BAB II
LANDASAN TEORI

2.1 Airfield Lighting System (AFL)

Airfield Lighting System (AFL) atau disebut juga dengan *Aeronautical Ground Lighting* (AGL) merupakan istilah yang digunakan pada bandara untuk membantu dan melayani pilot secara visual menggunakan berbagai jenis lampu pada saat pesawat melakukan proses tinggal landas, mendarat, dan melakukan *taxi* agar dapat bergerak secara efisien dan aman.[1]

Secara umum lampu AFL yang terdapat pada wilayah utama bandara (*landing movement*) dikelompokkan menjadi tiga jenis lampu yaitu lampu *Runway*, lampu *Taxiway* dan Apron. Setiap jenis lampu AFL memiliki nama lampu rambu yang berbeda-beda sesuai dengan lokasi penempatan lampu. Berikut ini dijelaskan lampu rambu sebagai berikut:[1]

1. Lampu *Runway*

   Runway lighting yaitu lampu yang dipasang sepanjang landasan pacu (*runway*) baik di pinggir maupun di tengah, lampu *Runway* juga dipasang pada akhir landasan pacu. Beberapa lampu rambu pada *Runway* yaitu sebagai berikut:

   a. Touchdown Zone Light

   Touchdown Zone Light (*TZL*) yaitu lampu yang dipasang setelah melewati lampu *Threshold* jika dilihat dari penerbang ke arah pendaratan.

   b. Runway Centerline Light

   Runway Centerline Light (RCL) yaitu lampu yang ditempatkan sepanjang landasan pacu tepat ditengah-tengah landasan.

   c. Runway Edge Light

   Lampu *Runway edge light* yaitu lampu landasan pacu yang dipasang di tepi kiri dan kanan landasan.

   d. *Threshold Light*

   Lampu *Trheshold light* yaitu yang dipasang pada batas ambang landasan pacu dengan jarak tertentu.

e. *Runway End Light*

Lampu *Runway End Light* yaitu lampu yang dipasang pada batas ambang landasan, berfungsi sebagai penunjuk batas akhir landasan.

f. *Apporach Light*.

Approach light yaitu lampu yang dipasang pada perpanjangan landasan pacu, berfungsi sebagai petunjuk kepada penerbang tentang posisi arah pendaratan dan jarak terhadap ambang landasan pada saat pendaratan.

g. *Sequence Flashing Light* (SQFL)

Yaitu lampu penerangan berkedip berurutan (sekuensial) pada arah pendekatan pendaratan. Lampu SQFL biasanya dipasang pada Bar 1 sampai dengan Bar 21.

h. PAPI (*Precision Approach Path Indicator*)

Lampu PAPI yaitu lampu yang menginformasikan kepada penerbang mengenai sudut luncur yang benar dan memandu penerbang melakukan pendekatan menuju titik pendaratan pada daerah *Touchdown*.

2. Lampu *Taxiway*

Taxiway light yaitu lampu yang dipasang pada tepi kiri dan kanan *Taxiway*, berfungsi untuk memandu penerbang dari landasan pacu ke/dari tempat parkir pesawat. Beberapa lampu *Taxiway* yaitu sebagai berikut:

a. *Taxiway Edge Lighting*

Yaitu lampu *Taxiway* yang dipasang pada tepi kiri dan kananan sepanjang jalur *Taxi*.

b. *Taxiway Centerline Lighting*

Yaitu lampu yang dipasang sepanjang jalur *Taxi*, lampu ini berada tepat ditengah-tengah jalur *Taxiway*.

c. *Taxiway End Lighting*

Yaitu lampu yang dipasang pada ujung jalur *Taxiway*, lampu ini berfungsi sebagai petunjuk akhir dari jalur *Taxiway*.

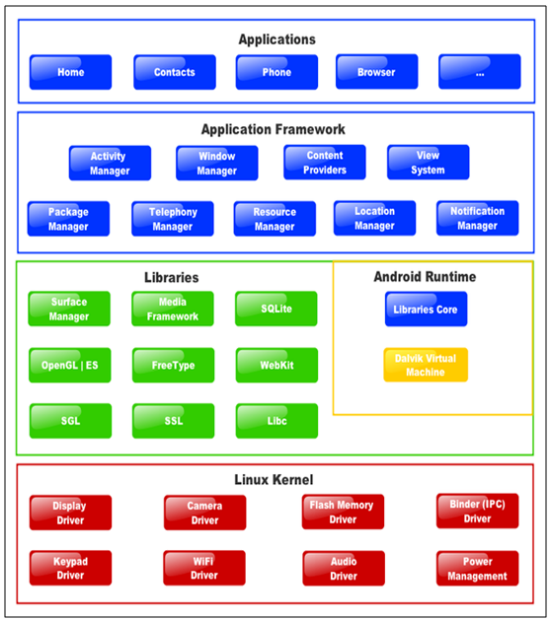
3. Lampu *Apron*

   Yaitu rambu penerangan yang dipasang di tepi *Apron* untuk memberi tanda batas pinggir *Apron*.

2.2 Sistem Operasi Android

Android adalah sistem operasi yang berbasis Linux atau *open source* yang digunakan untuk perangkat *mobile* seperti *smartphone* dan tablet PC. Pengembangan program aplikasi lebih mudah karena disediakannya Android SDK (*Software Development Kit*) oleh Google. Selain Android SDK untuk memudahkan dalam pengembangan aplikasi, Android juga bebas dalam bentuk *operating system*.[2]

Kernel Linux versi 2.6 menjadi dasar bagi sistem operasi Android yang mencakup *middleware*, perpustakaan, dan fungsi API yang ditulis dalam bahasa C dan perangkat lunak aplikasi berjalan pada kerangka aplikasi Java yang kompatibel perpustakaan berbasis Apache. Sistem operasi Android berjalan menggunakan *Dalvik Virtual Machine* (DVM) untuk menjalankan kode yang ditulis dalam bahasa pemrograman Java. Pada Gambar 2.1 merupakan arsitektur sistem operasi Android.[2]. (Sumber: Wikipedia, 2011)



Gambar 2.1. Arsitektur Sistem Operasi Android

2.2.1 Sejarah Android

Pada Juli 2000, Google bekerjasama dengan Android Inc., perusahaan yang berada di Palo Alto, California Amerika Serikat. Para pendiri Android Inc. bekerja pada Google, di antaranya Andy Rubin, Rich Miner, Nick Sears, dan Chris White. Saat itu banyak yang menganggap fungsi Android hanyalah sebagai perangkat lunak pada telepon seluler. Sejak saat itu muncul rumor bahwa Google hendak memasuki pasar telepon seluler, karena pada saat itu perusahaan Google memerintahkan sebuah tim yang dipimpin Andy Rubin bertugas mengembangkan program perangkat seluler yang didukung oleh kernel Linux. Hal ini menunjukkan indikasi bahwa Google sedang bersiap menghadapi persaingan dalam pasar telepon seluler.[2]

Sekitar September 2007 sebuah studi melaporkan bahwa Google mengajukan hak paten aplikasi telepon seluler (akhirnya Google mengenalkan Nexus One, salah satu jenis telepon pintar GSM yang menggunakan Android pada sistem operasinya. Telepon seluler ini diproduksi oleh HTC Corporation dan tersedia di pasaran pada 5 Januari 2010).[2]

Pada 9 Desember 2008, diumumkan anggota baru yang bergabung dalam program kerja Android ARM Holdings, Atheros Communications, diproduksi oleh Asustek Computer Inc, Garmin Ltd, Softbank, Sony Ericsson, Toshiba Corp, dan Vodafone Group Plc. Seiring pembentukan *Open Handset Alliance* (OHA), OHA mengumumkan produk perdana mereka, Android, perangkat bergerak (*mobile*) yang merupakan modifikasi kernel Linux 2.6. Sejak Android dirilis telah dilakukan berbagai pembaruan berupa perbaikan *bug* dan penambahan fitur baru.[2]

2.2.2 Keunggulan Android

Sistem operasi Android dikenal dengan "*platform mobile* pertama yang lengkap, Terbuka, dan bebas", Android juga memiliki keunggulan dibandingkan dengan sistem operasi *mobile* lainya, berikut keunggulan yang terdapat pada sistem operasi Android.[2]

- *Framework* aplikasi yang mendukung penggantian dan *reusable*.
- Mesin Virtual Dalvik dioptimalkan untuk perangkat *mobile*.
- *Integrated browser* berdasarkan *engine open source* WebKit.
- Grafis yang dioptimalkan dan didukung oleh *libraries* grafis 2D, grafis 3D berdasarkan spesifikasi openGL ES 1.0 (Opsional akselerasi *hardware*).
- SQLite untuk penyimpanan data (*database*).
- Media *support* yang mendukung audio, video, dan gambar.
- *Bloetooth*, EDGE, 3D, dan WiFi (tergantung *hardware*).
- Kamera, GPS, kompas, dan *accelerometer* (tergangtung *hardware*).

2.2.3 Versi Android

Sistem operasi Android mengalami perkembangan sangat cepat, dimungkinkan pada penghujung tahun 2012 diperkirakan hampir semua vendor selular di dunia menggunakan Android sebagai sistim operasinya. Adapun versi-versi android yang pernah dirilis adalah sebagai berikut:[3]

- Android Versi 1.1

  Pada 9 Maret 2009, Google merilis Android versi 1.1. Andorid versi ini dilengkapi dengan pembaruan estetis pada aplikasi, jam, alarm, *voice search* (pencarian suara), pengiriman pesan dengan Gmail, dan pemberitahuan email.
- Android Veris 1.5 (*Cupcake*)

  Pada pertengahan Mei 2009, Google kembali merilis telepon seluler dengan menggunakan Android dan SDK (*Softwre Development Kit*) dengan versi 1.5 (*Cupcake*). Terdapat beberapa pembaharuan termasuk juga penambahan beberapa fitur dalam seluler, yakni kemampuan merekam dan menonton video dengan modus kamera, meng-*upload* video ke Youtube dan gambar ke Picasa langsung dari telepon, dukungan *Bluetooth* A2DP, kemampuan terhubung secara otomatis ke *headset Bluetooth*, animasi layar, dan *keyboard* pada layar yang dapat disesuaikan dengan sistem.
- Android Versi 1.6 (*Donut*)

  Donut (versi 1.6) dirilis pada September dengan menampilkan proses pencarian yang lebih baik dibanding sebelumnya, penggunaan baterai

indikator dan kontrol *applet* VPN. Penambahan fitur lainnya adalah galeri yang memungkinkan pengguna untuk memilih foto yang akan dihapus, kamera, *camcorder* dan galeri yang diintegrasikan, CDMA/EVDO, 802.1x, VPN, *Gestures*, dan *Text to speech engine*, kemampuan *dial* kontak, teknologi *text to change speech* (tidak tersedia pada semua ponsel, pengadaan resolusi VWGA).

- Android Versi 2.0/2.1 (*Éclair*)

  Pada 3 Desember 2009 kembali diluncurkan ponsel Android dengan versi 2.0/2.1 (*Eclair*), perubahan yang dilakukan adalah pengoptimalan *hardware*, peningkatan Google Maps 3.1.2, perubahan UI dengan *browser* baru dan dukungan HTML5, daftar kontak yang baru, dukungan *flash* untuk kamera 3,2 MP, *digital Zoom*, dan *Bluetooth* 2.1.

  Untuk bergerak cepat dalam persaingan perangkat generasi berikut, Google melakukan investasi dengan mengadakan kompetisi aplikasi *mobile* terbaik (*killer apps* - aplikasi unggulan).

- Android Versi 2.2 (Froyo: *Frozen Yoghurt*)

  Pada 20 Mei 2010, Android versi 2.2 (Froyo) diluncurkan, perubahan-perubahan umum terhadap versi sebelumnya antara lain dukungan Adobe Flash 10.1, kecepatan kinerja dan aplikasi 2 sampai 5 kali lebih cepat, intergrasi V8 JavaScript *engine* yang dipakai Google Chrome yang mempercepat kemampuan *rendering* pada *browser*, pemasangan aplikasi dalam SD *Card*, kemampuan WiFi Hotspot portabel, dan kemampuan *auto update* dalam aplikasi Android Market.

- Android versi 2.3 (*Gingerbread*)

  Pada 6 Desember 2010, Android versi 2.3 (*Gingerbread*) diluncurkan, perubahan-perubahan umum yang didapat dari Android versi ini antara lain peningkatan kemampuan permainan (*gaming*), peningkatan fungsi *copy paste*, layar antarmuka (*User Interface*) didesain ulang, dukungan format video VP8 dan WebM, efek audio baru (*reverb, equalization, headphone virtualization*, dan *bass boost*), dukungan kemampuan *Near Field Communication* (NFC), dan dukungan jumlah kamera yang lebih dari satu.

- Android Versi 3.0/3.1 *Honeycomb*.

  Android *Honeycomb* dirancang khusus untuk tablet PC. Android versi ini mendukung ukuran layar yang lebih besar. *User Interface* pada *Honeycomb* juga berbeda karena sudah didesain untuk tablet. *Honeycomb* juga mendukung multiprosesor dan juga akselerasi perangkat keras (*hardware*) untuk grafis. Tablet pertama yang dibuat dengan menjalankan *Honeycomb* adalah Motorola Xoom, tablet PC ini masuk pasar Indonesia pada bulan Mei 2011.
- Android versi 4.0 (ICS :*Ice Cream Sandwich*)

  Diumumkan pada tanggal 19 Oktober 2011, membawa fitur *Honeycomb* untuk *smartphone* dan menambahkan fitur baru termasuk membuka kunci dengan pengenalan wajah, jaringan data pemantauan penggunaan dan kontrol, terpadu dengan kontak jaringan sosial, perangkat tambahan fotografi, mencari *email* secara *offline*, dan berbagi informasi dengan menggunakan NFC.

2.3 Mikrokontroler ATmega16

Mikrokontroler AVR adalah mikrokontroler RISC (*Reduce Instruction Set Compute*) 8 bit berdasarkan arsitektur *Harvard*, teknologi ini dilihat oleh Atmel pada tahun 1996. AVR mempunyai kepanjangan *Advanced Versatile* RISC atau *Alf and Vegards RISC processor* yang berasal dari dua mahasiswa Norwegian Institute of Technology (NTH), yaitu Alf-Egil Bogen dan Vegard Wollan.[4]

AVR memiliki keunggulan dibandingkan dengan mikrokontroler lain, yaitu AVR memiliki kecepatan eksekusi program lebih cepat karena sebagian besar instruksi dieksekusi dalam 1 siklus *clock*, lebih cepat dibandingkan dengan mikrokontroler MCS51 yang memiliki arsitekatur CISC (*Complex Instruction Set Compute*) di mana mikrokontroler MCS51 membutuhkan 12 siklus *clock* untuk mengeksekusi 1 instruksi. Selain itu, mikrokontroler AVR memiliki fitur yang lengkap (ADC internal, EEPROM internal, *Timer/Counter*, *Watchdog Time*r, PWM, *Port* I/O, komunikasi serial, Komparator, I2C, dll).[4]

2.3.1 Fitur ATmega16

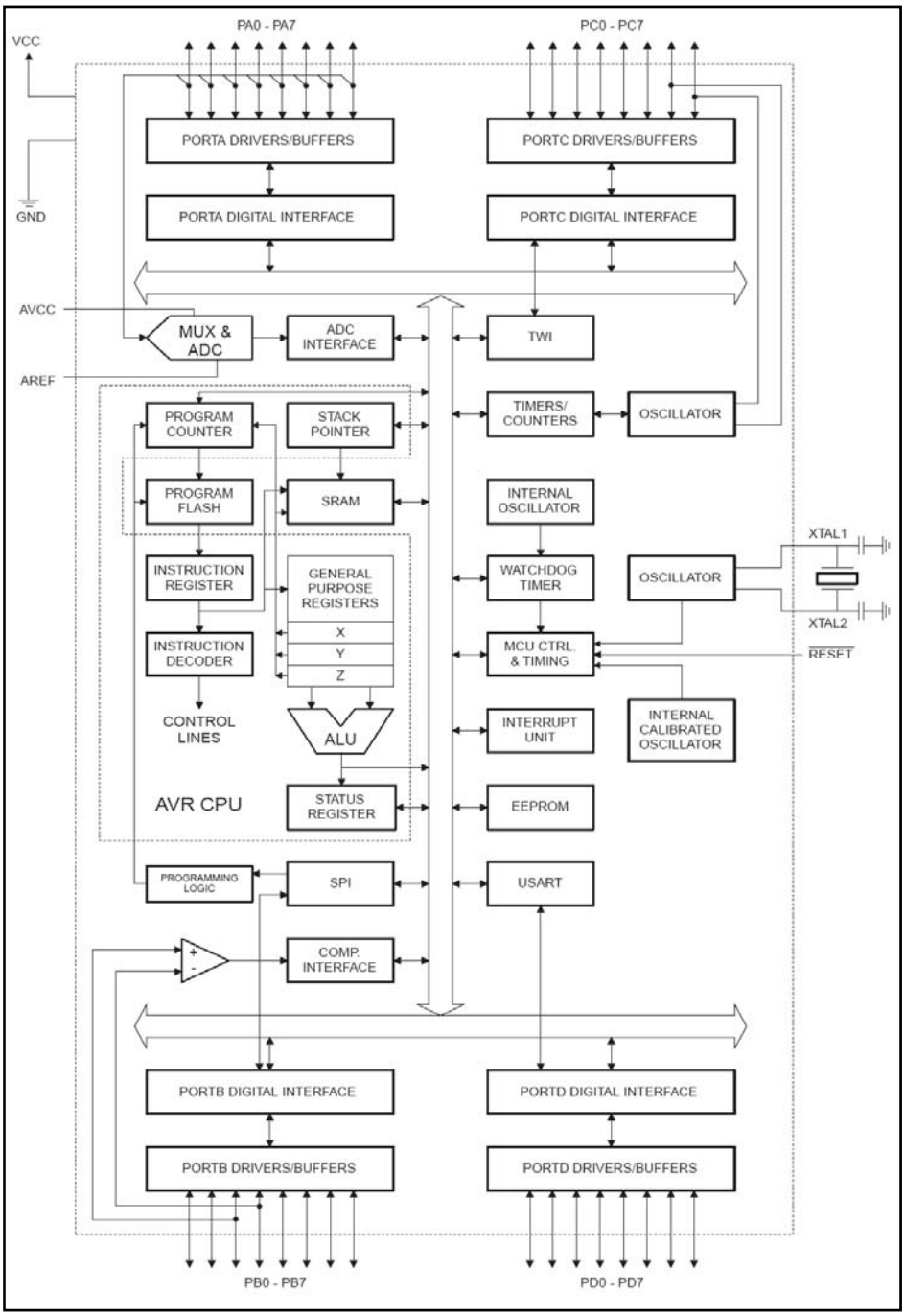
Penggunaan mikrokontroler disesuaikan dengan aplikasi yang akan dibangun, yaitu dengan menyesuaikan aplikasi dengan fitur yang dimiliki oleh mikrokontroler. Adapun fitur-fitur pada mikrokontroler ATmega16 sebagai berikut:[5]

- Mikrokontroler AVR 8 bit yang memiliki kemampuan tinggi, dengan daya rendah.
- Arsitektur RISC dengan *throughput* mencapai 16 MIPS pada frekuensi 16 MHz.
- Memiliki kapasitas memori *flash* 16 Kbyte, EEPROM 512 Byte dan SRAM 1KByte
- Saluran I/O sebanyak 32 buah, yaitu *Port* A, *Port* B, *Port* C dan *Port* D.
- CPU yang terdiri atas 32 buah register.
- Unit interupsi internal dan eksternal.
- *Port* USART untuk komunikasi serial.
- Fitur *peripheral*

  - Tiga buah *Timer/Counter* dengan kemampuan pembandingan.
    - 2 (dua) buah *Timer/Counter* 8 bit dengan *prescaler* terpisah dan mode *Compare*.
    - 1 (satu) buah *Timer/Counter* 16 bit dengan *prescaler* terpisah, mode *Compare*, dan mode *Capture*.
  - *Real time Counter* dengan *Oscillator* tersendiri.
  - 4 *channel* PWM.
  - 8 *Channel*, 10-bit ADC.
    - 8 *single-ended Channel*.
    - 7 *Differential Channel* hanya pada kemasan TQFP.
    - 2 *Differential Channel* dengan *Programmable Gain* 1x, 10x, atau 20x.
  - *Byte-oriented Two-wire Serial Interface.*
  - *Programmable Serial* USART.
  - Antarmuka SPI.
  - *Watchdog Timer* dengan *oscillator* internal.
  - *On-chip Analog Comparator*.

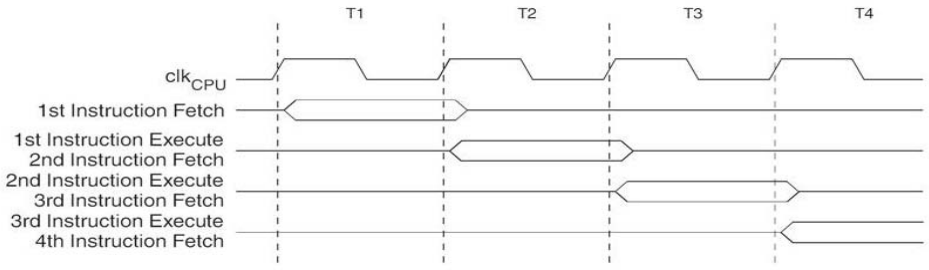
2.3.2 Blok Diagram ATmega16

Pada Gambar 2.2 merupakan blok diagram AVR ATmega16 yang menggunakan arsitektur *Harvard* yaitu dengan memisahkan antara bus memori untuk program dan data yang dapat memaksimalkan kemampuan dan kecepatan. (Sumber: Andrianto, 2008, hlm. 8)



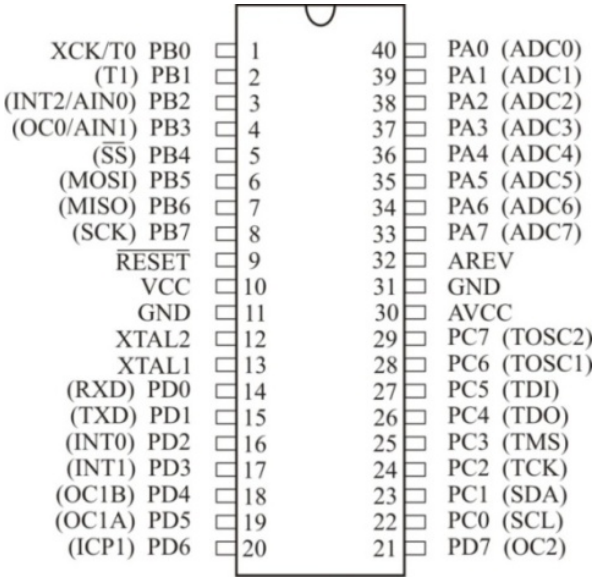
Gambar 2.2. Blok Diagram ATmega16

Instruksi dalam memori program dieksekusi dengan *pipelining single level*. Di mana ketika satu instruksi dieksekusi, instruksi berikunya diambil dari memori program. Konsep ini mengakibatkan instruksi dieksekusi setiap *clock cycle*. CPU terdiri dari 32x8 bit, *General Prupose Register* (GPR) dapat diakses dengan cepat dalam satu *clock cycle*, yang mengakibatkan operasi *Arithmetic Logic Unit* (ALU) dapat dilakukan dalam satu *cycle*. Pada operasi ALU, dua *operand* berasal dari register, kemudian operasi dieksekusi dan hasilnya disimpan kembali pada register dalam satu *clock cycle*. Oparasi aritmatika dan logika pada ALU akan mengubah bit-bit yang terdapat pada *Status Register* (SREG). Proses pengambilan instruksi dan pengeksekusian instruksi berjalan secara paralel diilustrasikan seprti yang terlihat pada Gambar 2.3.[5]. (Sumber: Andrianto, 2008, hlm. 10)



Gambar 2.3. Pengambilan Instruksi dan Pengeksekusian Secara Paralel

2.3.3 Konfigurasi Pin ATmega16



Gambar 2.4. Konfigurasi Pin ATmega16

Konfigurasi pin ATmega16 dengan kemasan 40 pin DIP (*Dual In-lin Package*) dapat dilihat pada Gambar 2.4. Dari Gambar 2.4 dapat dijelaskan fungsi dari masing-masing pin ATmega16 sebagai berikut:[5]

- VCC merupakan pin yang berfungsi sebagai masukan catu daya.
- GND merupakan pin Ground.
- *Port* A (PA0:PA7) merupakan pin *input/output* dua arah dan memiliki fungsi khusus yaitu sebagai pin masukan ADC 8 kanal.
- *Port* B (PB0:PB7) merupakan pin *input/output* dua arah dan pin fungsi khusus seperti pada Tabel 2.1. (Sumber: Andrianto, 2008, hlm. 6)

Tabel 2.1. Fungsi Khusus Port B

Pin	Fungsi Khusus
PB7	SCK (SPI *Bus Serial Clock*)
PB6	MISO (SPI *Bus Master Input/Slave Output*)
PB5	MOSI (SPI *Bus Master Output/Slave Input*)
PB4	SS (SPI *Slave Select Input*)
PB3	AIN1 (*Analog Comparator Negative Input*) OC0 (*Timer/Counter0 Output Compare Match Output*)
PB2	AIN0 (*Analog Comparator Positive Input*) INT2 (*External Interrupt 2 input*)
PB1	T0 T1 (*Timer/Counter1 External Counter Input*)
PB0	T0 T1 (*Timer/Counter0 External Counter Input*) XCK (*USART External Clock Input/Output*)

- *Port* C (PC0...PC7) merupakan pin *input/output* dua arah dan pin fungsi khusus, adapun fungsi khusus yang dimiliki *port* C seperti yang terlihat pada Tabel 2.2. (Sumber: Heri Andrianto, (2008), hlm. 6.)

Tabel 2.2. Fungsi Khusus Port C

Pin	Fungsi Khusus
PC7	TOSC2 (*Timer Oscillator Pin2*)
PC6	TOSC1 (*Timer Oscillator Pin1*)
PC5	TDI (JTAG *Test Data In*)
PC4	TDO (JTAG *Test Data Out*)
PC3	TMS (JTAG *Test Mode Select*)
PC2	TCK (JTAG *Test Clock*)
PC1	SDA (*Two-wire Serial Bus Data Input/Output Line*)
PC0	SCL (*Two-wire Serial Bus Clock Line*)

- *Port* D (PD0..PD7) merupakan pin *input/output* dua arah dan pin fungsi khusus, seperti yang terlihat pada Tabel 2.3. (Sumber: Andrianto, 2008, hlm. 7)

Tabel 2.3. Fungsi Khusus Port D

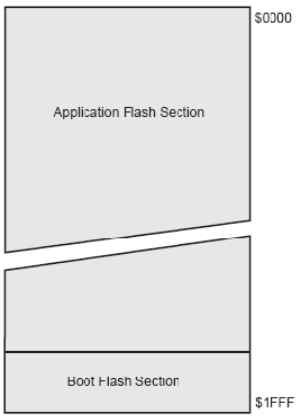
Pin	Fungsi Khusus
PD7	*OC2 (Timer/Counter2 Output Compare Match Output)*
PD6	*ICP (Timer/Counter1 Input Capture Pin)*
PD5	*OC1A (Timer/Counter1 Output Compare A Match Output)*
PD4	*OC1B (Timer/Counter1 Output Compare B Match Output)*
PD3	*INT1 (External Interrupt 1 Input)*
PD2	*INT0 (External Interrupt 0 Input)*
PD1	*TXD (USART Output Pin)*
PD0	*RXD (USART Input Pin)*

- RESET merupakan pin yang digunakan untuk me*reset* mikrokontroler.
- XTAL1 dan XTAL2 merupakan pin masukan *clock* eksternal.
- AVCC merupakan pin masukan tegangan untuk ADC.
- AREF merupakan pin masukan tegangan referensi ADC.

2.3.4 Pemetaan Memori ATmega16

Arsitektur AVR mempunyai dua memori utama, yaitu memori program dan memori data (SRAM). Selain itu, ATmega16 juga memilliki memori EEPROM untuk menyimpan data.[5]

a. Memori Program

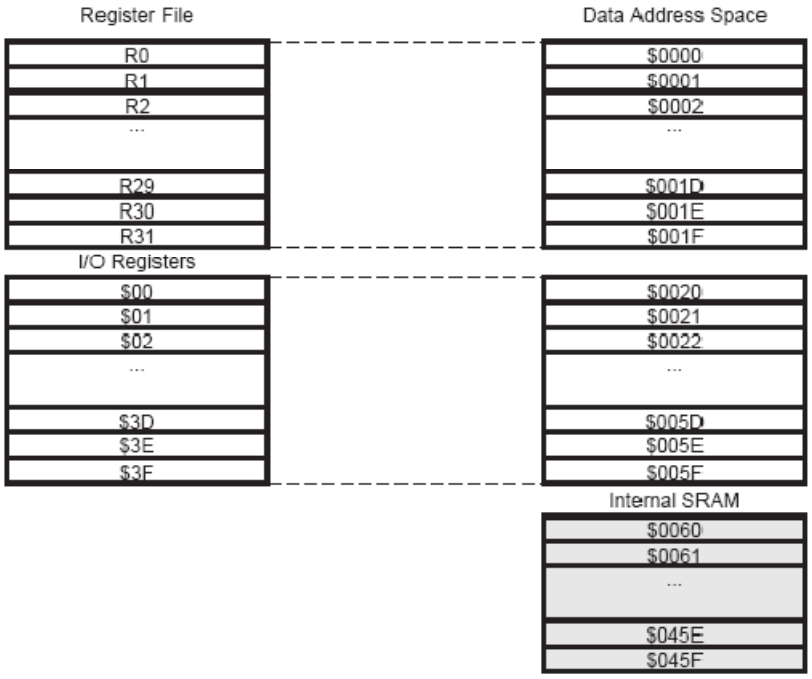


Gambar 2.5. Pemetaan Memori Program

ATmega16 memiliki 16K byte *Onchip In-System Programmable Flash memory* untuk menyimpan program, Pemetaan memori ATmega16 dapat dilihat pada Gambar 2.5. Untuk keamanan program, memori program, *flash* dibagi ke dalam dua bagian, yaitu bagian program *Boot* dan aplikasi. *Bootloader* adalah program kecil yang bekerja pada saat *start up time* yang dapat memasukan seluruh program aplikasi ke dalam memori prosesor. (Sumber: Andrianto, 2008, hlm. 13)

b. Memori Data

Memori data AVR ATmega16 terbagi menjadi 3 bagian, yaitu 32 buah register umum, 64 buah register I/O dan 1 Kbyte SRAM internal. *General Puspose Register* menempati alamat data terbawah, yaitu $00 sampai $1F. sedangkan memori I/O menempati 64 alamat berikutnya mulai dari $20 hingga $5F. Memori I/O merupakan register yang khusus digunakan untuk mengatur fungsi terhadap berbagai *peripheral* mikrokontroler seperti kontrol register, *timer/counter*, fungsi-fungsi I/O, dan sebagainya. 1024 alamat memori berikutnya mulai alamat $60 hingga $45F digunakan untuk SRAM internal. (Sumber: Andrianto, 2008, hlm. 14)



Gambar 2.6. Pemetaan Memori Data (SRAM)

c. Memori EEPROM

ATmega16 terdiri dari 512 Byte memori data EEPROM 8 bit, data dapat ditulis/baca dari memori ini, ketika catu daya dimatikan, data terakhir yang ditulis pada memori EEPROM masih tersimpan pada memori ini, atau dengan kata lain memori EEPROM bersifat *nonvolatile*. Alamat EEPROM mulai #000 sampai &1FF.

2.3.5 Register ATmega16

Mikrokontroler ATmega16 yang memiliki 32 register merupakan register-register yang harus dikonfigurasi agar proses kerja mikrokontroler sesuai dengan tujuan pemrograman. Tidak semua register ATmega16 dijelaskan disini, akan tetapi hanya register-register yang berhubungan dengan pembuatan sistem saja. Adapun beberapa register yang perlu diketahui diantarnya yaitu sebagai berikut:[5]

1. Register SREG

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 2.7. Register SREG

Status register berisi status yang dihasilkan pada setiap operasi yang dilakukan ketika suatu instruksi dieksekusi. SREG merupakan bagian dari inti CPU mikrokontroler, register SREG diilustrasikan seperti pada Gambar 2.7. Adapun bit penyusun pada register SREG adalah sebagai berikut:[5] (Sumber: Andrianto, 2008, hlm. 16)

a. Bit 7: I (*Global Interrupt Enable*) bit yang berfungsi untuk mengaktifkan interupsi.
b. Bit 6: T (*Bit Copy Storage*) bit yang berfungsi sebagai pemindahan data dari register dengan menggunakan instruksi BLD dan BST.
c. Bit 5: H (*Half Carry Flag*) bit yang berfungsi sebagai *Half* Carry dari operasi aritmatika.

d. Bit 4: S (*Sign Bit*) bit S merupkan hasil operasi EOR antara *flag*-N (*negative*) dan *flag* V (komplemen dua *overflow*).
e. Bit 3: V (*Two's Complement Overflow Flag*) bit guna untuk mendukung operasi aritmatika.
f. Bit 2: N(*Negative Flag*) apabila suatu operasi menghasilkan bilangan *negative*, maka *flag* N akan di-*set*.
g. Bit 1: Z (*Zero Flag*) bit akan di-*set* bila hasil operasi yang diperoleh adalah nol.
h. Bit 0: C (*Carry Flag*) bit akan di-*set* bila hasil operasi menghasilkan *carry*.

2. Register SP

Bit	15	14	13	12	11	10	9	8	
	SP15	SP14	SP13	SP12	SP11	SP10	SP9	SP8	SPH
	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0	SPL
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
	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	
	0	0	0	0	0	0	0	0	

Gambar 2.8. Register Stack Pointer

Pada Gambar 2.8 merupakan ilustrasi register *satck*. *Stack* merupakan register 16 bit yang fungsi utamanya digunakan untuk menyimpan data sementara, dan menyimpan *return address* setelah interupsi dan pemanggilan subrutin. Untuk menyimpan data ke dalam *stack* menggunakan instruksi PUSH dan untuk mengambil data kembali digunakan insruksi POP, penyimpanan data ke dalam *stack* akan mengurangi kapasitas *stack*.[5] (Sumber: Andrianto, 2008, hlm. 12)

3. Register UCSRA

Bit	7	6	5	4	3	2	1	0	
	RXC	TXC	UDRE	FE	DOR	PE	U2X	MPCM	UCSRA
Read/Write	R	R/W	R	R	R	R	R/W	R/W	
Initial Value	0	0	1	0	0	0	0	0	

Gambar 2.9. Register UCSRA

Register UCSRA digunakan untuk pengaturan *flag* komunikasi serial, ilustrasi register UCSRA dapat dilihat pada Gambar 2.9. Adapun bit penyusun pada

regsiter UCSRA dapat dijelaskan sebagai berikut. (Sumber: Andrianto, 2008, hlm. 117)

a. Bit RXC akan diset 1 jika telah diterima 8 bit data serial dan di-*clear* ketika penyangga penerima kosong.
b. Bit TXC akan diset 1 ketika melakukan *shift* pada data serial, akan di-*clear* secara otomatis ketika selasai pengiriman.
c. Bit UDRE akan bernilai 1 jika isi register UDR kosong.
d. Bit FE berfungsi untuk mendeteksi *frame error*.
e. Bit DOR berfungsi sebagai pendeteksi adanya data *OverRun*.
f. Bit PE berfungsi sebagai pendeteksi adanya paritas *error* pada 1 *frame*.
g. Bit U2X jika di-*set* 1 maka kecepatan *baud rate* dilipatgandakan.
h. Bit MPCM jika di-*set* 1 akan mengaktifkan multiprosesor.

4. Register UCSRB

Bit	7	6	5	4	3	2	1	0	
	RXCIE	TXCIE	UDRIE	RXEN	TXEN	UCSZ2	RXB8	TXB8	UCSRB
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Gambar 2.10. Register UCSRB

Gambar 2.10 merupakan isi dari register UCSRB. Register UCSRB berfungsi sebagai pengaturan komunikasi serial. Adapun bit penyusun pada register UCSRB adalah sebagai berikut. (Sumber: Andrianto, 2008, hlm. 118)

a. Bit RXCIE jika bernilai 1 akan mengaktifkan interupsi penerimaan data serial berdasar kondisi bit RXC.
b. Bit TXCIE jika di-*set* 1 akan mengaktifkan interupsi pengiriman data serial berdasar kondisi bit TXC.
c. Bit UDRIE jika di-*set* 1 akan mengaktikan interupsi kondisi bit UDRE.
d. Bit RXEN jika di-*set* 1 maka penerimaan data serial di-*enable*-kan.
e. Bit TXEN jika di-*set* 1 maka pengiriman data serial di-*enable*-kan.
f. Bit UCSZ2 berfungsi sebagai pengatur ukuran jumlah bit data serial yang bekerja bersama-sama dengan UCSZ1 dan UCSZ0 pada register UCSRC.

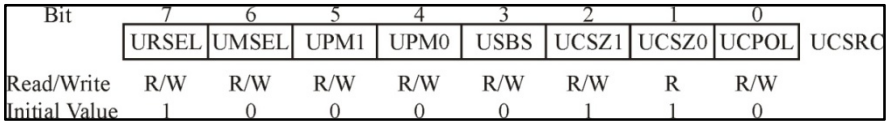
Adapun nilai UCSZ dalam menentukan ukuran karakter seperti yang terlihat pada Tabel 2.4. (Sumber: Andrianto, 2008, hlm. 118)

Tabel 2.4. Penentuan Ukuran Karakter

UCSZ [2..0]	Ukuran Data dalam Bit
000	5
001	6
010	7
011	8
100-110	Tidak dipergunakan
111	9

g. Bit RXB8 berfungsi untuk menerima bit yang ke-9 data serial.
h. Bit TXB8 berfungsi menampung bit yang ke-9 untuk untuk dikirimkan.

5. Register UCSRC



Gambar 2.11 Register UCSRC

Register UCSRC merupakan register yang digunakan untuk mengatur mode dan kecepatan komunikasi serial yang dilakukan, bit pada register UCSRC dapat diilustrasikan pada Gambar 2.11. Adapun bit penyusun pada register UCSRC adalah sebagai berikut. (Sumber: Andrianto, 2008, hlm. 119)

a. Bit URSEL merupakan bit pemilih akses antara UCSRC dan UBRR.
b. Bit UMSEL merupakan bit pemilih mode komunikasi serial antara sinkron dan asinkron. Pada Tabel 2.5 merupakan konfigurasi bit UMSEL. (Atmel, 2007, hlm. 166)

Tabel 2.5. Penentuan Mode Operasi

UMSEL	Mode
0	Asynchronous Operation
1	Synchronous Operation

c. Bit UPM[1:0] merupakan bit pengatur paritas. Untuk mengatur paritas dalam komunikasi serial, maka nilai UPM dapat dikonfigurasi seperti yang terlihat Tabel 2.6. (Sumber: Atmel, 2007, hlm. 166)

Tabel 2.6. Penentuan Mode Paritas

UPM1	UPM0	Parity Mode
0	0	*Disabled*
0	1	*Reserved*
1	0	*Enabled, Even Parity*
1	1	*Enabled, Odd Parity*

d. Bit UBSS merupakan bit pemilih ukuran bit stop. Nilai *default* adalah nol sehingga jumlah bit stop yaitu 1 bit. Pada Tabel 2.7 merupakan penentuan jumlah bit stop. (Sumber: Atmel, 2007, hlm. 166)

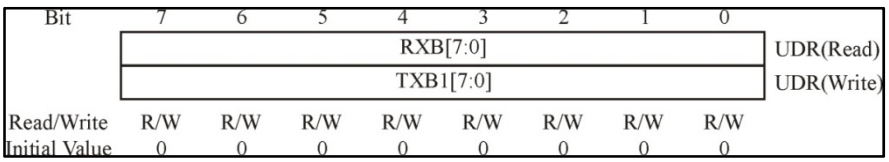
Tabel 2.7. Penentuan Jumlah Bit Stop

USBS	Stop Bit(s)
0	1-bit
1	2-bit

e. Bit UCSZ[1:0] merupakan bit pengatur jumlah bit karakter pengiriman data serial.

f. Bit UCPOL merupakan bit pengatur hubungan antara perubahan data keluaran dan data masukan serial dengan *clock* sinkronisasi. Hanya berlaku untuk mode sinkron. Untuk mode asinkron, bit ini di-*set* 0.

6. Register UDR

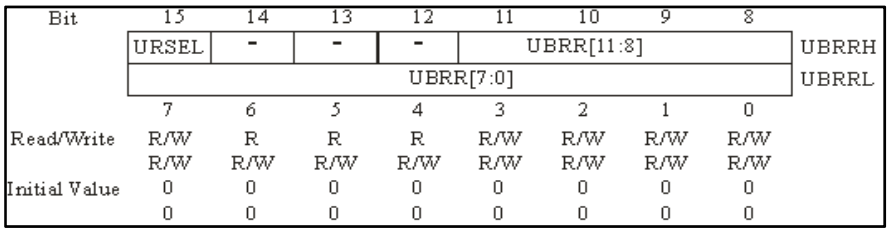


Gambar 2.12. Register UDR

Register UDR merupakan register yang digunakan sebagai penyangga dalam pengiriman maupun penerimaan data serial. Penyangga (*buffer*) pengirim dan penerima memiliki nama yang sama akan tetapi memiliki lokasi yang berbeda. Sedangkan Penulisan data ke register UDR berarti akan mengirimkan data secara serial, dan pembacaan register UDR berarti menerima data serial. Pada

Gambar 2.12 merupakan ilustrasi register UDR. (Sumber: Atmel, 2012, hlm. 163)

7. Register UBRR



Gambar 2.13. Register UBRR

Register UBRR seperti yang terlihat pada Gambar 2.13, merupakan register 16 bit, yang berfungsi untuk menentukan kecepatan transmisi data yang akan digunakan. UBRR dibagi menjadi dua yaitu UBRRH dan UBRRL. UBRR[11:0] merupakan bit penyimpan konstanta kecepatan komunikasi serial. UBRRH menyimpan 4 bit tertinggi, dan UBRRL menyimpan 8 bit sisanya. Data yang dimasukan ke UBRRH dan UBRL dapat dihitung dengan persamaan seperti pada Tabel 2.8. (Sumber: Andrianto, 2008, hlm. 117)

Tabel 2.8. Rumus Perhitungan UBRR

Mode Operasi	Persamaan Nilai UBRR
Asinkron mode kecepatan normal (U2X=0)	$UBRR = \frac{f_{osc}}{16 \times Baud\ rate} - 1$
Asinkron mode kecepagan ganda (U2X=1)	$UBRR = \frac{f_{osc}}{8 \times Baud\ rate} - 1$
Sinkron	$UBRR = \frac{f_{osc}}{2 \times Baud\ rate} - 1$

2.3.6 Alamat Vektor Interupsi ATmega16

Mikrokontroler ATmega16 memiliki 21 alamat vektor interupsi, *Program Counter* (PC) akan menunjuk ke alamat vektor interupsi yang ditunjuk jika terjadi interupsi. Adapun 21 alamat vektor interupsi tersebut seperti pada Tabel 2.9.[5] (Sumber: Andrianto, 2008, hlm. 163)

Tabel 2.9. Alamat Vektor Interupsi

No. Vektor	Alamat Program	Sumber Interrupt	Keterangan
1	$000(1)	RESET	*Esternal Pin, Power-on Reset, Brown-out Reset, Watchdod Reset, and JTAG AVR Reset.*
2	$002	INT0	*External Interrupt Request 0*
3	$004	INT1	*External Interrupt Request 1*
4	$006	TIMER2 COMP	*Timer/Counter 2 Compare Match.*
5	$008	TIMER2 OVF	*Timer/Counter2 Overflow*
6	$00A	TIMER1 CAPT	*Timer/Counter1 Event*
7	$00C	TIMER1 COMPA	*Timer/Counter Compare Match A*
8	$00E	TIMER1 COMPB	*Timer/Counter1 Compare Match B*
9	$010	TIMER1 OVF	*Timer/Counter1 Overflow*
10	$012	TIMER0 OVF	*Timer/Counter0 Overflow*
11	$014	SPI, STC	*Serial Transfer Complete*
12	$016	USART, RXC	*USART, Rx Complete*
13	$018	USART, UDRE	*Usart Data Registry Empty*
14	$01A	USART, TXC	*USART, Tx Complete*
15	$01C	ADC	*ADC Conversion Complete*
16	$01E	EE_RDY	*EEPROM Ready*
17	$020	ANA_COMP	*Analog Comparator*
18	022	TWI	*Two-wire Serial Interface*
19	$024	INT2	*External Interrupt Request 2*
20	$026	TIMER0 COMP	*Timer/Counter0 Compare Match*
21	$028	SPM_RDY	*Store Program Memory Ready*

2.4 Komunikasi Serial RS-232

Standar sinyal komunkasi serial yang banyak digunakan adalah standar RS-232 yang dikembangkan oleh *Electronic Industry Association and Telecomunications Industry Association* (EIA/TIA) yang pertama kali dipublikasikan pada tahun 1962. Standar ini hanya menyangkut komunikasi data anatara komputer (*Data Terminal Equipment* – DTE) dengan alat-alat pelengkap komputer (*Data Circuit-Termintaing Equipment* DCE).[6]

Pada standar RS-232 logika satu “1” disebut dengan *mark*, memiliki besar tegangan antara -3V sampai dengan -25V, sedangkan logika nol “0” disebut dengan *space* yang memiliki tegangan +3V sampai dengan +25V, besar tegangan antara -3V samapai dengan +3V akan menghasilkan nilai yang tidak terdefinisi. Dengan demikian antarmuka *port* serial komputer tidak dapat langsung

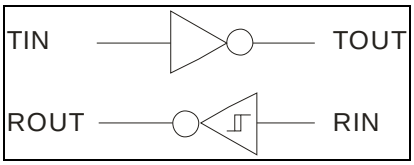
dihubungkan dengan perangkat TTL yang memiliki tegangan 0V sampai dengan 5V, sehingga untuk keperluan ini diperlukan konverter. Salah satu IC yang dapat digunakan untuk konverter komunikasi serial RS-232 menjadi serial TTL adalah IC MAX232.[6]

2.4.1 Deskripsi IC MAX232

IC MAX232 merupakan IC konverter RS-232 level EIA/TIA menjadi level TTL/CMOS atau sebaliknya, IC ini menyediakan dua buah *driver/receiver* yang dapat membangkitkan tegangan RS-232 standar level EIA dan tegangan standar level TTL.[7]

Bagian *Receiver* RS-232 pada MAX232 akan mengonversi tegangan *input* level EIA-232 menjadi level TTL/CMOS yaitu 0V sampai dengan 5V, besar tegangan *input* RS-232 yang dapat diterima oleh MAX232 mencapai ±30V. Begitu pula sebaliknya dengan *receiver* TTL/CMOS, bagian ini akan mengonversi tegangan 0V sampai dengan 5V menjadi level EIA-232.[7]

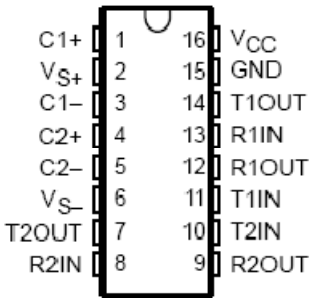
Logika *output* pada IC MAX232 merupakan invers dari logika *input*, pada Gambar 2.14 merupakan satu buah logika *driver/reciver* yang dimiliki oleh IC MAX232.



Gambar 2.14. Logika MAX232

2.4.2 Konfigurasi Pin IC MAX232

IC MAX232 memiliki 16 pin DIP, adapun konfigurasi pin yang dimiliki oleh IC MAX232 seperti yang terlihat pada Gambar 2.15. (Sumber: Texas Instruments Incorporated, 2004, hlm. 1)



Gambar 2.15. Konfigurasi Pin IC MAX232

Fungsi dari masing-masing pin adalah sebagai berikut:[7]

a. VCC, sebagai tegangan kerja untuk tegangan positif.
b. GND, sebagai tegangan kerja untuk ground.
c. C+, C-, sebagai *voltagepump*.
d. VS+, VS-, sebagai pembangkit tegangan positif dan tegangan negatif.
e. TIN, TOUT, sebagai *input* TTL dan *output* EIA-232.
f. RIN, ROUT, sebagi *input* EIA-232 dan *output* TTL.

2.5 Komunikasi Serial RS-485

Standar RS-485 mendukung komunikasi dua arah bergantian (*half duplex*) dan dua arah penuh (*full duplex*). Sistem *half duplex* hanya memerlukan 2 buah kabel (untuk sinyal A dan B) di mana sinyal A dan B untuk *driver* dan penerima dihubungkan secara paralel, sedangkan sistem *full duplex* memerlukan 4 buah kabel, masing-masing 2 untuk sinyal A dan B, pengirim dan penerima yang terpisah.[8]

Penerima RS-485 bekerja dengan membaca perbedaan tegangan antara sinyal A dan sinyal B. Jika sinyal A lebih besar minimal 200mV dari sinyal B, maka keluaran pada penerima akan berlogika tinggi. Sebaliknya, jika sinyal B lebih besar minimal 200mV dari sinyal A, keluaran penerima akan berlogika rendah. Perbedaan tegangan di bawah 200mV akan mengakibatkan keluaran yang tidak terdefinisi pada keluaran penerima. Salah satu IC yang dapat digunakan sebagai konversi komunikasi serial TTL menjadi serial RS-485 yaitu IC MAX488.

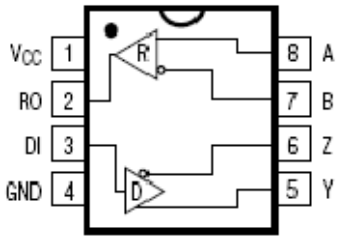
2.5.1 Deskripsi IC MAX488

IC MAX488 merupakan keluarga IC transceiver RS-422 dan RS-485, MAX488 dapat dihubungkan dengan *unit load* sebanyak 32 buah *transceiver*. Komunikasi yang disediakan oleh MAX488 yaitu komunikasi serial *Full Duplex*.[9]

Terminal A dan terminal B pada kondisi *short circuit* memiliki impedansi resistansi sebesar 12KΩ, dan pada kondisi *open circuit* tegangan antara terminal A dan terminal B adalah 0.2V, sehingga status *output* dapat diketahui yaitu berlogika satu "1".[9]

2.5.2 Konfigurasi Pin IC MAX488

Konfigurasi pin yang dimiliki oleh IC MAX488 seperti yang terlihat pada Gambar 2.16. [9] (Sumber: Maxim, 2009, hlm. 7)



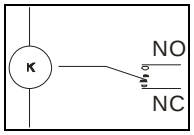
Gambar 2.16. Konfigruasi Pin IC MAX488

Adapun fungsi dari masing-masing pin IC MAX488 yaitu sebagai berikut:[7]

a. RO: berfungsi sebagai *output* serial level TTL.
b. DI: berfungsi sebagai *input* serial level TTL.
c. A: berfungsi sebagai sinyal penerima *noninverting* atau sinyal D+.
d. B: berfungsi sebagai sinyal penerima *inverting* atau sinyal D-.
e. Y: berfungsi sebagai sinyal pengirim *noninverting* atau sinyal D+
f. Z: berfungsi sebagai sinyal pengirim *inverting* atau sinyal D-.
g. VCC: berfungsi sebagai tegangan kerja IC MAX487, tegangan positif.
h. GND: berfungsi sebgai tegangan kerja IC MAX487, tegangan ground.

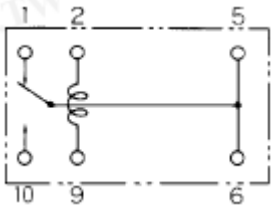
2.6 Relay

Relay merupakan saklar elektromagnetik yang prinsip kerjanya menggunakan azas kumparan listrik. Jika kumparan tersebut mendapat aliran listrik maka posisi saklar akan berubah dari *normaly close* (NC) menjadi *normaly open* (NO). Pada Gambar 2.17 merupakan simbol relay.



Gambar 2.17. Simbol Relay

Relay SY-5-K merupakan salah satu jenis relay yang memiliki resistansi kumparan sebesar 167Ω. Relay SY-5-K dapat bekerja jika kumparan diberi tegangan sebesar +5V dengan arus sebesar 30mA. Konfigurasi pin relay SY-5-K seperti yang terlihat pada Gambar 2.18.[10] (Sumber: Fujitsu, hlm. 7)

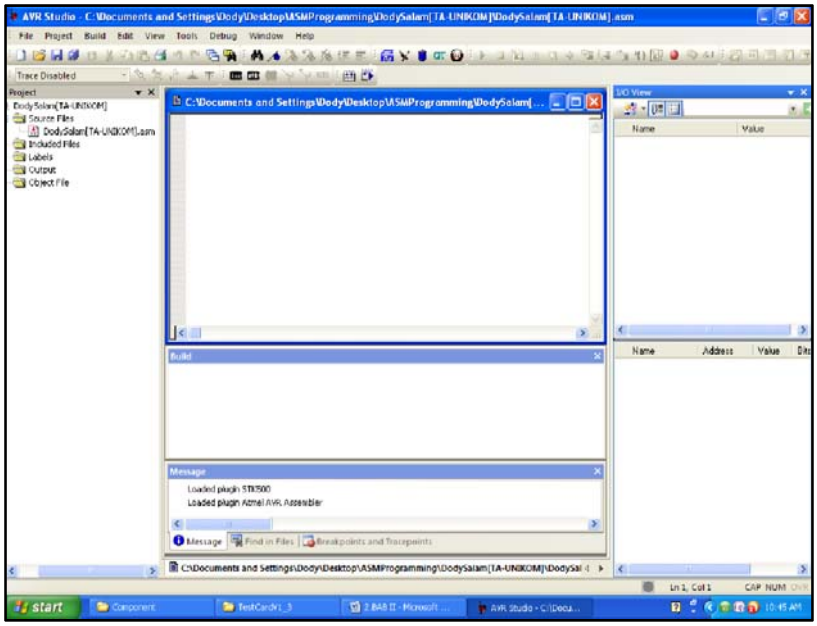


Gambar 2.18. Konfigurasi Pin Relay SY-5-K

2.7 Bahasa Pemrograman Assembly

2.7.1 IDE AVR Studio 4

Compiler AVR Studio 4 adalah perangkat lunak yang digunakan dalam menulis bahasa program *Assembly* untuk memprogram mikrokontroler keluarga AVR, dengan menggunakan *compiler* tersebut akan dihasilkan file dengan ekstensi .hex, file ini merupakan file yang nantinya akan di-*upload* ke dalam memori mikrokontroler. Pada Gambar 2.19 merupakan tampilan jendela kerja *compiler* AVR Studio 4.



Gambar 2.19. Jendela Kerja AVR Studio 4

2.7.2 Instruksi Assembly

Dalam menjalankan program, mikrokontroler akan melakukan pembacaan data yang tersimpan dalam memori program (internal atau eksternal). Alamat memori yang harus dibaca disimpan dalam sebuah register yang dinamakan *Program Counter* (PC). Data yang terbaca akan diartikan sebagai perintah atau instruksi yang harus dikerjakan. Pada mikrokontroler keluarga AVR terdapat empat jenis instruksi yang sering digunakan yaitu pemindahan data (*data transfer*), pemrosesan data (*data processing*), pengaturan alur program (*program control*), dan instruksi operasi bit.

a. Instruksi Aritmatika

   Instruksi aritmatika meliputi perkalian, pembagian, dan pengurangan serta operasi logika, pada Tabel 2.10 merupakan instruksi aritmatika yang terdapat pada mikrokontroler AVR.

Tabel 2.10. Instruksi Aritmatika

Mnemonic	Deskripsi
ADD	*Add two registers*
ADC	*Add with cary two registers*
ADIW	*Add immediate to word*
SUB	*Subtract two registers*
SUBI	*Subtract constant form register*
SBC	*Subtract with carry two registers*
SBCI	*Subtract with carry constant from register*
SBIW	*Subtract immediate from word*
AND	*Logical AND registers*
ANDI	*Logical AND register and constant*
OR	*Logical OR registers*
ORI	*Logical OR register and constant*
EOR	*Exclusive OR registers*
COM	*One's complement*
NEG	*Two's complement*
SBR	*Set bit(s) in register*
CBR	*Clear bit(s) in register*
INC	*Increment*
DEC	*Decrement*
TST	*Test for zero or minus*
CLR	*Clear register*
SER	*Set register*
MUL	*Multiply unsigned*
MULS	*Multiply signed*
MULSU	*Multiply signed with unsigned*
FMUL	*Fractional multiply unsigned*
FMULS	*Fractional multiply signed*
FMULSU	*Fractional multiply signed with unsigned*

b. Instruksi Pemindahan Data

Mikrokontroler AVR memiliki instruksi pemindahan data yang sangat berguna untuk memindahkan data byte. Sumber atau tujuan dapat berupa data konstanta, register fungsi khusus, dan data di memori program atau memori data. Instruksi pemindahan data disajikan pada Tabel 2.11.

Tabel 2.11. Instruksi Pemindahan Data

Mnemonic	Deskripsi
MOV	*Mov between registers*
MOVW	*Copy register word*
LDI	*Load immediate*
LD	*Load indirect*
LDD	*Load indirect with displacement*
LDS	*Load direct from SRAM*
ST	*Store indirect*
STD	*Store indirect with displacement*
STS	*Store direct to SRAM*
LPM	*Load Program Memory*
SPM	*Store Program Memory*
IN	*In Port*
OUT	*Out Port*
PUSH	*Push register on stack*
POP	*Push register from stack*

c. Instruksi Bit

Instruksi dalam pengoperasian bit pada mikrokontroler AVR meliputi instruksi logika perbit, rotasi bit, dan pertukaran *nibble*. Pada Tabel 2.12 merupakan instruksi bit pada mikrokontroler AVR.

Tabel 2.12. Instruksi Bit

Mnemonic	Deskripsi
SBI	*Set bit in I/O register*
CBI	*Clear bit in I/O register*
LSL	*Logical Shift Left*
LSR	*Logical Shift Right*
ROL	*Rotate Left Through Carry*
ROR	*Rotate Right Through Carry*
ASR	*Arithmatic Shift Right*
SWAP	*Swap Nibbles*
BSET	*Flag Set*
BCLR	*Flag Clear*
BST	*Bit Store Form register to T*
BLD	*Bit Load from T to register*
SEC	*Sect Carry*
CLC	*Clear Carry*
SEN	*Set Negatif Flag*
CLN	*Clear Negatif Flag*
SEZ	*Set Zero Flag*
CLZ	*Clear Zero Flag*
SEI	*Global Interrupt Enable*
CLI	*Global Interrup Disable*
SES	*Set Signed Test Flag*
CLS	*Clear Signed Test Flag*
SEV	*Set Twos Complement Overflow*
CLV	*Clear Twos Complement Overflow*
SET	*Set T in SREG*
CLT	*Clear T in SREG*
SEH	*Set Half Carry Flag in SREG*

d. Instruksi Percabangan

Instruksi percabangan yaitu instruksi yang meliputi instruksi lompat tak bersyarat dan lompat bersyarat. Adapun instruksi percabangan seperti yang terlihat pada Tabel 2.13.

Tabel 2.13. Instruksi Percabangan

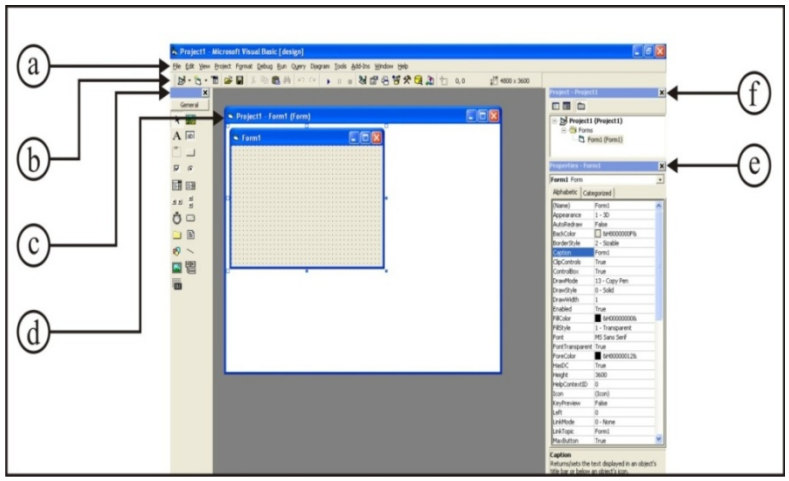
Mnemonic	Deskripsi
RJMP	*Relative Jump*
IJMP	*Indirect Jump to (Z)*
JMP	*Direct Jump*
RCALL	*Relative Subroutine Call*
ICALL	*Indirect call to (Z)*
CALL	*Direct Subroutine Call*
RET	*Subroutine Return*
RETI	*Interrupt Return*
CPSE	*Compare, Skip if Equal*
CP	*Compare*
CPC	*Compare with Carry*
CPI	*Compare register with Immediate*
SBRC	*Skip if Bit in Register Cleared*
SBRS	*Skip if Bit in Register is Set*
SBIC	*Skip if Bit in I/O Register Cleared*
SBIS	*Skip if Bit in I/O Register is Set*
BRBS	*Branch if Status Flag Set*
BRBC	*Branch if Status Flage Cleared*
BREQ	*Branch if Equal*
BRNE	*Branch if Not Equal*
BRCS	*Branch if Carry Set*
BRCC	*Branch if Carry Cleared*
BRSH	*Branch if Same or Higher*
BRLO	*Branch if Lower*
BRMI	*Branch if Minus*
BRPL	*Branch if Plus*
BRGE	*Branch if Greater or Equal, Signed*
BRLT	*Branch if Less than Zero, Signed*
BRHS	*Branch if Half Carry Flag Set*
BRHC	*Branch if Half Carry Flag Cleared*
BRTS	*Branch if T Flag Set*
BRTC	*Branch if T Flag Cleared*
BRVS	*Branch if Overflow Flag is Set*
BRVC	*Branch if Overflow Flag is Cleared*

2.8 Bahasa Pemrograman Visual Basic

Visual Basic adalah sebuah bahasa pemrograman yang digunakan untuk membuat program aplikasi berbasis orientasi objek atau *Object Oriented Program* (OOP). Visual Basic merupakan pengembangan dari bahasa Basic. Dengan Visual

Basic, perancangan sebuah program aplikasi akan lebih mudah karena didukung oleh komponen-komponen pelengkap yang memiliki standar Windows.[11]

2.8.1 Lingkungan Kerja Visual Basic



Gambar 2.20. Jendela Kerja Visual Basic

Seperti pada Gambar 2.20 fungsi dari masing-masing bagian jendela kerja Visual Basic dapat dijelaskan sebagai berikut:[11]

a. MenuBar

   MenuBar terdiri dari beberapa menu yang berfungsi untuk berinteraksi dengan pembuatan program aplikasi.

b. ToolBar

   ToolBar merupakan alternatif dari *MenuBar* yang ditampilkan dalam bentuk ikon.

c. ToolBox

   ToolBox adalah jendela yang menampung objek-objek kontrol yang digunakan pada saat mendesain *form*.

d. Form

   Form merupakan sebuah objek kontainer dari objek-objek lain yang dimiliki oleh Visual Basic, bagian dari sebuah *project*, dan digunakan sebagai alat berinteraksi antara program dengan pemakai (*user*).

e. Jendela Properties

Jendela *properties* adalah jendela yang memuat jenis-jenis properti yang dimiliki oleh suatu objek.

f. Jendela Project Explorer

Jendela *project explorer* adalah jendela yang menampilkan nama *project* dan komponen-komponen yang disertakan dalam sebuah *project*.

2.8.2 Objek Kontrol



Gambar 2.21. Objek Kontrol Visual Basic

Objek kontrol pada Visual Basic terletak pada jendela *ToolBox*, secara *default* isi dari *ToolBox* seperti terlihat pada Gambar 2.21, *ToolBox* ini menyediakan macam-macam objek yang sering digunakan seperti *Label*, *TextBox*, *Frame*, *ComboBox*, dan lain-lain. Untuk menampilkan objek kontrol lain dapat ditambahkan dengan mengklik menu *Project* kemudian pilih *Component*, atau dengan *shortkey* Ctrl+T.[11]

2.8.3 Tipe Data

Umumnya data dibagi menjadi dua bagian utama, yaitu karakter dan numerik, walaupun pada perkembangan selanjutnya dikembangkan menjadi beberapa tipe. Tipe data utama Visual Basic seperti terlihat pada Tabel 2.14.[11] (Sumber: Novian, 2004, hlm.39)

Tabel 2.14. Tipe Data Visual Basic

Tipe Data	Nilai Penyimpanan	Jangkauan
Integer	2 Byte	-32.768 s/d 32.767
Long	4 Byte	-2.1E9 s/d 2.1E9
Single	4 Byte	Untuk nilai negatif: -3,402823E s/d -1,401298E-45 Untuk nilai positif: 1,401298E-45 s/d 3,402823E38
Double	8 Byte	Untuk nilai negative: -1,79769313486232E308 s/d - 4,9406564581247E-324 Untuk nilai positif: 4,9406564581247E-324 s/d1,79769313486232E308 s/d
Decimal	8 Byte	Untuk bilangan tanpa koma: +/-79.228.162.514.337.593.543.950.335 Untuk bilangan decimal: +/-79,2281625143375935439503 35 Bilangn terkecil yang mungkin: 0,0000000000000000000000000001
Currency	8 Byte	-922.337.203.685.477,5808 s/d 922.337.203.685.477,5808
String	1 Byte / Char	0 s/d 2E32 karakter
Byte	1 Byte	0 s/d 255
Boolean	2 Byte	True atau False
Date	8 Byte	1 Jan 100 s/d 31 Dec 9999
Object	4 Byte	Referensi object
Variant	16 Byte + 1Byte/Karakter	Null Error Nilai numeric s/d jangkauan tipe data Double. String, Object, atau array.

Tipe data pada Visual Basic selain yang tertera pada Tabel 2.16, dapat juga dibuat tipe data gabungan dari berbagai tipe data. Pendeklarasian suatu variabel dengan menggunakan tipe data yang tepat, akan membuat program mengeksekusi lebih efisien dan penghematan dalam penggunaan memori.

2.8.4 Struktur Program

Setiap bahasa pemrograman tidak lepas dari struktur program, seperti struktur percabangan, struktur pengulangan. Pada Visual Basic struktur program dijelaskan sebagai berikut:[11]

a. Struktur percabangan.

Suatu program akan melakukan lumpatan ke baris program lainnya dengan membandingkan suatu kondisi dengan kondisi lainnya, atau program akan menjalan baris program berikutnya jika kondisi terpenuhi. Terdapat beberapa struktur percabangan diantaranya yaitu:

- If…Then.

  Struktur ini digunakan untuk menjalankan satu atau beberapa perintah bila suatu kondisi terpenuhi. Bentuk umum penulisan If…Then yaitu:

  ```
  If  kondisi Then [perintah]
  ```

- If…Then…Else

  Struktur If…Then…Else adalah pengembangan dari If…Then. Hanya saja, di sini ada banya pilihan kondisi untuk dikerjakan. Bila kondisi pertama bernilai benar, maka kerjakan perintah-perintah pertama. Bila kondisi pertama bernilai salah, maka periksa kondisi kedua dan seterusnya. Bentuk umum penulisan struktur ini adalah:

  ```
  IF  kondisi1 Then
  [perintah]
  ElseIf kondisi2 Then
  [perintah]
  Else
  [perintah]
  End IF
  ```

- Select Case.

  Struktur ini merupakan alternatif pengganti dari struktur If…Then…Else. Select Case mempunyai pembacaan yang lebih mudah sehingga penulisan dapat menjadi lebih efisien. Bentuk umum dari penulisan Select Case adalah:

  ```
  Select Case kondisi
  Case|Case Is = ekspresi1
  [perintah]
  Case|Case Is=ekspresi2
  [perintah]
  Case Else
  [Perintah]
  End Case
  ```

b. Struktur pengulangan.

Struktur pengulangan akan mengeksekusi perintah selama dan atau sampai kondisi terpenehui. Beberapa struktur pengulangan yaitu:

- Do While…Loop

  While akan memaksa program untuk melakukan pengulangan selama suatu kondisi masih bernilai benar. Struktur pengulangan Do While…Loop memiliki dua bentuk penulisan yaitu:

  ```
  Do While kondisi
  [perintah]
  Loop
  ```

  atau dengan menggunakan bentuk ke dua:

  ```
  Do
  [perintah]
  Loop While kondisi
  ```

  Pengulangan Do While…Loop mempunyai sebuah *statement* untuk menghentikan pengulangan walaupun kondisi masih bernilai benar. *Statement* tersebut adalah Exit Do.

- Do Until…Loop

  Until akan memaksa program untuk melakukan pengulangan sampai suatu kondisi bernilai benar. Seperti halnya Do While…Loop, Do Until…Loop pun memiliki dua bentuk yaitu:

  ```
  Do Until kondisi
  [perintah]
  Loop
  ```

  atau dengan menggunakan bentuk kedua:

  ```
  Do
  [perintah]
  Loop Until kondisi
  ```

- While…Wend

  Pengulangan While…Wend sebanarnya bentuk lain Do While…Loop. Struktur pengulangan akan mengeksekusi program selama suatu kondisi bernilai benar. Bentuk umum dari struktur While…Wend adalah:

  ```
  While kondisi
  [perintah]
  Wend
  ```

- For…Next

  For…Next digunakan jika telah diketahui banyaknya pengulangan yang akan dilakukan. Sebelum menggunakan For..Next nilai awal yang berfungsi sebagai indeks (*counter*) harus dideklarasikan terlebih dahulu. Bentuk umum pengulangan For…Next adalah sebagai berikut:

  ```
  For counter=awal To akhir [Step tahpa]
  [perintah]
  Next counter
  ```

- For Each…In… Next

  For Each…In…Next digunakan untuk pengulangan yang jumlah pengulangannya sesuai dengan jumlah elemen suatu koleksi objek yang biasanya tidak diketahui dengan pasti berapa jumlahnya. Bentuk umum penulisannya adalah sebagai berikut:

  ```
  For Each elemen In group
  [perintah]
  Next [elemen]
  ```

2.9 Bahasa Pemrograman Java

2.9.1 Perlengkapan Pemrogaman

Untuk pembuatan program aplikasi pada Android yang menggunakan bahasa pemrograman Java, maka diperlukan perangkat lunak yaitu sebagai berikut:

a. Java Development Kit (JDK)

   Yaitu program Java yang berfungsi sebagai kompilasi kode sumber Android.

b. Android SDK (*Software Development Kit*)

   Yaitu *tools* API (*Application Programming Interface*) yang diperlukan untuk memulai mengembangkan aplikasi pada *platform* Android menggunakan bahasa pemrograman Java.

c. IDE Eclipse

   Yaitu IDE yang digunakan untuk menulis program sumber dalam pembuatan aplikasi Android.

d. *Android Development Tools* (ADT)

Yaitu penghubung antara IDE Eclipse dengan Android SDK, atau disebut juga dengan *Plugins Eclipse*.

2.9.2 Activity

Activity merupakan *public class* dalam aplikasi Android. Setiap *activity* merupakan sesuatu yang unik atau *single,* yang ditunjukan untuk meng-*handle* macam-macam hal yang bisa dilakukan oleh *user*. Umumnya, *activity* berbuhubungan dengan *user* di mana *activity* menciptakan *Windows* atau UI yang mana ditampilkan dengan *concepts* setContentView(View). Ada 2 *method* yang pasti dimiliki oleh satu *activity* yaitu:

1. *onCreate* untuk menginisialisasi suatu *activity*, biasanya dipanggil dengan perintah setContentView(int) untuk *resource* yang didefinisikan di *layout* UI, dan perintah findViewById(int) untuk memanggil *widget* yang dibutuhkan UI untuk berinteraksi dengan aplikasi.
2. *onPause* untuk menyatakan ketika *user* meninggalkan suatu *activity*.

Untuk penggunaan dengan Context.startActivity(), semua kelas *activity* harus sesuai dengan <activity> yang dideklarasikan dalam suatu paket di AndroidManifest.xml. *Activity* adalah bagian penting dari keseluruhan siklus aplikasi, bagaiman *activity* dijalankan merupakan bagian penting dari model aplikasi.[11]

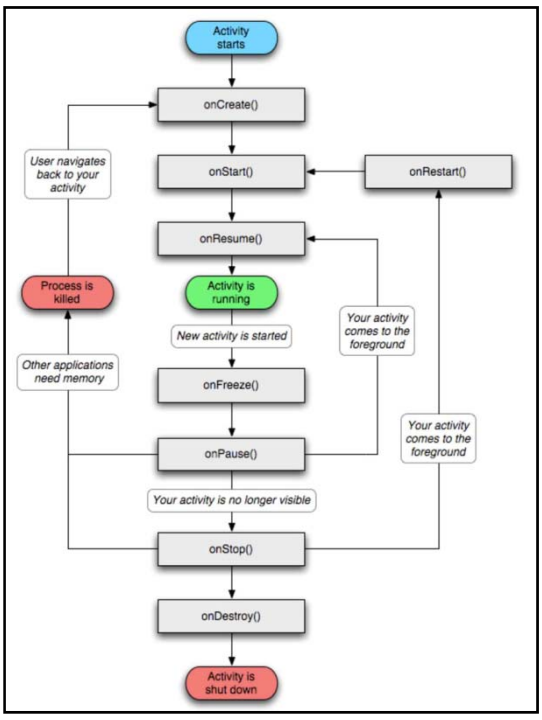
Activity aplikasi Android dikelola dengan sistem yang dikenal dengan *activity stack*. Ketika suatu *activity start*, *activity* diletakan pada *stack* yang paling atas dan *activity-activity* yang sudah berjalan berada dibawahnya dan akan terus berada pada posisi atas *stack* sampai muncul *activity* yang baru. Pada dasarnya *activity* memiliki empat keadaan yaitu:

1. *Active/running*, jika *activity* berada pada posisi atas *stack*.
2. *Pause*, jika *activity* tidak dipakai atau dibutuhkan pada suatu saat tertentu, tetapi *activity* itu masih ada atau *visible*, ketika *activity* baru yang ditangani oleh sistem *activity* yang lama disebut *pause* dan masih berada

di memori, bisa jadi suatu *activity* yang sudah keadaan *pause* tidak ada di memori yang kemungkinan disebabkan oleh kerbatasan memori.

3. *Stopped*, jika *activity* sudah tidak dipakai dan digantikan oleh *activity* lain, *activity* yang sudah *stopped* tidak akan pernah dipanggil lagi, dan secara permanen memori pun juga tidak menyimpan informasi mengenai *activity* ini.
4. *Restrart*, jika *activity pause* atau *stopped*, sistem dapat men-*drop activity* ini dari sistem memori, dan ketika *user* membutuhkan *activity* tersebut, *activity* akan kembali keadaan awal, artinya *activity* tersebut mengalami proses *restart*.

Pada Gambar 2.22 adalah ilustrasi mengenai *activity lifecycle* atau siklus dari *activity* di dalam sistem Android. (Sumber: Safaat H, 2011, hlm. 61)



Gambar 2.22. Lifecycle Activity

Berdasarkan Gambar 2.22 *lifecycel activity* di atas, secara umum dapat dijelaskan bahwa *looping* (perulangan) yang mungkin dialami oleh *activity* ada tiga perulangan yaitu:

1. *Entire lifetime* yaitu *activity* yang terjadi mulai dari onCreate() sampai dengan onDestroy(), biasanya *activity* ini akan dibuat *setup* global ketika mendefinisikannya.
2. *Visible lifetime* yaitu *activity* yang terjadi mulai dari onCreate() sampai dengan onStop().
3. *Foreground lifetime* yaitu *activity* yang terjadi diantara onResume() dan onPause().

2.9.3 User Interface

Secara umum arsitektur *User Interface* (UI) pada aplikasi Android adalah *user interface* yang meliputi *Activity* dan *user interface* yang terdiri dari komponen. Semua yang berhubungan dengan *user interface* pada aplikasi Android biasanya berada pada lokasi *res/layout/filename.xml*. Di mana *coding* Java memanggilnya yang dikenal dengan `R.layout.filename`.

2.9.4 AndroidManifest

File AndroidManifest.xml diperlukan oleh setiap aplikasi android, file ini mendeskripsikan variabel global dari paket aplikasi yang digunakan, berikut adalah contoh *listing* pada AndroidManifest.xml.

```
<?xml version=”1.0” encoding=”utf-8”?>
<manifest xmlns:android=http://schemas.android.com/apk/res/android
package=”com.tugas_akhir.afl”
android:versioncode=”1”
android:versionName=”1.0”>
<application android:icon=”@drawable/icon”
Android:label=”@string/app_name”>
<activity android:name”.afl” android:label=”@string/app_name”>
<intent-filter>
<action android:name=”android.intent.action.MAIN”/>
<category android:name=”android.intent.category.LAUNCHER”/>
</inent-filter>
</activity>
</application>
<uses-sdk android:minSdkVersion=”9”/>
</manifest>
```

Di dalam file AndroidManifest.xml ini berisi semua pendefinisian hal-hal yang dibutuhkan oleh aplikasi Android, beberapa elemen yang dapat dicantumkan di dalam file ini adalah sebagai berikut:

<Manifest>

Titik *root* utama dari AndroidManifest.xml, berisi atribut *package* aplikasi serta paket *activity* dalam program Android.

<uses-permision>

Menjelaskan tentang *user permission/security permission* yang harus diberikan agar aplikasi dapat berjalan sebagaimana mestinya, misalnya menggunakan *resources* yang tersedia dari sistem, seperti pengiriman SMS, dan sebagainya.

<permission>

Menjelaskan tentang *user permission/security permission*.

<instrumentation>

Mendeklarasikan komponen instrumen yang tersedia untuk menguji fungsionalitas dari paket aplikasi yang digunakan dalam aplikasi android.

<application>

Element *root* yang berisi deklarasi aplikasi android.

<intent-filter>

Mendeklarasikan *Intent* yang dibutuhkan oleh aplikasi Android yang digunakan, atribut-atribut bisa diberikan disini untuk men-*supply* label, ikon, data dan informasi yang digunakan dalam aplikasi android.

<action>

Berisi tentang *action type* yang didukung oleh komponen-komponen yang berada dalam aplikasi Android.

<category>

Mendeklarasikan kategori-kategori yang didukung oleh aplikasi Android.

<data>

Mendeklarasikan tipe MIME, URL, *authority* penggunaan URL serta penentuan *path* yang digunakan dalam URL.

<meta-data>
Mendeklarasikan meta data yang dibutuhkan sebagai tambahan data yang ada untuk digunakan pada aplikasi Android.

<receiver>
Mendeklarasikan dimana aplikasi diberikan informasi mengenai sesuai perubahan atau aksi yang terjadi, seperti menerima SMS.

<service>
Mendeklarasikan komponen yang dapat berjalan sebagai *service* (berjalan di *background*).

<provider>
Mendeklarasikan komponen-komponen yang mengelola data dan mem-publikasikannya untuk dikelola atau dipakai oleh aplikasi lain.

<uses-sdk uses-sdk android:minSdkVersion=?>
Mendeklarasikan SDK Android yang digunakan, bisa juga menentukan SDK minimum yang digunakan.

2.10 Perangkat Pendukung

2.10.1 Access Point

Perangkat keras *access point* merupakan perangkat yang digunakan transmisi data melalui *wireless*, di pasaran terdapat berbagai merk *access point* salah satunya yaitu dengan merk D-LINK. Meskipun merk *access point* berbeda-beda akan tetapi frekuensi yang dimiliki oleh *access point* mengacu pada standar IEEE (*Institute of Elelectrical and Electronics Enginers*) 802.11.

Access point yang akan digunakan yaitu D-LINK dengan model DWL-2100AP, *access point* ini mengacu pada standar IEEE 802.11g yaitu memiliki frekuensi 2.4GHz dengan kecepatan transfer mencapai 108Mbps. Pada Gambar 2.23 merupakan bentuk fisik *access point* D-LINK DWL-2100AP.



Gambar 2.23. Access Point D-LINK DWL-2100AP

2.10.2 Tablet PC

Tablet PC merupakan salah satu perangkat *mobile* yang biasanya digunakan untuk memudahkan pengguna terhubung dengan jaringan tanpa harus berada di tempat tertentu. Di pasaran terdapat berbagai merk tablet PC, pada tugas akhir ini akan digunakan tablet PC dengan merk Samsung GalaxyTab model GT-P6200. Pada Gambar 2.24 merupakan bentuk fisik tablet PC Samsung GalaxyTab GT-P6200.



Gambar 2.24. Tablet PC GalaxyTab Model GT-P6200

Adapun spesifikasi yang dimiliki oleh GalaxyTab GT-P6200 yaitu sebagai berikut:

a. Prosesor : 1.2GHz Dual Core
b. EDGE/GPRS : Quad Band (850/900/1800/1900 MHz)
c. Layar : 7” WSVGA (1024X600) PLS LCD
d. Wireless : 802.11 a/b/g/n
e. Wireless Bluetooth : Bluetooth 3.0
f. Antarmuka : USB 2.0
g. Sistem Operasi : Android 3.2
h. Baterai : 4000 mAh