BAB II
TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Penelitian serupa yang pernah dilakukan oleh Ari Beni Santoso dalam jurnal FEMA dengan judul Pembuatan Otomasi Pengaturan Kereta Api, Pengereman, dan Palang Pintu Pada Rel Kereta Api Mainan Berbasis Mikrokontroler. Alat pengaturan kereta api, pengereman dan palang pintu pada rel kereta api dibuat untuk mempermudah manusia dalm menjalankan tugas. Tanpa pemanfaatan sistem kontrol panel maka kemajuan teknologi akan sulit berlangsung. Maka merupakan kewajiban untuk meneliti pemanfaatan sistem kontrol panel dengan baik sehingga daya gunanya lebih efektif dan efesien. Untuk penerapan teknologi transportasi di Indonesia diperlukan suatu teknologi yang murah dan sederhana sehingga dapat digunakan dengan harga yang terjangkau.

Salah satu teknologi yang dapat diterapkan yaitu pengontrolan secara otomatis kereta api ketika berjalan dalam rel, supaya dapat mempermudah manusia dalam menjalankan tugas. Alat ini berfungsi sebagai sistem kontrol panel dalam pengontrolan secara otomatis kereta api ketika berjalan dalam rel. Proses dari alat ini bahwa apabila kereta mendeteksi adanya benda ataupun mahluk hisup pada jarak 10 cm maka akan memberikan perintah untuk mematikan dinamo kereta sehingga kereta akan otomatis berhenti dan mati dengan sendirinya, tetapi apabila kereta tidak mendeteksi adanya benda ataupun mahluk hidup yang berada di atas rel kereta pada jarak 10 cm maka memberikan perintah untuk tetap menghidupkan dinamo kereta sehingga kereta akan tetap berjalan. Dari hasil penelitian yang telah dilakukan maka didapat kesimpulan bahwa pembuatan otomasi pengaturan kerta api, pengereman, dan palang pintu pada rel kereta api mainan berbasis mikrokontroler, jarak yang paling sesuai dimana volt pada saat terhalang tidak terlalu tinggi ataupun rendah sehingga sinyal yang dihasilkan akan baik dan tegangan negatif yang dihasilkan tidak besar, pengaturan mekanisme kerja kereta api mainan dengan sistem kontrol otomatis dapat mempermudah pengoperasian.

Penelitian kedua serupa yang pernah dilakukan oleh Dedy Cahyadi dalam jurnal informatika mulawarman dengan judul Desain Sistem Absensi Berbasis Teknologi RFID. Rancangan sistem absensi menggunakan RFID dapat diimplementasikan sebagai pengganti sistem absensi PNS manual ke digital dengan keunggulan dapat digunakan semua PNS normal maupun cacat (sementara/tetap) anggota badan yang dijadikan ID dalam sistem biometrik. Alat ini merupakan sistem absensi PNS berbasis teknologi RFID. Tujuannya sebagai pengganti sistem absensi PNS manual ke digital dapat digunakan semua PNS normal maupun cacat (sementara/tetap) anggota badan yang dijadikan ID dalam sistem biometrik. Pada tahap awal sistem absensi digital berbasis RFID Card pada instasinya sendiri sebagai sample project yang kemudian akan disosialisasikan ke berbagai badan atau dinas. Sistem yang dibuat masih menggunakan lokal database dalam LAN perkantoran tersebut, sedangkan data kepegawaian di ambil dari database kepegawaian. Penerapan digitalisasi absensi sekaligus menghilangkan kekhawatiran bahwa absesnsi digital tidak diakui legalitasnya dalam sistem PNS Indonesia.

Penelitian ketiga serupa yang pernah dilakukan oleh Marlin Marlluka di dalam jurnal TESLA dengan judul Model Sistem Otomatisasi Pengisian Ulang Air Minum. Dalam hal ini mesin filterisasi air yang dapat mengubah air kurang bermutu menjadi air yang layak dikonsumsi secara langsung tanpa harus memasaknya terlebih dahulu. Salah satu cara untuk memproses air yang kurang bermutu menjadi air yang layak dikonsumsi adalah memasak air sampai suhu mendidih dan dipertahankan untuk jangka waktu tertentu sebelum siap dikonsumsi. Untuk dikonsumsi sebagai air minum, pada umumnya harus didinginkan kemabli melalui proses pendinginan yang memakan waktu. Hal ini tidak efesien dan merepotkan. Inilah yang menyebabkan kegemaran masyarakat akan air isi ulang yang dapat dikonsumsi langsung sebagai air minum. Dengan mesin filterisasi ini masyarakat hanya perlu membeli wadah galon yang bisa diisi ulang pada depot-depot pengisian ulang air minum. Selanjutnya dengan alat ini konsumen yang ingin membeli air sesuai dengan kebutuhan dan jumlah uang yang dimiliki, dapat memaksimalkan sistem kerja pengisisan ulang air minum, baik dari

petugas pengisi air minum maupun dari kepastian volume air yang diisi sesuai dengan biaya yang dikeluarkan. Mesin pengisian ulang air minum ini dirancang untuk mengubah air kurang bermutu menjadi air yang layak dikonsumsi secara langsung tanpa harus memasaknya terlebih dahulu.

Pada pengujian alat ini langsung menjalankan sistem secara keseluruhan. Modul pemancar dihubungkan dengan PC melalui RS-232 serta printer dihubungkan ke PC. Maka akan ada tampilan form utama terdiri dari 4 kotak yang berisi form stting, form new account, form deposit dan form beli. Setelah mengisi data, data yang akan dimasukkan oleh konsumen untuk melakukan pembelian kemudian dipancarkan oleh transmitter ke receiver dan diolah oleh mikrokontroler. Pada LCD akan muncul tampilan “Place bottle to the position”, dimana konsumen harus meletakkan wadah yang akan digunakan untuk pengisian sesuai dengan tempat yang telah tersedia. Lalu tekan tombol hijau, maka push button akan memberi input ke mikrokontroler akan masuk sebagai input relay. Output relay akan menjadi input pompa, dimana pompa akan aktif. Sistem ini pada dasarnya dapat bekerja dan berfungsi secara otomatis dengan baik. Sensor air yang terdapat pada selang pengisisan dapat mendeteksi kepenuhan ait dan menghentikan pengisisan bilamana konsumen melakukan pembelian dengan jumlah liter dan kapasitas wadah yang digunakan tidak sama. Selama proses hasil pengujian yang dilakukan maka dapat ditarik kesimpulan sebagai berikut bahwa sistem ini pada dasarnya dapat bekerja dan berfungsi secara otomatis dengan baik. Jumlah nominal liter air terkecil untuk pengisian pada sistem ini adalah 1 liter, dibawah 1 liter sistem ini tidak dapat berfungsi dan jumlah maksimum pengisian adalah 15 liter. Keterbatasan kepekaan sensor photointerrupter menunjukkan penyimpangan hasil kalibrasi sistem ini 1 liter sampai dengan 15 liter terjadi penyimpangan sebesar 5% dari volume sesungguhnya.

Penelitian keempat serupa yang pernah dilakukan oleh Dan Lajanto di dalam jurnal Universitas Tanjungpura dengan judul Sistem Kendali Umpan Balik Pada Lampu Berbasis *Short Massage Service* (SMS). Pemanfaatan media *handphone* selain sebagai alat komunikasi telah mengalami banyak perkembangan, namun semua pemanfaatan *handphone* tersebut belum ada yang menggunakan umpan

balik. Alat sistem kendali umpan balik pada lampu berbasis SMS yang selain sebagai kendali lampu juga dapat mengetahui kondisi lampu tersebut dengan sensor cahaya LDR dan dirancang dengan memanfaatkan media inframerah, gelombang radio, internet dan saluran telepon. Sistem kendali umpan balik pada lampu ini bekerja dengan kode/perintah yang ditetapkan. Bila isi pesan sms yang dikirimkan ke HP server tidak sesuai dengan yang ditetapkan, maka akan dijalankan. Pengujian dilakukan dengan menjalankan perangkat lunak yang dibuat kemudian melakukan koneksi terhadap modul dan ponsel. Setelah semua terhubung selanjutnya mengaktifkan sistem dengan mengklik/menekan tombol aktif sms. Aktifitas pembacaan isi pesan dapat diketahui pada bagian keterangan. Jika ada sms yang masuk dan sesuai kode maka perintah akan dilaksanakan sesuai kondisi dan pesan tersebut akan disimpan pada sms log sebelum dihapus. Alat ini dapat membantu pemakai untuk mengendalikan dan memonitoring kondisi lampu listrik dari jarak jauh menggunakan HP yang dimiliki. Kesimpulan dari pengujian bahwa peralatan ini dapat digunakan untuk mengendalikan lampu dari jarak jauh, dapat membantu pemakai untuk mengendalikan lampu dari jarak jauh, dapat membantu pemakai untuk mengendalikan dan me*monitoring* kondisi lampu listrik dari jarak jauh menggunakan *handphone* yang dimiliki, adanya SMS balasan yang menyatakan beban (lampu) telah dikendalikan, ini membuktikan umpan balik bekerja.

2.2 Pengenalan Mikrokontroler

Mikrokontroler merupakan suatu keping IC dimana terdapat mikroprosessor dan memori program ROM (*Read Only Memory*) serta memori serba guna RAM (*Random Accses Memory*) bahkan ada beberapa jenis mikrokontroler yang memiliki fasilitas ADC, PLL, EEPROM dalam satu kemasan. Penggunaan mikrokontroler dalam bidang kontrol sangat luas dan popular. (Santoso, 2013)

Dengan kata lain, mikrokontroler adalah versi mini atau mikro dari sebuah komputer karena mikrokontroler sudah mengandung beberapa periferal yang langsung bisa dimanfaatkan, mislanya port paralel, port serial, komparator,

konversi digital ke analog (DAC), konversi analog ke digital dan sebagainya hanya menggunakan sistem minimum yang tidak rumit atau kompleks.

Secara teknis, hanya ada 2 macam mikrokontroler. Pembagian ini didasarkan pada kompleksitas instruksi-instruksi yang dapat diterapkan pada mikrokontroler tersebut. Pembagiannya yaitu RISC (*Reduce Instruction Set Computer*) yaitu instruksi yang dimiliki terbatas, tetapi memiliki fasilitas yang lebih banyak, contohnya MCS51 yaitu AT89S52. CISC (*Complex Instruction Set Computer*) yaitu instruksi bisa dikatakan lebih lengkap tetapi dengan fasilitas secukupnya, contohnya mikrokontroler AVR yaitu Atmega 16.

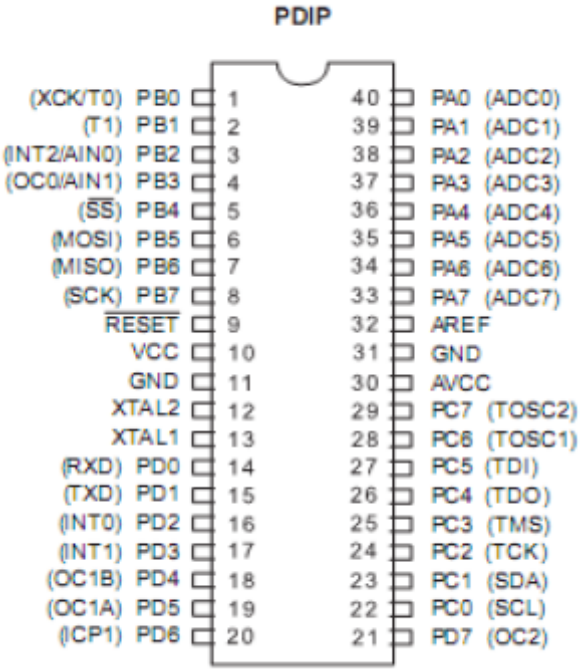
2.3 Mikrokontroler ATMega 16

ATMega16 merupakan mikrokontroler CMOS 8-bit buatan atmel keluarga AVR berbasis arsitektur RISC (*Reduced Instruction Set Computer*). Hampir semua instruksi dieksekusi dalam satu siklus *clock*. AVR mempunyai 32 register general-purpose, timer/counter fleksibel dengan mode compare, interrupt internal dan eksternal, serial UART, programmable *Watchdog Timer*, dan mode power saving, ADC dan PWM internal. AVR juga mempunyai In-system *Programmable Flash on-chip* yang mengijinkan memori program untuk diprogram ulang dalam sistem menggunakan hubungan serial SPI. ATMega16.

ATMega16 mempunyai troughput mendekati 1 MIPS per MHz membuat desainer sistem untuk mengoptimasi konsumsi daya versus kecepatan proses. (Didik, 2012)

2.3.1 Konfigurasi PIN Mikrokontroler ATMega 16

Pin-pin pada ATMega16 dengan kemasan 40-pin DIP (*dual in-line package*) ditunjukkan oleh gambar berikut. Guna memaksimalkan performa, AVR menggunakan arsitektur Harvard (dengan memori dan bus terpisah untuk program dan data)



Gambar 2.1 Pin-pin ATMega16 (Hadi, 2008)

2.3.2 Deskripsi Mikrokontroler Atmega16

ATMega16 mempunyai empat buah port yang bernama PortA, PortB, PortC, dan PortD. Keempat port tersebut merupakan jalur bi-directional dengan pilihan *internal pull-up*. Tiap port mempunyai tiga buah register bit, yaitu DDxn, PORTxn, dan PINxn. Huruf ‘x’ mewakili nama huruf dari port sedangkan huruf ‘n’ mewakili nomor bit. Bit DDxn terdapat pada I/O address DDRx, bit PORTxn terdapat pada I/O address PORTx, dan PINxn terdapata pada I/O *address* PINx.

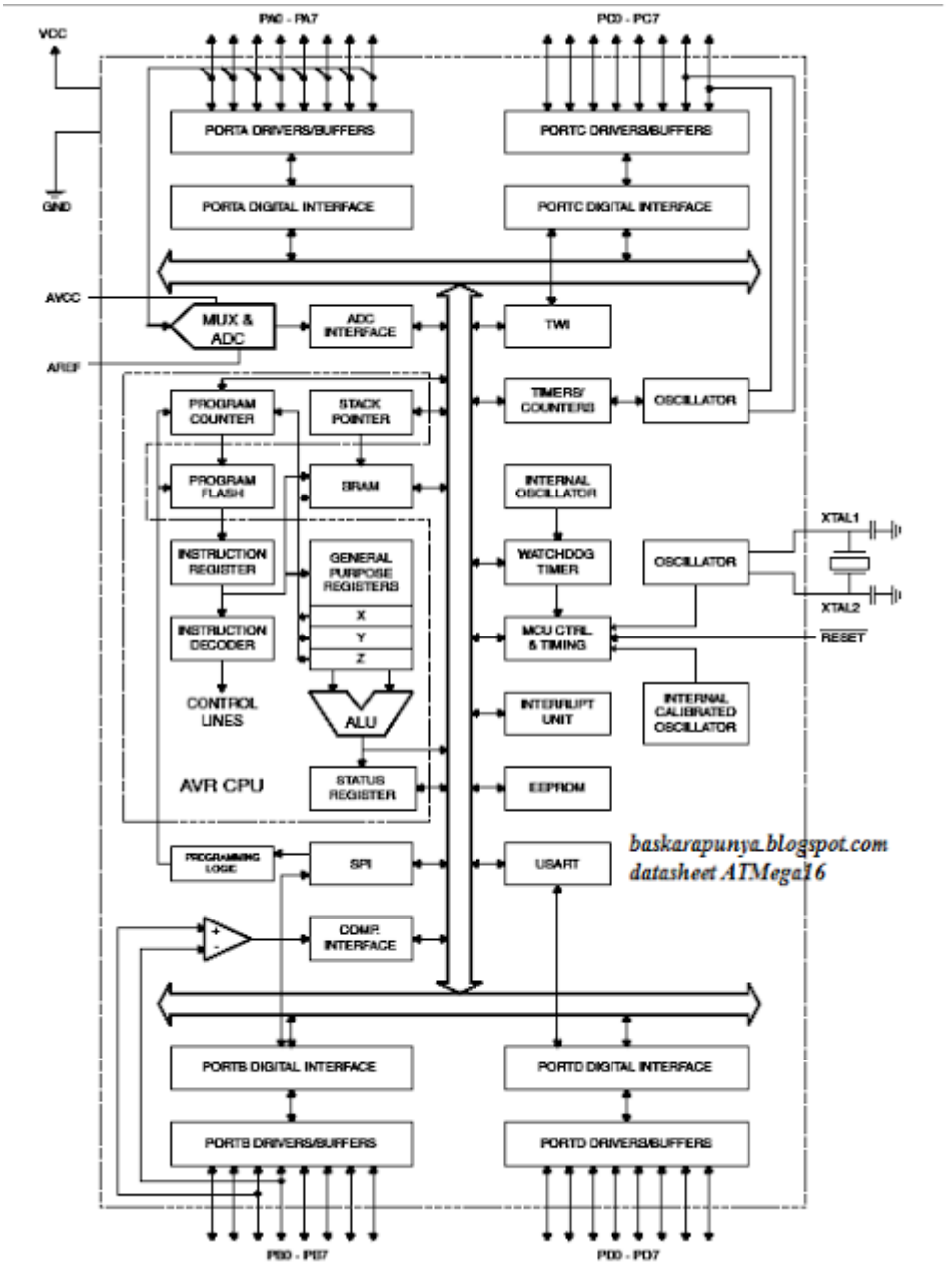
Bit DDxn dalam register DDRx (*Data Direction Refister*) menentukan arah pin. Bila DDxn diset 1 maka Px berfungsi sebagai pin output. Bila DDxn diset 0 maka Px berfungsi sebagai pin input, maka resistor pull-up diaktifkan. Untuk mematikan resistor pull-up, PORTxn harus diset 0 atau pin dikonfigurasi sebagai pin output. Pin port adalah tri-state setelah kondisi reset. Bila PORTxn diset 1 pada saat pin terkonfigurasi sebagai pin output maka pin terkonfigurasi sebagai pin output maka pin akan berlogika 0. Saat mengubah kondisi port dari kondisi tri-state (DDxn=0, PORTxn=0) ke kondisi output high (DDxn=1,

PORTxn=1) maka harus ada kondisi peralihan apakah itu kondisi pull-up enabled (DDxn=0, PORTxn=1) atau kondisi output low (DDxn=1, PORTxn=0).

Biasanya, kondisi pull-up enabled dapat diterima sepenuhnya, selama lingkungan impedansi tinggi tidak memperhatikan perbedaan antara sebuah strong high driver dengan sebuah pull-up. Jika ini bukan suatu masalah, maka bit PUD pada register SFIOR dapat diset 1 untuk mematikan semua pull-up dalam semua port. Peralihan dari kondisi input dengan pull-up ke kondisi output low juga menimbulkan masalah yang sama. Harus digunkan kondisi tri-state (DDxn=0, PORTxn=0) atau kondisi output high (DDxn=1, PORTxn=0) sebagai kondisi transisi. (Didik, 2012)

Tabel 2.1 Konfigurasi Pin Port

DDxn	PORTxn	PUD (in SFIOR)	I/O	Pull-up	Comment
0	0	X	Input	No	Tri-state (Hi-Z)
0	1	0	Input	Yes	Pxn will source current if ext. pulled low.
0	1	1	Input	No	Tri-state (Hi-Z)
1	0	X	Output	No	Output Low (Sink)
1	1	X	Output	No	Output High (Source)

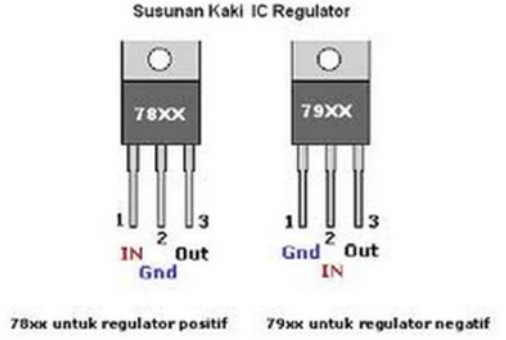


Gambar 2.2 Diagram Blok ATMega16

2.4 Regulator

Regulator adalah rangkaian regulasi atau pengatur tegangan keluaran dari sebuah catu daya agar efek dari naik turunnya tegangan jala-jala tidak mempengaruhi tegangan catu daya sehingga menjadi stabil.

Salah satu metode agar dapat menghasilkan tegangan output DC stabil adalah dengan menggunakan IC 78XX untuk tegangan positif dan IC 79XX untuk tegangan negatif dalam sistem Regulator Tegangan. Di bawah ini adalah besarnya tegangan output yang dapat dihasilkan IC regulator 78XX dan 79XX dimana XX adalah angka yang menunjukan besar tegangan output stabil.



Gambar 2.5 IC Regulator (Marlin, 2008)

2.7 Optocoupler

Optocoupler adalah suatu piranti yang terdiri dari 2 bagian yaitu transmitter dan receiver, yaitu antara bagian cahaya dengan bagian deteksi sumber cahaya dengan bagian deteksi sumber cahaya terpisah. Biasanya optocoupler digunkan sebagai saklar elektrik yang bekerja secara otomatis. Optocoupler *transmitter* (pengirim) dan *receiver* (penerima) yaitu sebagai berikut :

1. Transmiter

Merupakan bagian yg terhubung dengan rangkaian input atau rangkaian kontrol. Pada bagian ini terdapat sebuah LED infra merah (IR LED) yang berfungsi untuk mengirimkan sinyal kepada *receiver*. Pada *transmitter* dibangun dari sebuah LED infra merah. Jika dibandingkan dengan menggunakan LED biasa, LED infra merah memiliki ketahanan yang lebih baik terhadap sinyal tampak. Cahaya yang

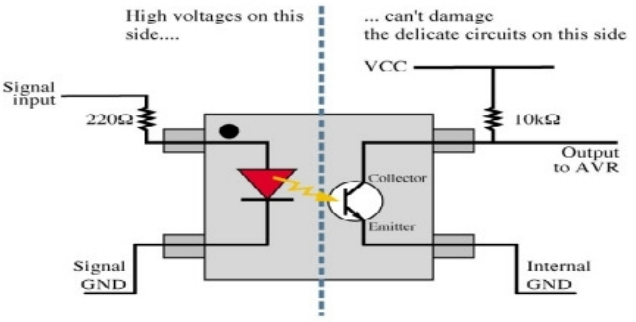
dipancarkan oleh LED infra merah tidak terlihat oleh mata telanjang.
2. Receiver

Merupakan bagian yg terhubung dengan rangkaian output atau rangkaian beban, dan berisi komponen penerima cahaya yang dipancarkan oleh transmitter. Komponen penerima cahaya ini dapat berupa photodioda atapun phototransistor. Pada bagian receiver dibangun dengan dasar komponen phototransistor. Phototransistor merupakan suatu transistor yang peka terhadap tenaga cahaya. Suatu sumber cahaya menghasilkan energi panas, begitu pula dengan spektrum infra merah. Karena spekrum infra mempunyai efek panas yang lebih besar dari cahaya tampak, maka phototransistor lebih peka untuk menangkap radiasi dari sinar infra merah. Jika dilihat dari penggunaannya, optocoupler biasa digunakan untuk mengisolasi common rangkaian input dengan common rangkaian output. Sehingga supply tegangan untuk masing-masing rangkaian tidak saling terbebani dan juga untuk mencegah kerusakan pada rangkaian kontrol (rangkaian input). (Santoso, 2013)

Optocoupler merupakan gabungan dari LED infra merah dengan fototransistor yang terbungkus menjadi satu *chips*. Cahaya infra merah termasuk dalam gelombang elektromagnetik yang tidak tampak oleh mata telanjang. Sinar ini tidak tampak oleh mata karena mempunyai panjang gelombang berkas cahaya yang terlalu panjang bagi tanggapan mata manusia. Sinar infra merah mempunyai daerah frekuensi 1 x 1012 Hz sampai dengan 1 x 1014 GHz atau daerah frekuensi dengan panjang gelombang 1µm – 1mm. LED infra merah ini merupakan komponen elektronika yang memancarkan cahaya infra merah dengan konsumsi daya sangat kecil. Jika diberi prasikap maju, LED infra merah yang terdapat pada optocoupler akan mengeluarkan panjang gelombang sekitar 0,9 mikrometer. Proses terjadinya pancaran cahaya pada LED infra merah dalam optocoupler adalah sebagai berikut. Saat dioda menghantarkan arus, elektron lepas dari ikatannya karena memerlukan tenaga dari catu daya listrik. Setelah elektron lepas, banyak elektron yang bergabung dengan lubang yang ada di sekitarnya (memasuki lubang lain yang kosong). Pada saat masuk lubang yang lain, elektron melepaskan tenaga yang akan diradiasikan dalam bentuk cahaya, sehingga dioda

akan menyala atau memancarkan cahaya pada saat dilewati arus. Cahaya infra merah yang terdapat pada optocoupler tidak perlu lensa untuk memfokuskan cahaya karena dalam satu chip mempunyai jarak yang dekat dengan penerimanya.

Pada optocoupler yang bertugas sebagai penerima cahaya infra merah adalah fototransistor. Fototransistor merupakan komponen elektronika yang berfungsi sebagai detektor cahaya infra merah. Detektor cahaya ini mengubah efek cahaya menjadi sinyal listrik, oleh sebab itu fototransistor termasuk dalam golongan detektor optik. Fototransistor memiliki sambungan kolektor–basis yang besar dengan cahaya infra merah, karena cahaya ini dapat membangkitkan pasangan lubang elektron. Dengan diberi prasikap maju, cahaya yang masuk akan menimbulkan arus pada kolektor. Fototransistor memiliki bahan utama yaitu germanium atau silikon yang sama dengan bahan pembuat transistor. Tipe fototransistor juga sama dengan transistor pada umumnya yaitu PNP dan NPN. Perbedaan transistor dengan fototransistor hanya terletak pada rumahnya yang memungkinkan cahaya infra merah mengaktifkan daerah basis, sedangkan transistor biasa ditempatkan pada rumah logam yang tertutup. Simbol optocoupler seperti terlihat pada gambar di bawah :

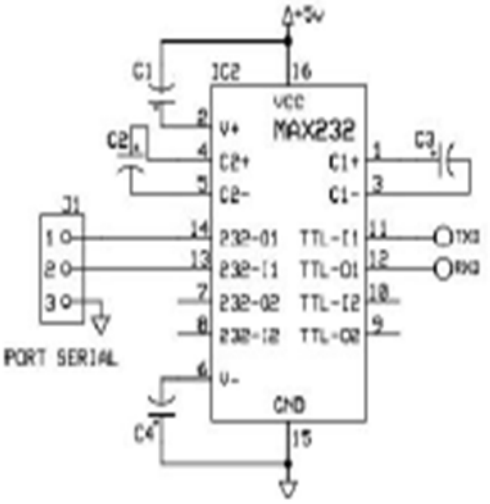


Gambar 2.6 Optocoupler (Santoso, 2013)

2.8 MAX 232

MAX232 merupakan salah satu jenis IC rangkaian antar muka dual RS-232 transmitter / receiver yang memenuhi semua spesifikasi standar EIA-232-E. IC MAX232 hanya membutuhkan power supply 5V (single power supply) sebagai catu. IC MAX232 di sini berfungsi untuk merubah level tegangan pada

COM1 menjadi level tegangan TTL / CMOS. IC MAX232 terdiri atas tiga bagian yaitu dual charge-pump voltage converter, driver RS232, dan receiver RS232.



Gambar 2.7 Rangkaian Elektronik IC MAX232 (Lajanto, 2012)

Dual Charge-Pump Voltage Converter.

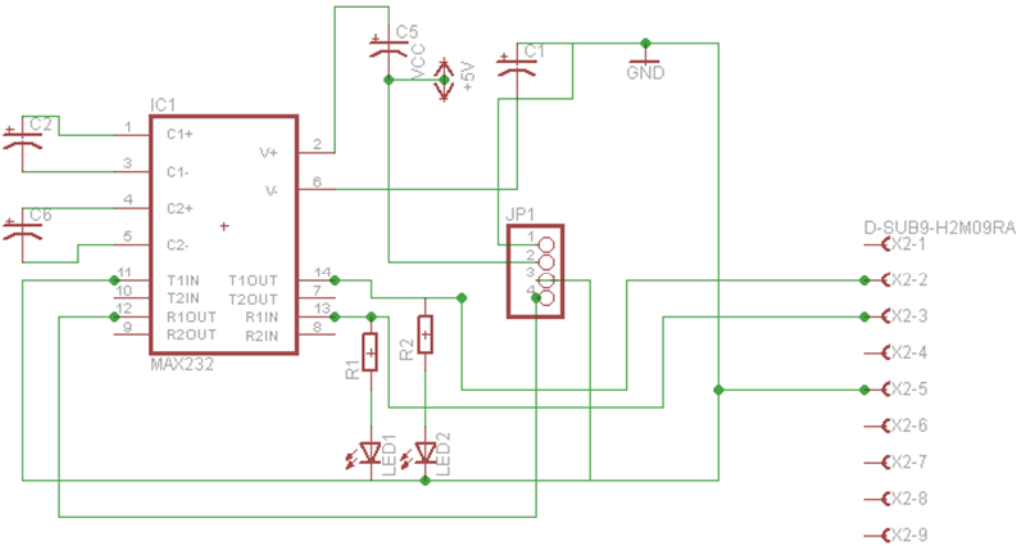
IC MAX232 memiliki dua charge-pump internal yang berfungsi untuk menkonversi tegangan +5V menjadi ±10V (tanpa beban) untuk operasi driver RS232. Konverter pertama menggunakan kapasitor C1 untuk menggandakan tegangan input +5V menjadi +10V saat C3 berada pada output V+. Konverter kedua menggunakan kapasitor C2 untuk merubah +10V menjadi -10V saat C4 berada pada output V-

- **Driver RS232**

Output ayunan tegangan (voltage swing) driver typical adalah ±8V. Nilai ini terjadi saat driver dibebani dengan beban nominal receiver RS232 sebesar 5kΩ atau Vcc = 5V. Input pada driver yang tidak digunakan bisa dibiarkan tidak terhubung kemana – mana. Hal ini dapat terjadi karena dalam kaki input driver IC MAX232 terdapat resistor pull-up sebesar 400kΩ yang terhubung keVcc. Resistor pull-up mengakibatkan output driver yang tidak terpakai menjadi low karena semua output driver diinversikan.

- **Receiver RS232**

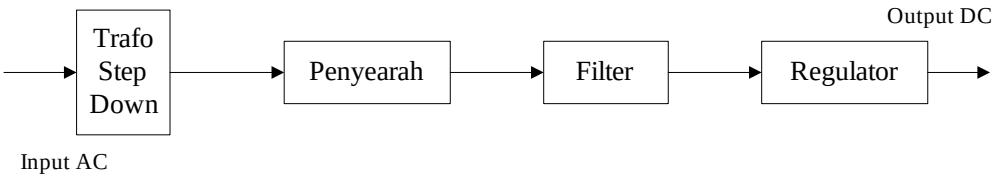
EIA mendefinisikan level tegangan lebih dari 3V sebagai logic 0, berdasarkan hal tersebut semua receiver diinversikan. Input receiver dapat menahan tegangan input sampai dengan ±25V dan menyiapkan resistor terminasi input dengan nilai nominal 5k. Nilai input receiver hysteresis typical adalah 0,5V dengan nilai minimum 0,2V, dan nilai delay propogasi typicalnya adalah 600ns.



Gambar 2.8 Rangkaian RS232

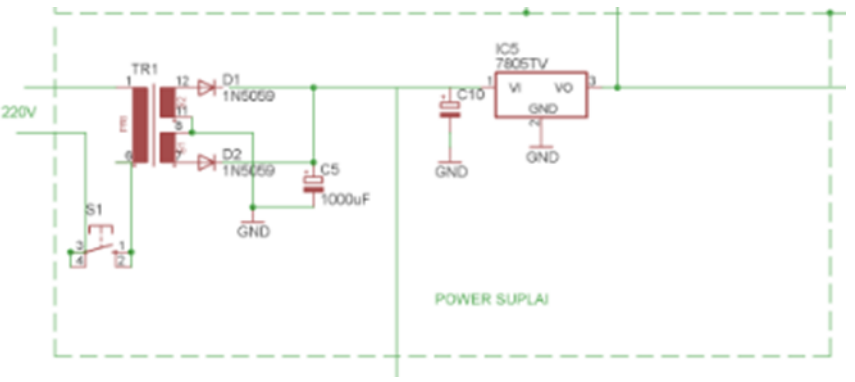
2.9 Power Supply

Power Supply merupakan piranti penyedia daya suatu rangkaian elektronis. Sumber energy listrikdari PLN merupakan arus listrik bolak-balik (AC), sedangkan perangkat keras dalam proyek akhir ini membutuhkan arus listrik searah (DC) yang nilainya konstan. Untuk itu diperlukan suatu tahapan proses yang secara umum.



Gambar 2.9 Diagram Kotak Power Supply

1. Trafo atau transformator merupakan komponen utama dalam membuat rangkaian catu daya yang berfungsi untuk mengubah tegangan listrik. Trafo dapat menaikkan dan menurunkan tegangan
2. *Step up* (penaik tegangan) apabila tegangan belitan scundair yang kita butuhkan lebih tinggi dari tegangan primair
3. *Step down* (penurun tegangan) apabila tegangan belitan scundair yang kita butuhkan lebih rendah dari tegangan primair
4. Dioda *Receiver* (Penyearah) untuk mengubah tegangan listrik AC yang berasal dari trafo *step down* atau trafo adaptor menjadi tegangan listrik arus searah DC
5. *Filter* (Penyaring) merupakan bagian yang terdiri dari kapasitor yang berfungsi sebagai penyaring atau meratakan tegangan listrik yang berasal dari *receiver*. Selain menggunakan filter juga menggunakan resistor sebagai tahanan.
6. Regulator berfungsi sebagai penstabil dan pengatur tegangan (regulator) yang berasal dari rangakaian penyaring



Gambar 2.10 Rangkaian Power Supply

2.10 Motor Servo

Motor servo adalah sebuah motor dengan sistem umpan balik tertutup di mana posisi dari motor akan diinformasikan kembali ke rangkaian kontrol yang ada di dalam motor servo. Motor servo merupakan salah satu jenis motor DC. Berbeda dengan motor stepper, motor servo beroperasi secara *close loop*. Poros motor dihubungkan dengan rangkaian kendali, sehingga jika putaran poros belum sampai pada posisi yang diperintahkan maka rangkaian kendali akan terus mengoreksi posisi hingga mencapai posisi yang diperintahkan. Motor servo banyak digunakan pada peranti R/C (remote control) seperti mobil, pesawat, helikopter, dan kapal, serta sebagai aktuator robot maupun penggerak pada kamera.

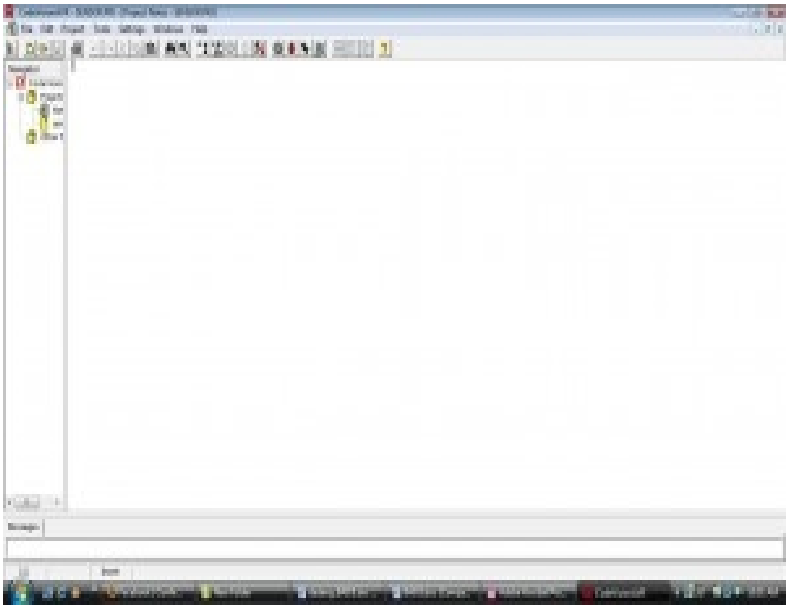
Gambar 2.11 Motor Servo (Djuandi, 2006)

Seperti namanya, servomotor adalah sebuah servo. Lebih khusus lagi adalah servo loop tertutup yang menggunakan umpan balik posisi untuk mengontrol gerakan dan posisi akhir. Masukan kontrolnya adalah beberapa sinyal, baik analog atau digital, yang mewakili posisi yang diperintahkan untuk poros output. Motor dipasangkan dengan beberapa jenis encoder untuk memberikan

posisi dan kecepatan umpan balik. Dalam kasus yang paling sederhana, hanya posisi yang diukur. Posisi diukur dari output dibandingkan dengan posisi perintah, input eksternal ke controller. Jika posisi keluaran berbeda dari yang diperlukan, sinyal error yang dihasilkan yang kemudian menyebabkan motor berputar pada kedua arah, yang diperlukan untuk membawa poros output ke posisi yang sesuai. Sebagai pendekatan posisi, sinyal error tereduksi menjadi nol dan motor berhenti. (Dujandi, 2006)

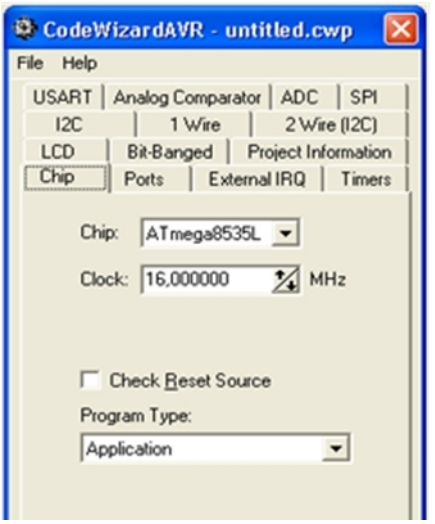
2.11 Codevision AVR

Codevision AVR merupakan salah satu s*oftware* kompiler yang khisis digunakan untuk mikrokontroler keluarga. Meskipun Codevision AVR termasuk software komersial, namun tetap dapat menggunakannya dengan mudah karena terdapat evaluasi yang tersedia secara gratis walaupun dengan kemampuan yang dibatasi (Saddam, 2013)



Gambar 2.12 Tampilan Code Vision AVR (Saddam, 2013)

Sealin itu, Codevision AVR juga menyediakan sebuah *tool* yang dinamakan dengan *Code Generator* atau CodeWizardAVR. CodeWizardAVR merupakan salah satu fasilitas yang disediakan oleh Codevision AVR yang dapat digunakan untuk mempercepat listing program. Secara otomatis CodeWizard dibuatkan kerangka program melalui menu-menu yang disediakan. Fasilitas ini sangat membantu apabila lupa dengan nama register yang akan digunakan untuk mengatur mode kerja fitur-fitur yang ada dalam mikrokontroler. Dengan kata lain, fasilitas ini digunakan untuk membantu mempercepat penulisan program serta mengingat kembali bagaimana penggunaan register-register apabila lupa. Selain itu, CodeVisionAVR juga menyediakan sebuah tool yang dinamakan dengan Code Generator atau CodeWizardAVR. Secara praktis, ini sangat bermanfaat membentuk sebuah kerangka program (*template*), dan juga memberi kemudahan bagi programmer dalam peng-inisialisasian register-register yang terdapat pada microcontroller AVR yang sedang diprogram. Dinamakan Code Generator, karena perangkat lunak CodeVision ini akan membangkitkan kode-kode program secara otomatis setelah fase inisialisasi pada jendela CodeWizardAVR selesai dilakukan. Secara teknis, penggunaan tool ini pada dasarnya hampir sama dengan application wizard pada bahasa-bahasa pemrograman Visual untuk komputer (seperti Visual C, Borland Delphi, dan sebagainya).



Gambar 2.13 Tampilan CodeWizard AVR

2.12 UML (*Unified Modelling Language*)

2.12.1 Pengertian *Unified Modelling Language* (UML)

Unified Modelling Language adalah suatu alat untuk memvisualisasikan dan mendokumentasikan hasil analisa dan desain yang berisi sintak dalam memodelkan sistem secara visual (Braun, *et.al:* 2001 dalam Haviluddin, *Memahami Penggunaan UML* (*Unified Modelling Language*): 2011). UML juga merupakan satu kumpulan konvensi pemodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem *software* yang terkait dengan objek (Braun, *et.al:* 2001 dalam Haviluddin, *Memahami Penggunaan UML* (*Unified Modelling Language*): 2011).

Gambar 2.14 Logo *Unified Modelling Language* (UML)

2.12.2 Sejarah Singkat *Unified Modelling Language* (UML)

UML dimulai secara resmi pada Oktober 1994 ketika *Rumbaugh* bergabung dengan *Booch* pada *Relation Software Corporation*. Proyek ini memfokuskan pada penyatuan metode *Booch* dan *OMT*. UML versi 0.8 merupakan metode penyatuan yang dirilis pada bulan Oktober 1995. Disini beberapa *partner* yang berkontribusi pada UML 1.0, diantaranya *Digital Equipment Corporation, Hewlett-Packard, I-Logix, Intellicorp, IBM, ICON Computing, MCI Systemhouse, Microsoft, Oracle, Relation, Texas Instruments* dan *Unisys*, dari kolaborasi ini dihasilkan UML 1.0 yang merupakan bahasa pemodelan yang ditetapkan secara baik, *expressive*, kuat dan cocok untuk lingkungan masalah yang luas. UML 1.0 ditawarkan menjadi standarisasi dari *Object Management Group (OMG)*, dan pada Januari 1997 dijadikan sebagai standar bahasa pemodelan. Pada bulan Januari 1997 ini lahirlah UML versi 1.0. Pada bulan September 1997 UML versi 1.1, dengan 8 buah diagram, yaitu:

1. Use Case diagram
2. Activity diagram
3. Sequence diagram
4. Collaboration diagram
5. Class diagram
6. Statechart diagram
7. Component diagram
8. Deployment diagram

Pada tahun 2002 lahirlah UML versi 2.0, menjadi 13 buah diagram, dengan penambahan dan penggantian yaitu :

1. Use Case Diagram
2. Activity Diagram
3. Sequence Diagram
4. Communication Diagram (Collaboration diagram in versi 1.x)
5. Class Diagram
6. State Machine Diagram (Statechart diagram in versi 1.x)
7. Component Diagram
8. Deployment Diagram
9. Composite Structure Diagram
10. Interaction Overview Diagram
11. Object Diagram
12. Package Diagram
13. Timing Diagram

2.12.3 Diagram-diagram pada UML (*Unified Modelling Language*)

Diagram-diagram yang terdapat didalam UML antara lain:

1. **Use Case Diagram**

Use case merupakan gambaran fungsionalitas dari suatu system, sehingga pengguna system mengerti kegunaan system yang akan dibangun. Use case diagram adalah penggambaran system dari sudut user, sehingga pembuatan use case lebih dititik beratkan pada fungsionalitas yang ada pada system, bukan berdasarkan alur kegiatan system (Haviluddin: 2011). Komponen-komponen yang terlibat dalam use case diagram :

a. Actor

Pada dasarnya *actor* bukanlah bagian dari *use case diagram*, namun untuk dapat terciptanya suatu *use case diagram* diperlukan beberapa *actor*. *Actor* tersebut mempresentasikan seseorang atau sesuatu (seperti perangkat, sistem lain) yang berinteraksi dengan sistem. Sebuah *actor* mungkin hanya memberikan informasi inputan pada sistem, hanya menerima informasi dari sistem atau keduanya menerima, dan memberi informasi pada sistem. *Actor* hanya berinteraksi dengan *use case*, tetapi tidak memiliki kontrol atas *use case*. *Actor* digambarkan dengan *stick man* . *Actor* dapat digambarkan secara secara umum atau spesifik, dimana untuk membedakannya kita dapat menggunakan *relationship* (Haviluddin: 2011).

b. Use Case

Use case adalah gambaran fungsionalitas dari suatu sistem, sehingga pengguna sistem paham dan mengerti mengenai kegunaan sistem yang akan dibangun (Haviluddin: 2011).

2. Activity Diagram

Pada dasarnya diagram Activity sering digunakan pada *flowchart*. Diagram ini berhubungan dengan diagram Statechart. Diagram Statechart berfokus pada obyek yang dalam suatu proses (atau proses menjadi suatu obyek), diagram Activity berfokus pada aktifitas-aktifitas yang terjadi yang terkait dalam suatu proses tunggal. Jadi dengan kata lain, diagram ini

menunjukkan bagaimana aktifitas-aktifitas tersebut bergantung satu sama lain.

Tabel 2.2 Komponen-komponen Activity Diagram:

No	Gambar	Nama	Keterangan
1		Activity	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2		Action	State dari sistem yang mencerminkan eksekusi dari suatu aksi
3		Start State	Bagaimana objek dibentuk atau diawali
4		End State	Bagaimana objek dibentuk dan dihancurkan
5		State Transition	State transition menunjukkan kegiatan apa berikutnya setelah suatu kegiatan
6		Fork	Percabangan yang menunjukkan aliran pada activity diagram
7		Join	Penggabungan yang menjadi arah aliran pada acivity diagram
8			Pilihan untuk mengambil keputusan
9		Flow Final	Aliran akhir

3. Sequence Diagram

Diagram sequence merupakan salah satu diagram *Interaction* yang menjelaskan bagaimana suatu operasi itu dilakukan. Diagram ini diatur berdasarkan waktu. Objek-objek yang berkaitan dengan proses berjalannya operasi diurutkan dari kiri ke kanan berdasarkan waktu terjadinya dalam pesan yang terurut.

4. Class Diagram

Class adalah sebuah spesifikasi yang akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class diagram* menggambarkan struktur statis dari kelas dalam sistem anda dan menggambarkan atribut, operasi dan hubungan antara kelas. Class menggambarkan keadaan (*atribut / property*) suatu sistem sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut. Class diagram menggambarkan struktur dan deskripsi class, package dan objek beserta hubungan satu sama lain sepeti *containment*, pewarisan, asosiasi, dan lain-lain. *Class diagram* membantu dalam memvisualisasikan struktur kelas-kelas dari suatu sistem dan merupakan tipe diagram yang paling banyak dipakai. Selama tahap desain, *class diagram* berperan dalam menangkap struktur dari semua kelas yang membentuk arsitektur sistem yang dibuat (Haviluddin: 2011). Class memiliki tiga area pokok yaitu:

a. Nama
b. Atribut
c. Metoda

5. Object Diagram atau Overview Diagram

Overview Diagram adalah pencangkokan secara bersama antara activity diagram dan sequence diagram. *Interaction Overview Diagram* dapat dianggap sebagai *activity diagram* dimana semua aktivitas diganti dengan sedikit *sequence diagram*, atau bisa juga dianggap sebagai *sequence diagram* yang dirincikan dengan notasi *activity diagram* yang digunakan untuk menunjukkan aliran pengawasan (Haviluddin: 2011).

Table 2.3 Komponen-komponen Overview Diagam

Simbol	Deskripsi
status awal 	status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal
aktivitas 	aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
percabangan / *decision* 	asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
penggabungan / *join*	asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
status akhir 	status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
swimlane 	memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi
fork, 	digunakan utk menunjukkan kegiatan yg dilakukan secara paralel
join, 	digunakan utk menunjukkan kegiatan yg digabungkan

2.12.4 Tujuan UML

Berikut ini tujuan utama dalam desain UML adalah (Sugrue J. 2009 dalam Haviluddin, *Memahami Penggunaan UML* (*Unified Modelling Language*). 2011):

1. Menyediakan bagi pengguna (analisis dan desain sistem) suatu bahasa pemodelan visual yang ekspresif sehingga mereka dapat mengembangkan dan melakukan pertukaran model data yang bermakna.
2. Menyediakan mekanisme yang spesialisasi untuk memperluas konsep inti.
3. Memberikan dasar formal untuk pemahaman bahasa pemodelan.

2.12.5 Cakupan UML

Adapun cakupan UML antara lain:

1. UML menggabungkan konsep *Bocch*, *OMT*, dan *OOSE*, sehingga UML merupakan suatu bahasa pemodelan tunggal yang umum dan digunakan secara luas oleh para *user* ketiga metode tersebut dan bahkan para *user* metode lainnya.
2. UML menekankan pada apa yang dapat dikerjakan dengan metode-metode tersebut.

3. UML berfokus pada suatu bahasa pemodelan standar, bahkan pada proses standar. Meskipun UML harus diaplikasikan dalam konteks sebuah proses dari pengalaman bahwa organisasi dan masalah yang berbeda juga memerlukan proses yang berbeda pula.

2.13 Notasi pada UML

Terdapat beberapa macam notasi pada UML, yaitu diantaranya:

1. ***Actor***

   Actor menggambarkan segala pengguna software aplikasi (*user*). Actor memberikan suatu gambaran jelas tentang apa yang harus dikerjakan software aplikasi. Sebagai contoh sebuah actor dapat memberikan input kedalam dan menerima informasi dari software aplikasi, perlu dicatat bahwa sebuah actor berinteraksi dengan use case, tetapi tidak memiliki kontrol atas use case. Sebuah actor mungkin seorang manusia, satu device, hardware atau sistem informasi lainnya (Haviluddin: 2011).

Gambar 2.15 Notasi *Actor*

2. **Use Case**

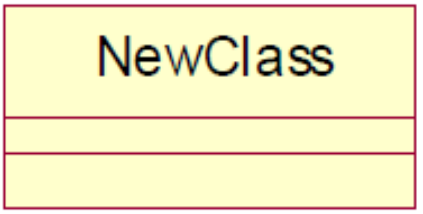
   Use case menjelaskan urutan kegiatan yang dilakukan *actor* dan sistem untuk mencapai suatu tujuan tertentu. Walaupun menjelaskan kegiatan, namun *use case* hanya menjelaskan apa yang dilakukan oleh *actor* dan sistem bukan bagaimana *actor* dan sistem melakukan kegiatan tersebut.

Gambar 2.16 Notasi *Use Case*

3. **Class**

   Class merupakan pembentuk utama dari sistem berorientasi objek, karena class menunjukkan kumpulan objek yang memiliki atribut dan operasi yang sama. Class digunakan untuk mengimplementasikan *interface*. Class

digunakan untuk mengabstraksikan elemen-elemen dari sistem yang sedang dibangun. Class bisa mempresentasikan baik perangkat lunak maupun perangkat keras, baik konsep maupun benda nyata.

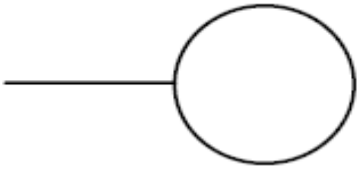


Gambar 2.17 Notasi Class

Notasi class berbentuk persegi panjang yang berisi 3 bagian. Persegi panjang paling atas untuk nama class, persegi panjang paling bawah untuk operasi dan persegi panjang ditengah untuk atribut. Atribut juga digunakan untuk menyimpan informasi. Nama atribut menggunakan kata benda yang bisa dengan jelas merepresentasikan informasi yang tersimpan didalamnya. Operasi menunjukkan sesuatu yang bisa dilakukan oleh objek dan menggunakan kata kerja.

4. Interface

Interface merupakan kumpulan operasi tanpa implementasi dari suatu class. Implementasi operasi dalam *interface* dijabarkan oleh operasi didalam class. Oleh karena itu keberadaan *interface* selalu disertai oleh class yang mengimplementasikan operasinya. *Interface* ini merupakan salah satu cara mewujudkan *prinsip enkapsulasi* dalam objek.



Gambar 2.18 Notasi *Interface*

5. Interaction

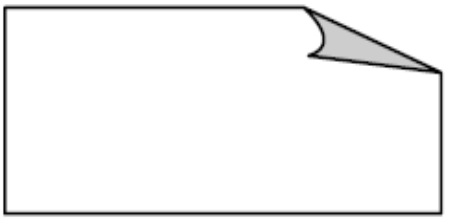
Interaction digunakan untuk menunjukkan baik aliran pesan atau informasi antar objek maupun hubungan antar objek. Biasanya *interaction* ini dilengkapi juga dengan teks bernama *operation signature* yang tersusun

dari nama operasi, parameter yang dikirim dan tipe parameter yang dikembalikan.

Gambar 2.19 Notasi *Interaction*

6. Note

Note digunakan untuk memberikan keterangan atau komentar tambahan dari suatu elemen sehingga bisa langsung terlampir dalam model. *Note* ini bisa disertakan ke semua elemen notasi yang lain.



Gambar 2.20 Notasi *Note*

7. Dependency

Dependency merupakan relasi yang menunjukkan bahwa perubahan pada salah stu elemen memberi pengaruh pada elemen lain. Elemen yang ada dibagian tanda panah adalah elemen yang tergantung pada elemen yang ada dibagian tanpa tanda panah. Terdapat 2 *stereotype* dan *dependency*, yaitu *include* dan *extend*. *Include* menunjukkan bahwa suatu bagian dari elemen (yang ada digaris tanpa panah) memicu eksekusi bagian dari elemen lain (yang ada digaris dengan panah). *Extend* menunjukkan bahwa sutau bagian dari elemen di garis tanpa panah bisa disisipkan kedalam elemen yang ada digaris dengan panah.

Gambar 2.21 Notasi *Dependency*

8. Association

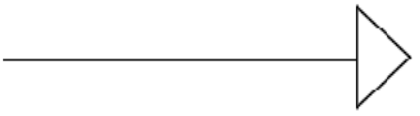
Association menggambarkan navigasi antar class (*navigation*), berapa banyak objek lain yang bisa berhubungan dengan satu objek (*multiplicity* antar class) dan apakah suatu class menjadi bagian dari class lainnya

(*aggregation*). *Navigation* dilambangkan dengan penambahan tanda panah di akhir garis. *Bidirectional navigation* menunjukkan bahwa dengan mengetahui salah satu class bisa didapatkan informasi dari class lainnya. Sementara *Unidirectional navigation* hanya dengan mengetahui class diujung garis *association* tanpa panah kita bisa mendapatkan informasi dari class di ujung dengan panah, tetapi tidak sebaliknya.

Gambar 2.22 Notasi *Association*

9. Generalization

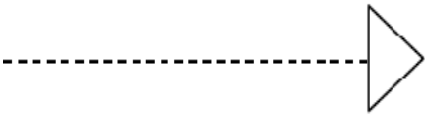
Generalization menunjukkan hubungan antara elemen yang lebih umum ke elemen yang lebih spesifik. Dengan *generalization,* class yang lebih spesifik (*subclass)* akan menurunkan atribut dan operasi dari class yang lebih umum (*superclass*) atau *"subclass is superclass"*. Dengan menggunakan notasi *generalization* ini, konsep *inheritance* dari prinsip hirarki dapat dimodelkan.



Gambar 2.23 Notasi Generalization

10. Realization

Realization menunjukkan hubungan bahwa elemen yang ada di bagian tanpa panah akan merealisasikan apa yang dinyatakan oleh elemen yang ada di bagian dengan panah. Misalnya class merealisasikan *package,* komponen merealisasikan class atau *interface.*



Gambar 2.24 Notasi *Realization*