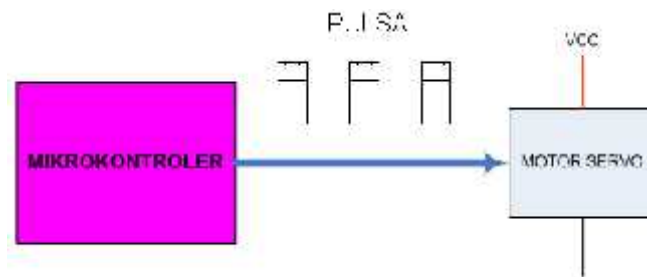
Salah satu jenis motor servo adalah motor servo jenis *standar*. Motor servo jenis *standar* hanya dapat berputar 180° searah atau berlawanan arah jarum jam. Prinsip kerja motor servo *standar* adalah kaki *sinyal* motor servo diberi sinyal *digital* dengan lebar *sinyal* antara 0,60 milidetik sampai 2,00 milidetik. Sinyal akan dideteksi setiap 20 milidetik. Apabila dalam selang waktu lebih dari 20 milidetik motor servo tidak mendeteksi *sinyal* maka motor servo akan *slip*. Bentuk *sinyal pengontrolan* motor servo dapat dilihat pada gambar 2.9.



Gambar 2.11 Bentuk *sinyal* pengontrolan motor servo

Untuk memposisikan poros motor servo pada sudut tertentu dan selama waktu tertentu, lebar dan jumlah pulsa tertentu harus diberikan pada kaki sinyal motor servo. Agar motor servo dapat diposisikan pada sudut tertentu terlebih dahulu lebar pulsa untuk posisi 0° dan posisi 180° harus diketahui. Setelah mengetahui lebar pulsa pada kedua posisi tersebut, lebar pulsa untuk posisi sudut yang lain bisa diketahui dengan cara *interpolasi*.

Karena lamanya pulsa yang diberikan sangat singkat maka pemberian pulsa pada kaki *sinyal* tidak mungkin dilakukan secara *manual*. Selain itu sejumlah pulsa harus diberikan selama selang waktu tertentu. Oleh karena itu, pemberian pulsa pada kaki *sinyal* motor servo dilakukan dengan menggunakan mikrokontroler. Mikrokontroler akan terus memberikan sejumlah pulsa ke kaki *sinyal* pada motor servo sesuai dengan program. *Skematis* pengiriman pulsa dapat dilihat pada gambar 2.12

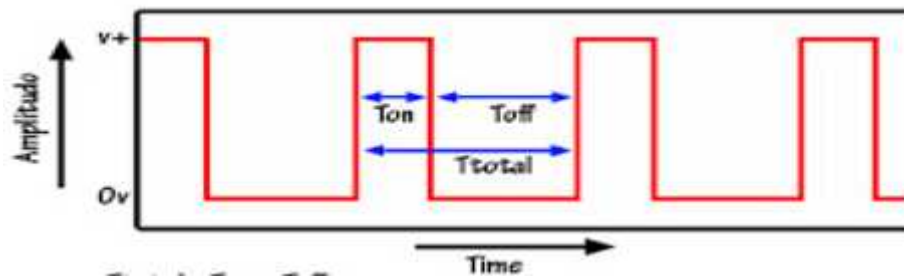


Gambar 2.12 Skema Pengiriman Pulsa Motor Servo

2.5 Pulsh Width Modulation

Pulse Width Modulation (PWM) secara umum adalah sebuah cara memanipulasi lebar *sinyal* yang dinyatakan dengan pulsa dalam suatu perioda, untuk mendapatkan tegangan rata-rata yang berbeda. Beberapa contoh aplikasi *PWM* adalah *pemodulasian* data untuk telekomunikasi, *pengontrolan* daya atau tegangan yang masuk ke beban, *regulator* tegangan, *audio effect* dan penguatan, serta aplikasi-aplikasi lainnya. Aplikasi *PWM* berbasis mikrokontroler biasanya berupa pengendalian kecepatan motor DC, pengendalian motor servo, pengaturan nyala terang *LED* dan lain sebagainya.

Sinyal *PWM* pada umumnya memiliki *amplitudo* dan *frekuensi* dasar yang tetap, namun memiliki lebar pulsa yang bervariasi. Lebar Pulsa *PWM* berbanding lurus dengan *amplitudo* sinyal asli yang belum termodulasi. Artinya, Sinyal *PWM* memiliki *frekuensi* gelombang yang tetap namun *duty cycle* bervariasi (antara 0% hingga 100%).



Gambar 2.13 Sinyal PWM

(sumber :http://andri_mz.staff.ipb.ac.id/files/2013/11/sinyal-PWM-2.jpg)

Keterangan :

Ton = Waktu pulsa "High"

Toff = Waktu pulsa "Low"

D = *Duty cycle* adalah lamanya pulsa *high* dalam satu perioda

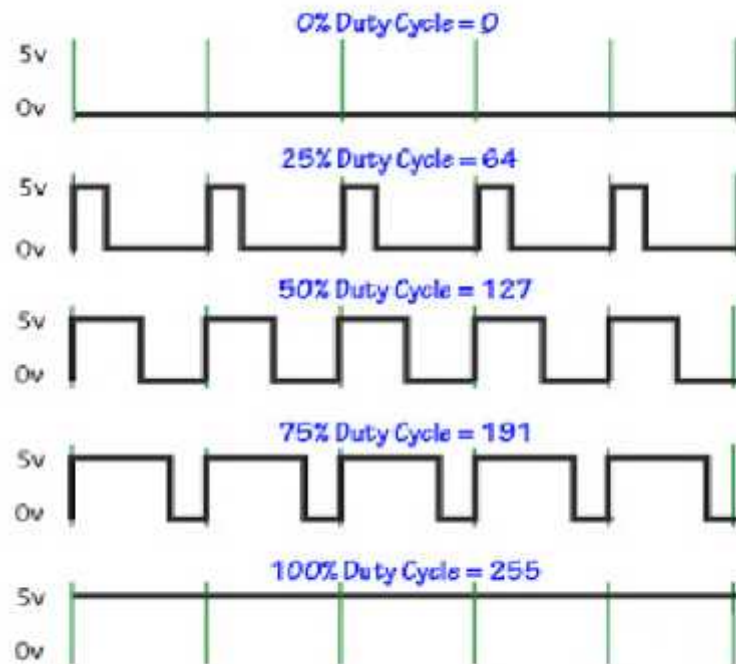
Rumus :

$$T_{total} = T_{on} + T_{off}$$

$$D = \frac{T_{on}}{T_{total}}$$

$$V_{out} = \frac{T_{on}}{T_{total}} \times V_{in}$$

Pulse Width Modulation (PWM) merupakan salah satu teknik untuk mendapatkan *signal analog* dari sebuah *piranti digital*. Sebenarnya *Sinyal PWM* dapat dibangkitkan dengan banyak cara, dapat menggunakan metode analog dengan menggunakan rangkaian *op-amp* atau dengan menggunakan metode *digital*. Dengan metode *analog* setiap perubahan *PWM*-nya sangat halus, sedangkan menggunakan metode *digital* setiap perubahan *PWM* dipengaruhi oleh *resolusi* dari *PWM* itu sendiri. *Resolusi* adalah jumlah variasi perubahan nilai dalam *PWM* tersebut. Misalkan suatu *PWM* memiliki *resolusi 8 bit* berarti *PWM* ini memiliki variasi perubahan nilai sebanyak $2^8 = 256$ variasi mulai dari 0 – 255 perubahan nilai yang mewakili *duty cycle* 0 – 100% dari keluaran *PWM* tersebut



Gambar 2.14 Pulsa PWM

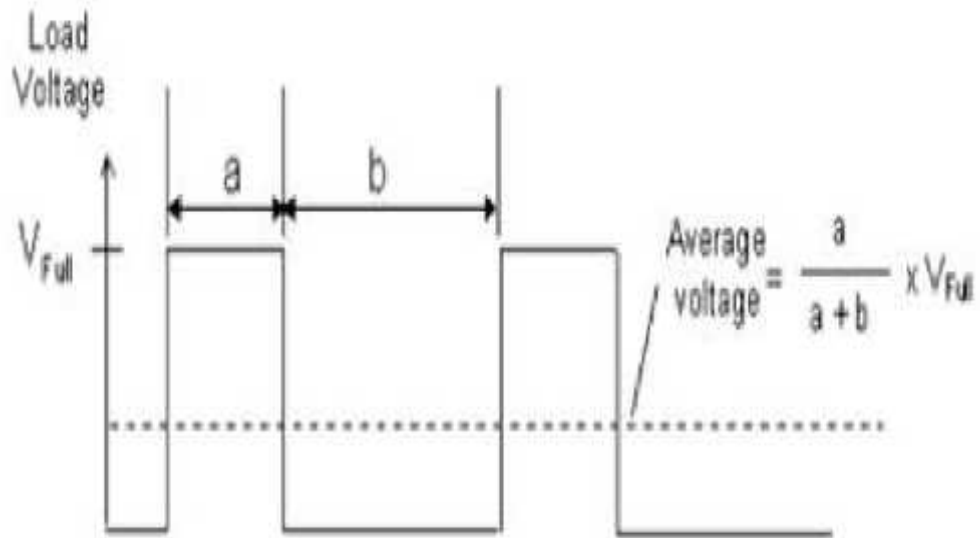
(Sumber: http://andri_mz.staff.ipb.ac.id/files/2013/11/pulsa-pwm-2.jpg)

Dengan cara mengatur lebar pulsa “on” dan “off” dalam satu perioda gelombang melalui pemberian besar *signal referensi output* dari suatu PWM akan didapat *duty cycle* yang diinginkan. *Duty cycle* dari PWM dapat dinyatakan sebagai:

$$\text{DutyCycle} = t_{\text{ON}} / (t_{\text{ON}} + t_{\text{OFF}}) \times 100\% \dots \dots \dots (1)$$

Duty cycle 100% berarti *signal* tegangan pengatur motor dilewatkan seluruhnya. Jika tegangan catu 100V, maka motor akan mendapat tegangan 100V. pada *duty cycle* 50%, tegangan pada motor hanya akan diberikan 50% dari total tegangan yang ada, begitu seterusnya.

Untuk melakukan perhitungan pengontrolan tegangan *output* motor dengan metode PWM cukup sederhana sebagaimana dapat dilihat pada ilustrasi Gambar 2.15 di bawah ini.



Gambar 2.15 Perhitungan PWM

(http://andri_mz.staff.ipb.ac.id/files/2013/11/perhitungan-pwm.jpg)

2.6 Webcam

Web camera atau yang biasa dikenal dengan *webcam*, adalah kamera yang gambarnya bisa di akses menggunakan *world wide web* (www), program *instant messaging*, atau aplikasi komunikasi dengan tampilan *video* pada PC. *Webcam* juga digambarkan sebagai kamera *video digital* yang sengaja di *desain* sebagai kamera dengan *resolusi* rendah. *Webcam* dapat digunakan untuk *25ndust* keamanan. Pada beberapa *webcam*, ada yang di lengkapi dengan *software* yang mampu mendeteksi pergerakan dan suara. Dengan *software* tersebut, memungkinkan PC yang terhubung ke kamera untuk mengamati pergerakan dan suara, serta merekamnya ketika terdeteksi. Hasil rekaman ini bisa disimpan pada komputer, e-mail atau di *upload* ke *internet* (Wibowo, 2010).

Webcam sangat bermanfaat dalam bidang telekomunikasi, bidang keamanan dan bidang *25ndustry*. Sebagai contoh *webcam* digunakan untuk *videocall chatting*, *surveillience camera*, dan sebagai *video conference* oleh beberapa *user*.



Gambar 2.16 Contoh Kamera Webcam

Sebuah *web camera* yang sederhana terdiri dari sebuah lensa standar, dipasang di sebuah papan sirkuit untuk menangkap *sinyal* gambar; *casing* (*cover*), termasuk *casing* depan dan *casing* samping untuk menutupi lensa standar dan memiliki sebuah lubang lensa di *casing* depan yang berguna untuk memasukkan gambar; kabel *support*, yang dibuat dari bahan yang *fleksibel*, salah satu ujungnya dihubungkan dengan papan *sirkuit* dan ujung satu lagi memiliki *connector*, kabel ini *dikontrol* untuk menyesuaikan ketinggian, arah dan sudut pandang *web camera*. Sebuah *web camera* biasanya dilengkapi dengan *software*, *software* ini mengambil gambar-gambar dari kamera *digital* secara terus menerus ataupun dalam *interval* waktu tertentu dan menyiarkannya melalui *koneksi internet*. Ada beberapa metode penyiaran, metode yang paling umum adalah *hardware* mengubah gambar ke dalam bentuk file *JPG* dan *uploadnya* ke *web server* menggunakan *File Transfer Protocol* (FTP).

Frame rate mengindikasikan jumlah gambar sebuah *software* dapat ambil dan *transfer* dalam satu detik. Untuk *streaming video*, dibutuhkan minimal 15 *frame per second* (fps) atau idealnya 30 fps. Untuk mendapatkan *frame rate* yang tinggi, dibutuhkan *koneksi internet* yang



tinggi kecepatannya. Sebuah *web camera* tidak harus selalu terhubung dengan komputer, ada *web camera* yang memiliki *software webcam* dan *web server built-in*, sehingga yang diperlukan hanyalah *koneksi internet*. *Web camera* seperti ini dinamakan “*network camera*”. Kita juga bisa menghindari penggunaan kabel dengan menggunakan hubungan radio, *koneksi Ethernet* ataupun *WiFi*.
