BAB II

TINJAUAN PUSTAKA

II.1. Perancangan

Perancangan sistem adalah sebuah teknik pemecahan masalah yang saling melengkapi (dengan analisis sistem) yang merangkai kembali bagian-bagian komponen menjadi sistem yang lengkap harapannya, sebuah sistem yang diperbaiki. Hal ini melibatkan penambahan, penghapusan, dan perubahan-perubahan bagian relatif pada sistem awal (Hanif Al Fatta; 2008 : 44).

Desain atau perancangan dalam pengembangan perangkat lunak merupakan upaya untuk mengkontruksi sebuah sistem yang memberikan kepuasan (mungkin informal) akan spesifikasi kebutuhan fungsional memenuhi target, memenuhi kebutuhan secara implisit dan eksplisit dari segi performansi maupun penggunaan sumber daya, kepuasan batasan pada proses desain dari segi biaya, waktu, dan peranggkat. (Rosa A.S, M. Shalahuddin ; 2011: 20)

II.2. Simulasi

Simulasi ialah suatu metodologi untuk melaksanakan percobaan dengan menggunakan model dari satu sistem nyata (Siagian, 1987). Menurut Hasan (2002), simulasi merupakan suatu model pengambilan keputusan dengan mencontoh atau mempergunakan gambaran sebenarnya dari suatu sistem kehidupan dunia nyata tanpa harus mengalaminya pada keadaan yang sesungguhnya.

Simulasi adalah suatu teknik yang dapat digunakan untuk memformulasikan dan memecahkan model – model dari golongan yang luas. Golongan atau kelas ini sangat luasnya sehingga dapat dikatakan , " Jika semua cara yang lain gagal, cobalah simulasi" (Schroeder, 1997).

II.2.1. Kelebihan Simulasi

Meskipun model analitik sangat berguna dan sering digunakan, namun masih terdapat beberapa keterbatasan, yaitu :

1. Model analitik tidak mampu menelusuri perangai suatu sistem pada masa lalu dan masa mendatang melalui pembagian waktu. Model analitik hanya memberikan penyelesaian secara menyeluruh, suatu jawab yang mungkin tunggal dan optimal tetapi tidak menggambarkan suatu prosedur operasional untuk masa lebih singkat dari masa perencanaan. Misalnya, penyelesaian persoalan program linier dengan masa perencanaan satu tahun, tidak menggambarkan prosedur operasional untuk masa bulan demi bulan, minggu demi minggu, atau hari demi hari.

2. Model matematika yang konvensional sering tidak mampu menyajikan sistem nyata yang lebih besar dan rumit (kompleks). Sehingga sukar untuk membangun model analitik untuk sistem nyata yang demikian. Kalaupun model matematika mampu menyajikan sistem nyata yang kompleks demikian, tetapi bisa jadi tidak mungkin diselesaikan dengan hanya menggunakan teknik analitis yang sudah ada. Seperti sistem pedesaan yang dikaitkan dengan faktor ekonomi, sosial, politik, dan lain – lain.

3. Model analitik terbatas pemakaiannya dalam hal – hal yang tidak pasti dan aspek dinamis (faktor waktu) dari persoalan manajemen. Berdasarkan hal di atas, maka konsep simulasi dan penggunaan model simulasi merupakan solusi terhadap ketidakmampuan dari model analitik.

Beberapa alasan yang dapat menunjang kesimpulan di atas adalah sebagai berikut :

1. Simulasi dapat memberi solusi kalau model analitik gagal melakukannya.

2. Model simulasi lebih realistis terhadap sistem nyata karena memerlukan asumsi yang lebih sedikit. Misalnya, tenggang waktu dalam model persediaan tidak perlu harus deterministik.

3. Perubahan konfigurasi dan struktur dapat dilaksanakan lebih mudah untuk menjawab pertanyaan : *what happen if...* Misalnya, banyak aturan dapat dicoba untuk mengubah jumlah langganan dalam sistem antrian.

4. Dalam banyak hal, simulasi lebih murah dari percobaannya sendiri.

5. Simulasi dapat digunakan untuk maksud pendidikan.

6. Untuk sejumlah proses dimensi, simulasi memberikan penyelidikan yang langsung dan terperinci dalam periode waktu khusus.

II.2.2. Kekurangan Simulasi

1. Simulasi bukanlah presisi dan juga bukan suatu proses optimisasi. Simulasi tidak menghasilkan solusi, tetapi ia menghasilkan cara untuk menilai solusi termasuk solusi optimal.

2. Model simulasi yang baik dan efektif sangat mahal dan membutuhkan waktu yang lama dibandingkan dengan model analitik.

3. Tidak semua situasi dapat dinilai melalui simulasi kecuali situasi yang memuat ketidakpastian (Siagian, 1987).

II.2.3. Model – model Simulasi

Model – model simulasi yang ada dapat dikelompokkan ke dalam beberapapenggolongan, antara lain :

1. Model *Stochastic* atau *probabilistic*

Model stokastik adalah model yang menjelaskan kelakuan sistem secara probabilistik; informasi yang masuk adalah secara acak .Model ini kadang – kadang juga disebut sebagai model simulasi Monte Carlo. Di dalam proses *stochastic* sifat – sifat keluaran (*output*) merupakan hasil dari konsep random (acak). Meskipun output yang diperoleh dapat dinyatakan dengan rata – rata, namun kadang – kadang ditunjukkan pula pola penyimpangannya. Model yang mendasarkan pada teknik peluangdan memperhitungkan ketidakpastian (*uncertainty*) disebut model probabilistic atau model stokastik

2. Model Deterministik

Pada model ini tidak diperhatikan unsur random, sehingga pemecahan masalahnya menjadi lebih sederhana.

3. Model Dinamik

Model simulasi yang dinamik adalah model yang memperhatikan perubahan – perubahan nilai dari variabel – variabel yang ada kalau terjadi pada waktu yang berbeda.

4. Model Statik

Model statik adalah kebalikan dari model dinamik. Model statik tidak memperhatikan perubahan – perubahan nilai dari variabel – variabel yang ada kalau terjadi pada waktu yang berbeda.

5. Model Heuristik

Model heuristik adalah model yang dilakukan dengan cara coba – coba, kalau dilandasi suatu teori masih bersifat ringan, langkah perubahannya dilakukan berulang – ulang, dan pemilihan langkahnya bebas, sampai diperoleh hasil yang lebih baik, tetapi belum tentu optimal (Subagyo,2000).

II.2.4. Langkah – Langkah Dalam Proses Simulasi

Pada umumnya terdapat 5 langkah pokok yang diperlukan dalam menggunakan simulasi, yaitu :

1. Menentukan persoalan atau sistem yang hendak disimulasi.
2. Formulasikan model simulasi yang hendak digunakan.
3. Ujilah model dan bandingkan tingkah lakunya dengan tingkah laku dari sistem nyata, kemudian berlakukanlah model simulasi tersebut.
4. Rancang percobaan – percobaan simulasi.
5. Jalankan simulasi dan analisis data (Levin, dkk, 2002).

II.3. Ayunan

Ayunan adalah buaian (perkakas yg bergantung untuk menidurkan anak, terbuat dr rotan,kain panjang, dsb) ; [2] mainan untuk berayun-ayun: di taman itu disediakan berbagai ~ untuk anak- anak ; [3] ark bandul (buaian pada lonceng): ~;

[4] gerakan mengayun: elok benar ~ lenggangnya ; [5] pengumban (tali untuk melontarkan batu): diambilnya batu laludilontarkannya dengan ~; ~ sederhana ayunan tunggal; ~ tunggal benda yg digantungkan pd ujung seutas tali, dapat berayun di suatu kedudukan setimbang dl keadaan ideal. (Dendy Sugono ; 2008 :107)

II.4. Port Serial / RS-232

Input / output unit merupakan bagian dari komputer untuk menerima data maupun mengeluarkan / menampilkan data setelah diproses oleh prosessor. Port I/O adlah port atau gerbang atau tempat dipasangnya konektor dari peralatan I/O. Dimana setiap port I/O dibawah kontrol dari prosesor.

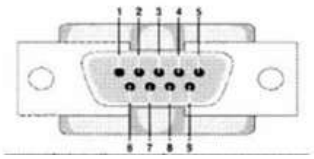
Pada prinsipnya, serial ialah pengiriman data dilakukan per bit, sehingga lebih lambat dibandingkan parallel seperti pada port printer yang mampu mengirim 8 bit sekaligus dalam sekali detak. Beberapa contoh serial ialah mouse, scanner dan system akuisisi data yang terhubung ke port COM1/COM2. Device pada serial port dibagi menjadi 2 (dua) kelompok yaitu *Data Communication Equipment* ***(DCE)*** dan *Data Terminal Equipment* ***(DTE).*** Contoh dari *DCE* ialah modem, plotter, scanner dan lain lain sedangkan contoh dari *DTE* ialah terminal di komputer. Spesifikasi elektronik dari serial port merujuk pada *Electronic Industry Association (EIA)* :

a. "Space" (logika 0) ialah tegangan antara + 3 hingga +25 V.
b. "Mark" (logika 1) ialah tegangan antara –3 hingga –25 V.
c. Daerah antara + 3V hingga –3V tidak didefinisikan /tidak terpakai

d. Tegangan open circuit tidak boleh melebihi 25 V.

e. Arus hubungan singkat tidak boleh melebihi 500mA.

Komunikasi serial membutuhkan port sebagai saluran data. Berikut tampilan port serial DB9 yang umum digunakan sebagai port serial digunakan sebagai port serial



Pin	Signal	Pin	Signal
1	*Data Carrier Detect* (DCD)	6	*Data Set Ready* (DSR)
2	*Received Data* (RxD)	7	*Request To Send* (RTS)
3	*Transmitted Data* (TxD)	8	*Clear To Send* (CTS)
4	*Data Terminal Ready* (DTR)	9	*Ring Indicator* (RI)
5	*Signal Ground* (*common*)		

Gambar II.1. Port DB9
Sumber : (http://elib.unikom.ac.id/files/disk1/535/jbptunikompp-gdl-indrapurna-26711-5-unikom_i-i.pdf.)

Keterangan

· Pin 1 = Data Carrier Detect (DCD)

· Pin 2 = Received Data (RxD)

· Pin 3 = Transmitted Data (TxD)

· Pin 4 = Data Terminal Ready (DTR)

· Pin 5 = Signal Ground (common)

· Pin 6 = Data Set Ready (DSR)

· Pin 7 = Request To Send (RTS)

· Pin 8 = Clear To Send (CTS)

· Pin 9 = Ring Indicator (RI)

Konektor port serial terdiri dari 2 jenis, yaitu konektor 25 pin (DB25 dan 9 pin (DB9) yang berpasangan (jantan dan betina). Bentuk dari konektor DB-25 sama persis dengan port paralel. Umumnya COM1 berada dialamat 3F8H, sedangkan COM2 dialamat 2F8H.

(a) (b)

Gambar II.2.a. DB9 male Gambar II.2.b. DB9 female (http://elib.unikom.ac.id/files/disk1/535/jbptunikompp-gdl-indrapurna-26711-5-unikom_i-i.pdf.)

Tabel 1. Jenis Sinyal RS232 yang Umum Digunakan

Nama Sinyal	Arah Sinyal	Nomor Kaki Konektor	
		DB9	DB25
Signal common	-	5	7
Transmitted Data (TD)	Ke DCE	3	2
Received Data (RD)	Dari DCE	2	3
Request To Send (RTS)	Ke DCE	7	4
Clear To Send (CTS)	Dari DCE	8	5
DCE Ready (DSR)	Dari DCE	6	6
DTE Ready (DTR)	Ke DCE	4	20
Ring Indicator (RI)	Dari DCE	9	22
Data Carrier Detect (DCD)	Dari DCE	1	6

Sumber : (http://elib.unikom.ac.id/files/disk1/535/jbptunikompp-gdl-indrapurna-26711-5-unikom_i-i.pdf.)

a. *Receive Line signal detect*, dengan saluran ini *DCE* memberitahukan ke *DTE* bahwa pada terminal masukkan ada data masuk.

b. *Receive Data*, digunakan *DTE* menerima data dari *DCE*.

c. *Transmit Data*, digunakan *DTE* mengirimkan data ke DCE.

d. *Data Terminal Ready*, pada saluran ini *DTE* memberitahukan kesiapan terminalnya.

e. *Signal Ground*, saluran ground.

f. *Ring Indicator*, pada saluran ini *DCE* memberitahukan ke *DTE* bahwa sebuah stasiun menghendaki berhubungan dengannya.

g. *Clear To Send*, dengan saluran ini *DCE* memberitahukan bahwa *DTE* boleh mulai mengirim data.

h. *Request To Send*, dengan saluran ini *DCE* diminta mengirim data oleh *DTE*.

i. *DCE Ready*, sinyal aktif pada saluran ini menunjukkan bahwa *DCE* sudah siap.

Ada 2 macam cara komunikasi data serial yaitu Sinkron dan Asinkron. Pada komunikasi data serial sinkron, clock dikirimkan bersama sama dengan data serial, tetapi clock tersebut dibangkitkan sendiri – sendiri baik pada sisi pengirim maupun penerima.Sedangkan pada komunikasi serial asinkron tidak diperlukan clock karena data dikirimkan dengan kecepatan tertentu yang sama baik pada pengirim / penerima.

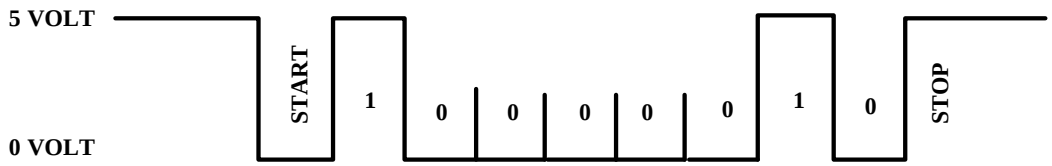
Pada IBM PC kompatibel port serialnya termasuk jenis asinkron. Komunikasi data serial ini dikerjakan oleh UART (*Universal Asynchronous Receiver Transmitter*). IC UART dibuat khusus untuk mengubah data parallel menjadi data serial dan menerima data serial yang kemudian dirubah lagi menjadi data parallel.IC UART 8250 merupakan salah satunya. Selain berbentuk IC mandiri berbagai macam mikrokontroller juga ada yang dilengkapi dengan UART, misalnya AT89S51/52/53 atau PIC16F877.

Pada UART, kecepatan pengiriman data (atau yang sering disebut dengan Baud Rate) dan fase clock pada sisi transmitter dan sisi receiver harus sinkron. Untuk itu diperlukan sinkronisasi antara Transmitter dan Receiver. Hal ini dilakukan

oleh bit “Start” dan bit “Stop”. Ketika saluran transmisi dalam keadaan idle, output UART adalah dalam keadaan logika “1”.

Ketika Transmitter ingin mengirimkan data, output UART akan diset dulu ke logika “0” untuk waktu satu bit. Sinyal ini pada receiver akan dikenali sebagai sinyal “Start” yang digunakan untuk menyinkronkan fase clocknya sehingga sinkron dengan fase clock transmitter. Selanjutnya data akan dikirimkan secara serial dari bit yang paling rendah (bit0) sampai bit tertinggi.Selanjutnya akan dikirimkan sinyal “Stop” sebagai akhir dari pengiriman data serial.

Sebagai contoh misalnya akan dikirimkan data huruf “A” dalam format ASCII (atau sama dengan 41 heksa atau 0100 0001.



Gambar II.3 . Pengiriman hufur “A” tanpa bit paritas (http://elib.unikom.ac.id/files/disk1/535/jbptunikompp-gdl-indrapurna-26711-5-unikom_i-i.pdf.)

Kecepatan transmisi (*baud rate*) dapat dipilih bebas dalam rentang tertentu. Baud rate yang umum dipakai adalah 110, 135, 150, 300, 600, 1200, 2400, dan 9600 (bit/perdertik).Dalam komunikasi data serial, baud rate dari kedua alat yang berhubungan harus diatur pada kecepatan yang sama. Selanjutnya harus ditentukan panjang data (6,7 atau 8 bit), paritas (genap, ganjil, atau tanpa paritas), dan jumlah bit “Stop” (1, 1 ½ , atau 2 bit).

Untuk dapat menggunakan port serial harus diketahui dahulu alamat dari port serial tersebut. Biasanya tersedia dua port serial pada CPU, yaitu COM1 dan COM2. Base Address COM1 biasanya 1016 (3F8h) dan COM2 biasanya 760 (2F8h). Alamat tersebut adalah alamat yang biasa digunakan, tergantung komputer yang digunakan.Tepatnya kita bisa melihat pada peta memori tempat menyimpan alamat tersebut, yaitu memori 0000.0400h untuk COM1 dan 0000.0402h untuk COM2. Berikut adalah nama – nama register yang digunakan beserta alamatnya.

a. *RX Buffer* , digunakan untuk menampung dan menyimpan data dari DCE.
b. *TX Buffer* , digunakan untuk menampung dan menyimpan data yang akan dikirim ke port serial.
c. *Baud Rate Divisor Latch LSB* , digunakan untuk menampung byte bobot rendah untuk pembagi clock pada IC UART agar didapat baud rate yang tepat.
d. *Baud Rate Divisor Latch MSB* , digunakan untuk menampung byte bobot tinggi untuk pembagi clock pada IC UART sehingga total angka pembagi adalah 4 byte yang dapat dipilih dari 0001h sampai FFFFh.

Port serial sering digunakan untuk interfacing komputer dan mikrokontroler, karena kemampuan jarak pengiriman data dibandingkan port paralel. Berikut contoh program assembly untuk komunikasi serial antara 2 PC. Untuk komunikasi ini, yang perlu dihubungkan :

a. Pin TxD ke pin RxD computer lain
b. Pin RXD dihubungkan ke pin TxD komputer lain

c. RTS dan CTS dihubung singkat
d. DSR dan DTR dihubung singkat
e. GND dihubungkan ke GND komputer lain

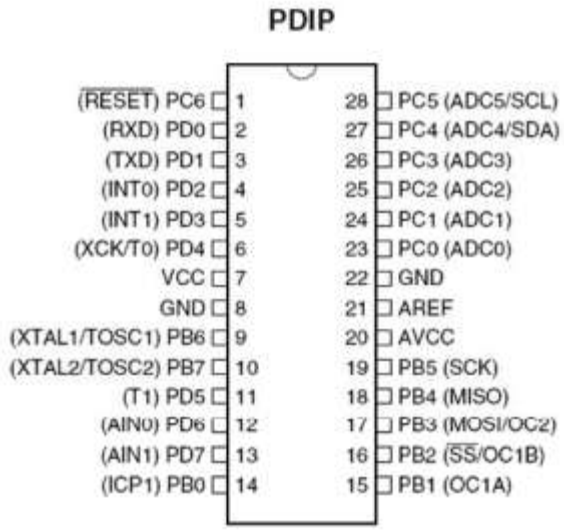
Bila dibandingkan cara komunikasi yang dilakukan secara paralel dengan cara komunikasi yang dilakukan secara serial, maka masing-masing akan memiliki keuntungan dan kelebihan yang tersendiri. Komunikasi yang dilakukan secara serial mempunyai keuntungan dari sisi pengkabelan, karena hanya memerlukan tiga buah kabel, TX, RX dan Ground. Dibandingkan dengan menggunakan port parallel penggunaan port serial terkesan lebih rumit. Berikut adalah keuntungan penggunaan port serial dibandingkan penggunaan port parallel.

a. Pada komunikasi dengan kabel yang panjang, masalah cable loss tidak akan menjadi masalah besar daripada menggunakan kabel parallel. Port serial mentransmisikan "1" pada level tegangan -3 Volt sampai -25 Volt dan "0" pada level tegangan +3 Volt sampai +25 Volt, sedangkan port parallel mentransmisikan "0" pada level tegangan 0 Volt dan "1" pada level tegangan 5 Volt.
b. Dubutuhkan jumlah kabel yang sedikit, bisa hanya menggunakan 3 kabel yaitu saluran Transmit Data, saluran Receive Data, dan saluran Ground (Konfigurasi Null Modem)
c. Saat ini penggunaan mikrokontroller semakin populer. Kebanyakan mikrokontroller sudah dilengkapi dengan *SCI* (*Serial Communication Interface*) yang dapat digunakan untuk komunikasi dengan port serial komputer. (Agfianto ; 2002 : 5 -9).

II.5. Mikrokontroler ATmega8

AVR merupakan salah satu jenis mikrokontroler yang di dalamnya terdapat berbagai macam fungsi. Perbedaannya pada mikro yang pada umumnya digunakan seperti *MCS51* adalah pada AVR tidak perlu menggunakan *oscillator* eksternal karena di dalamnya sudah terdapat *internal oscillator*. Selain itu kelebihan dari AVR adalah memiliki *Power-On Reset*, yaitu tidak perlu ada tombol *reset* dari luar karena cukup hanya dengan mematikan *supply*, maka secara otomatis AVR akan melakukan *reset*. Untuk beberapa jenis AVR terdapat beberapa fungsi khusus seperti ADC, EEPROM sekitar 128 *byte* sampai dengan 512 *byte*. AVR *ATmega8* adalah *mikrokontroler* CMOS 8-*bit* berarsitektur AVR RISC yang memiliki 8K *byte in-System Programmable Flash*. Mikrokontroler dengan konsumsi daya rendah ini mampu mengeksekusi instruksi dengan kecepatan maksimum 16MIPS pada frekuensi 16MHz. Jika dibandingkan dengan *ATmega8L* perbedaannya hanya terletak pada besarnya tegangan yang diperlukan untuk bekerja. Untuk *ATmega8* tipe L, mikrokontroler ini dapat bekerja dengan tegangan antara 2,7 - 5,5 V sedangkan untuk *ATmega8* hanya dapat bekerja pada tegangan antara 4,5 – 5,5 V.

II.5.1 Konfigurasi Pin Atmega8



Gambar II.4. Konfigurasi Pin Atmega8

Sumber : (http://elib.unikom.ac.id/files/disk1/535/jbptunikompp-gdl-indrapurna-26711-5-unikom_i-i.pdf.)

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

1. VCC

Merupakan *supply* tegangan digital.

2. GND

Merupakan *ground* untuk semua komponen yang membutuhkan *grounding*.

3. Port B (PB7...PB0)

Didalam Port B terdapat XTAL1, XTAL2, TOSC1, TOSC2. Jumlah *Port* B adalah 8 buah pin, mulai dari pin B.0 sampai dengan B.7. Tiap pin dapat digunakan sebagai *input* maupun *output*. *Port* B merupakan sebuah 8-*bit bi-*

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

4. Port C (PC5…PC0)

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

5. RESET/PC6

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

pin ini rendah dan pulsa yang ada lebih pendek dari pulsa minimum, maka akan menghasilkan suatu kondisi *reset* meskipun *clock*-nya tidak bekerja.

6. Port D (PD7…PD0)

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

7. AVcc

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

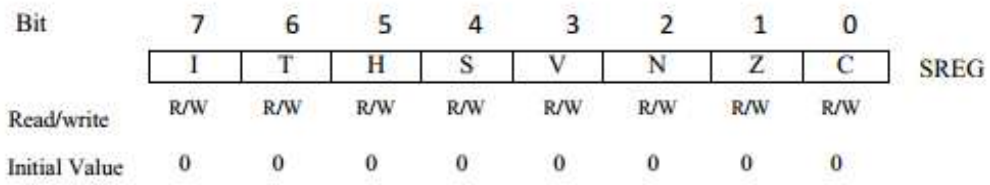
8. AREF

Merupakan pin referensi jika menggunakan ADC

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

Instruction Set Reference. Dalam hal ini untuk beberapa kasus dapat membuang penggunaan kebutuhan instrukasi perbandingan yang telah didedikasikan

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



Gambar II.5. Status Register ATMega8

Sumber : (http://elib.unikom.ac.id/files/disk1/535/jbptunikompp-gdl-indrapurna-26711-5-unikom_i-i.pdf.)

1. Bit 7(I)

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

2. Bit 6(T)

Merupakan *bit Copy Storage*. Instruksi *bit Copy Instructions* BLD (*Bit Load*) and BST (*Bit Store*) menggunakan bit ini sebagai asal atau tujuan untuk *bit* yang telah

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

3. Bit 5(H)

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

4. Bit 4(S)

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

5. Bit 3(V)

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

6. Bit 2(N)

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

7. Bit 1(Z)

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

8. Bit 0(C)

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

II.6. Bahasa C

Bahasa C dirancang oleh Dennis M. Ritche pada tahun 1972 di *AT&T Bell Laboratories*. Bahasa C pertama kali diganakan pada komputer DEC PDP-11 yang menggunakan sistem operasi UNIX. Bahasa C merupakan pengembangan dari bahasa BCPL (*Basic Combined Programming Language*) yang dibuat oleh Dr. Martin Richard yang kemudian dikembangkan oleh Ken Thompson menjadi bahsa pemrograman bahasa B.

Bahasa C memiliki beberapa keunggulan, diantaranya :

1. Proses eksekusi program yang cepat
2. Struktur bahasa yang baik (terstruktur)
3. Menyediakan pemrograman berorientasi objek

Bahasa C selain dapat dijalankan pada sistem operasi *Windows* juga dapat dijalankan pada sistem operasi *Linux*. Baris program yang kita buat pada sebuah *text* editor belum bisa dijalankan sebelum dieksekusi. Oleh karena itu program yang kita buat harus dieksekusi terlebih dahulu. Berikut adalah cara mengeksekusi program yang ditulis menggunakan bahasa C pada sistem operasi Linux.

gcc programku.c
gcc [option] program programku. c
Bahasa C memiliki struktur sebagai berikut :

1. Fungsi main ()
2. {}
3. Komentar
4. Aturan-aturan dalam bahasa C
5. Kata-kata Kunci dalam Bahasa C

6. Variabel
7. Inisialisasi Nilai Variabel
8. Tipe Data dan Format Data
9. Konstanta
10. Escape Sequence (Ema Utami dan Sukrisno ; 2005 : 1)

II.7. Pemodelan UML

Pemodelan adalah gambaran dari realita yang simpel dan dituangkan dalam bentuk pemetaan dengan aturan tertentu. (Rosa A.S, M. Shalahuddin; 2011: 116)

Pada perkembangan teknik pemrograman beroientasi objek, muncullah sebuah standarisasi bahasa pemodelan untuk pembangunan perangkat lunak yang dibangun dengan menggunakan teknik pemrograman berorientasi objek, yaitu *Unified Modeling Languange* (UML). *UML* hanya bergungsi untuk melakukan pemodelan. Jadi penggunaan *UML* tidak terbatas pada metodologi tertentu, meskipun pada kenyataannya *UML* paling banyak digunakan pada metodologi berorientasi objek. (Rosa A.S, M. Shalahuddin; 2011: 118)

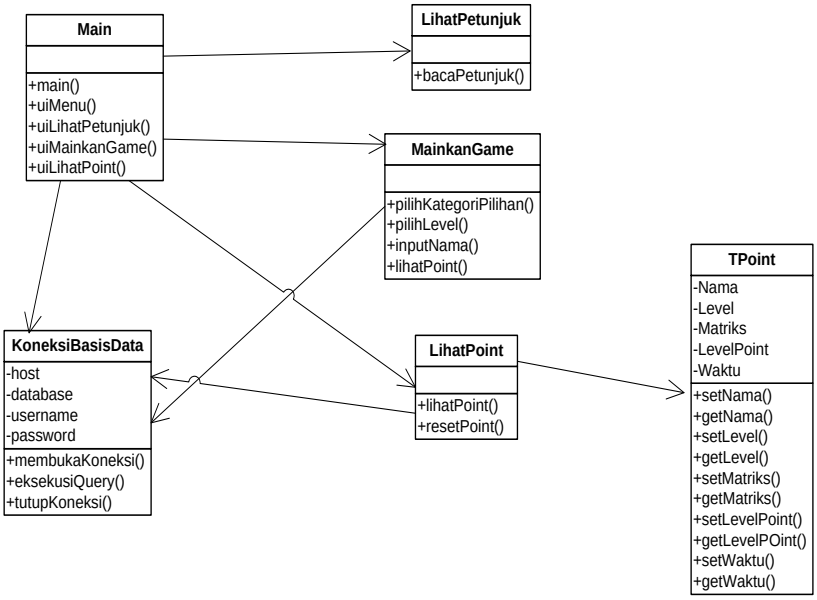
Diagram *UML* terdiri dari 13 macam diagram yang dikelompokkan dalam 3 kategori yaitu :

1. Structure Diagram

Yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan. Yang termasuk dalam structure diagram adalah sebagai berikut:

a. Class diagram

Digaram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefenisian kelas-kelas yang akan dibuat untuk membangun sistem. (Rosa A.S, M. Shalahuddin; 2011: 122). Contoh *class diagram* dapat dilihat pada gambar II.6.

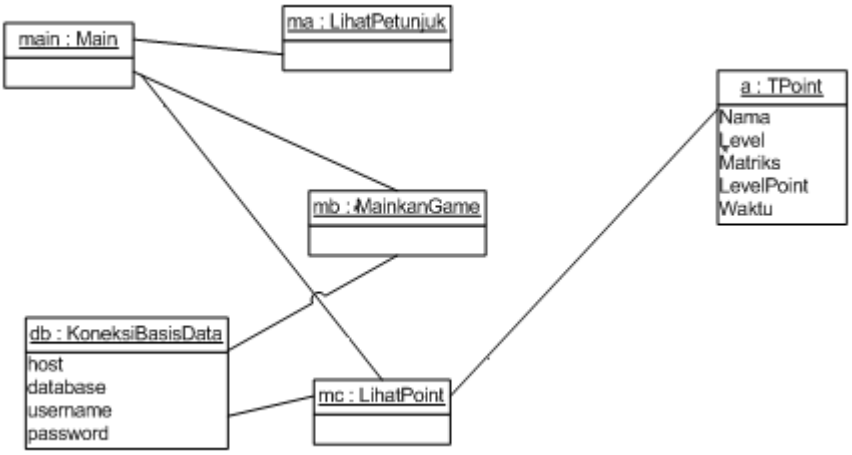


Gambar II.6. Contoh *Class* Diagram

Sumber : (Rosa. A.S. dan M. Shalahuddin ; 2011 : 161)

b. Object diagram

Diagram objek menggambarkan struktur sistem dari segi penamaan objek dan jalannya objek dalam sistem. (Rosa A.S, M. Shalahuddin; 2011: 124)

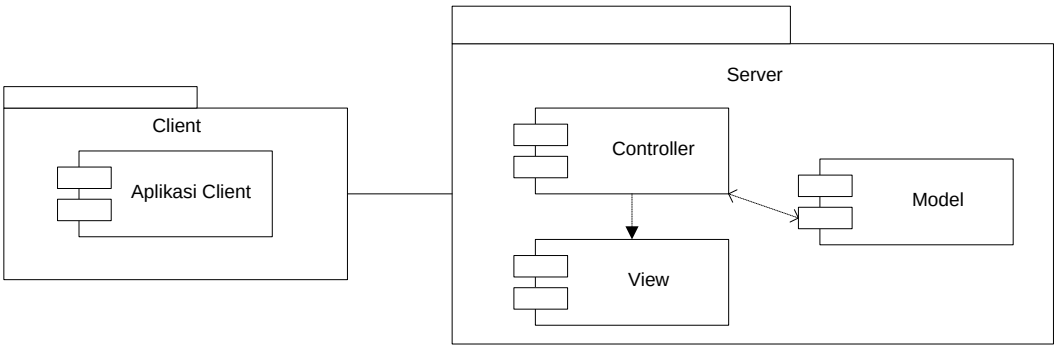


Gambar II.7. Contoh *Object* Diagram

Sumber : (Rosa A.S. dan M. Shalahuddin ; 2011 : 163)

c. Component diagram

Diagram komponen atau *component diagram* dibuat untuk menunjukkan organisasi dan ketergantungan di antara kumpulan komponen di dalam sebuah sistem. (Rosa A.S, M. Shalahuddin; 2011: 125)



Gambar II.8. Contoh *Component* Diagram

Sumber : (Rosa A.S, M. Shalahuddin; 2011: 125)

d. Composite diagram

Composite strukture diagram baru mulai ada pada UML versi 2.0, pada versi 1.x diagram ini belum muncul. Digaram ini dapat digunakanuntuk menggambarkan

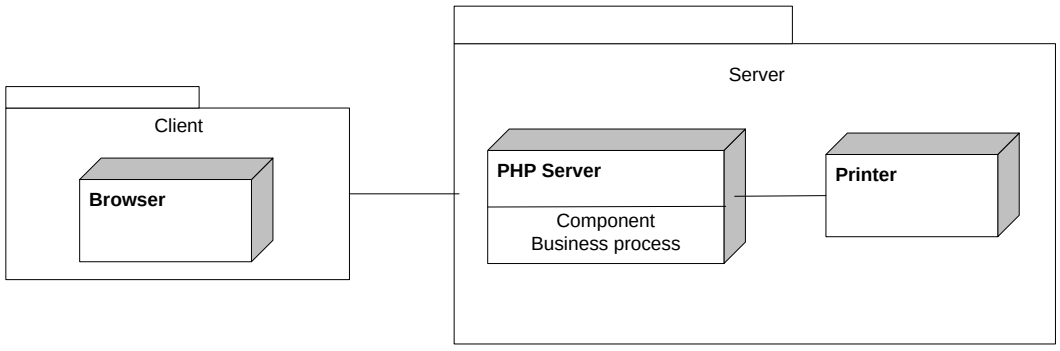
struktur dari bagian-bagian yang saling terhubung maupun mendeskripsikan struktur pada saat berjalan (*runtime*) dari *instance* yang saling terhubung. (Rosa A.S, M. Shalahuddin; 2011: 127)

e. Package diagram

Diagram ini meyediakan cara mengumpulkan elemen-elemen yang saling terkait dalam diagram UML. (Rosa A.S, M. Shalahuddin; 2011: 128)

f. Deployment diagram

Diagram deployment atau *deployment diagram* menunjukkan konfigurasi komponen dalam proses eksekusi aplikasi. (Rosa A.S, M. Shalahuddin; 2011: 129)



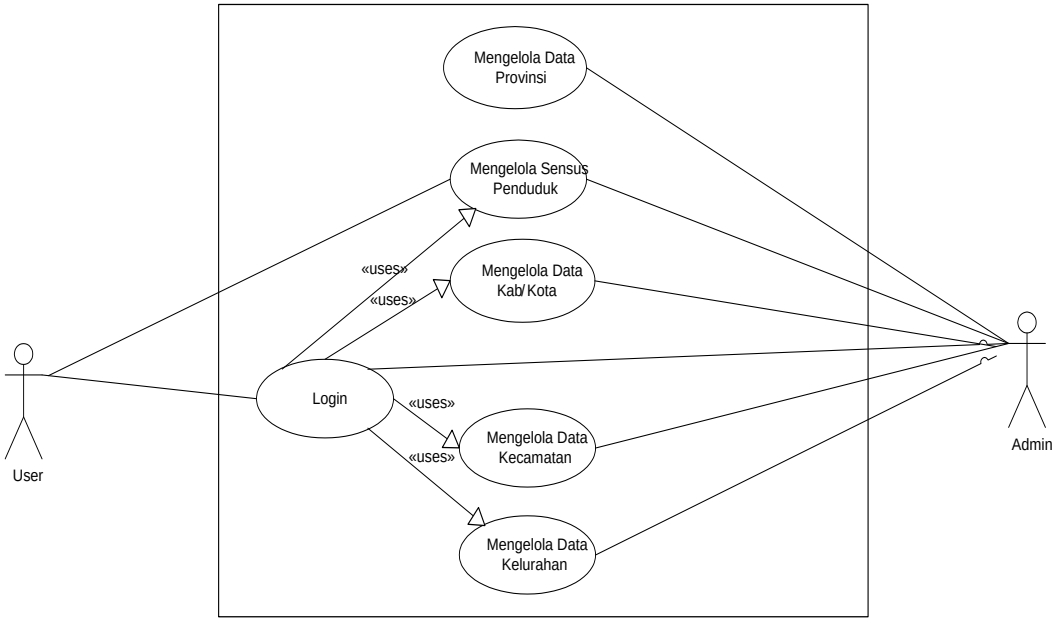
Gambar II.9. Contoh *Deployment* Diagram
Sumber : (Rosa A.S, M. Shalahuddin; 2011: 129)

2. Behavior Diagram

Yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem. Yang termasuk dalam *bahvior* diagram adalah sebagai berikut

a. Use case

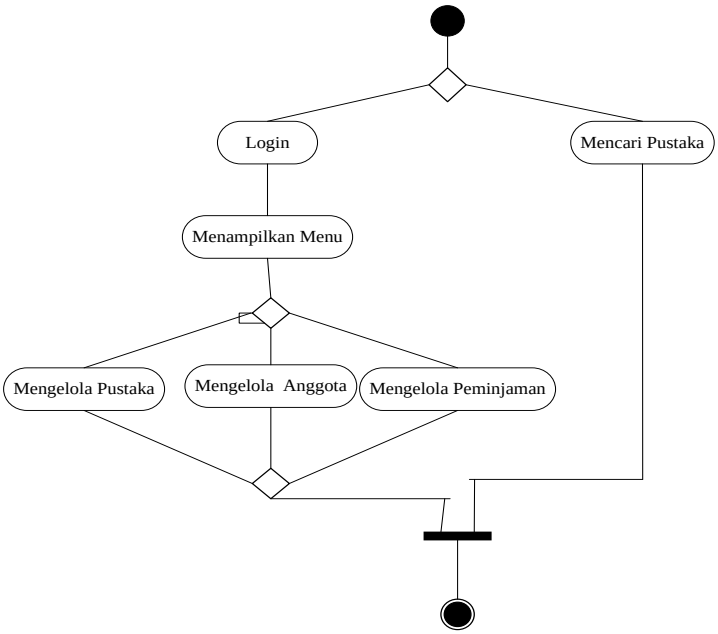
Use case atau diagram *use case* merupakan pemodelan untuk kelakuan (*bahavior*) sistem informasi yang akan dibuat. (Rosa A.S, M. Shalahuddin; 2011: 130). Lihat gambar II.10



Gambar II.10. Contoh *Use Case* Diagram
Sumber : (Rosa A.S. dan M. Shalahuddin ; 2011 : 160)

b. Activity diagram

Digaram aktivitas atau activity diagram menggambarkan workflow (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. (Rosa A.S, M. Shalahuddin; 2011: 134). Lihat gambar II.11

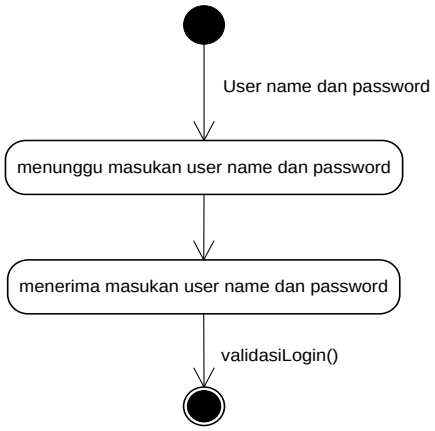


Gambar II.11. Contoh *Activity* Diagram

Sumber : (Rosa A.S. dan M. Shalahuddin ; 2011 : 177)

c. State machine diagram

Diagram mesin status digunakan untuk menggambarkan perubahan status atau transisi status dari sebuah mesin atau sistem. (Rosa A.S, M. Shalahuddin; 2011: 136).



Gambar II.12. Contoh *State Machine* Diagram

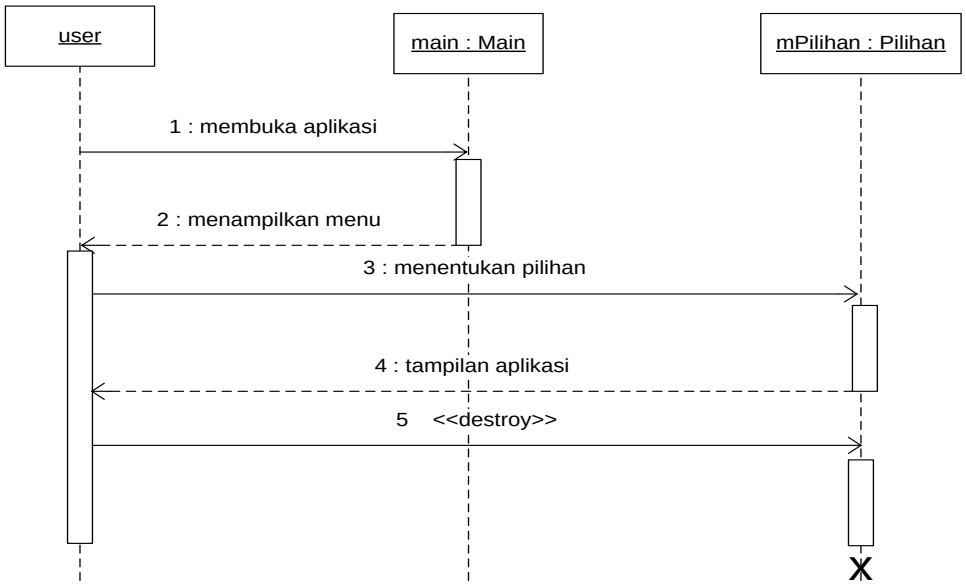
Sumber : (Rosa A.S, M. Shalahuddin; 2011: 174)

3. Interaction Diagram

Yaitu kumpulan diagram yang digunkan untuk menggambarkan interaksi antar sub sistem pada suatu sistem. Yang termasuk dalam *interaction* diagram adalah sebagai berikut :

a. Sequence diagram

Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. (Rosa A.S, M. Shalahuddin; 2011: 137). Lihat gambar II.13



Gambar II.13. Contoh *Sequence* Diagram

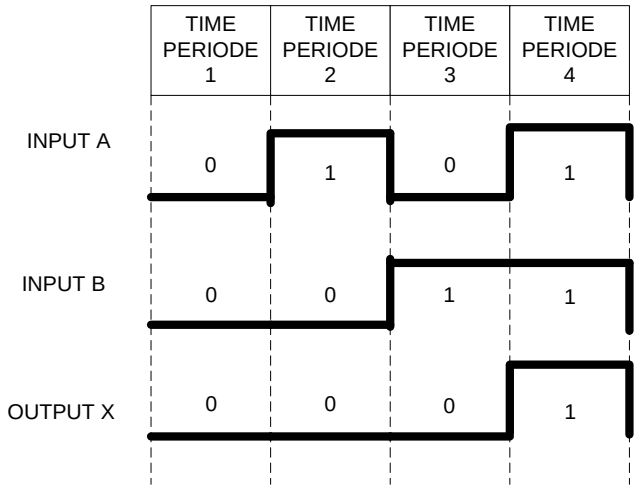
Sumber : (Rosa A.S, M. Shalahuddin; 2011: 164)

b. Communication diagram

Communication diagram atau diagram komunikasi menggambarkan interaksi antar objek /bagian dalam bentuk urutan pengiriman pesan. (Rosa A.S, M. Shalahuddin; 2011: 140).

c. Timing diagram

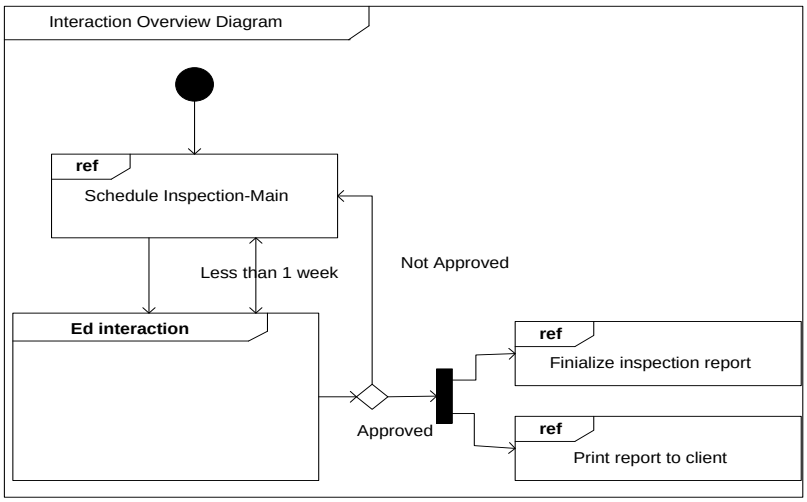
Timing diagram merupakan diagram yang fokus pada penggambaran terkait batasan waktu. (Rosa A.S, M. Shalahuddin; 2011: 141). Lihat gambar II.14



Gambar II.14. Contoh *Timing* Diagram
Sumber : (Rosa A.S, M. Shalahuddin; 2011: 142)

d. *Interaction overview diagram*

Diagram ini mirip dengan diagram aktivitas yang berfungsi untuk menggambarkan sekumpulan urutan aktivitas. (Rosa A.S, M. Shalahuddin; 2011: 143)



Gambar II.15. Contoh *Interaction Overview* Diagram

Sumber : (Rosa A.S, M. Shalahuddin; 2011: 145)

II.8. Flowchart

Tujuan utama dari pengunaan *flowchart* adalah untuk menggambarkan suatu tahapan penyelesaian masalah secara sederhana, terurai, rapi dan jelas dengan menggunakan simbol-simbol yang standar. (Budi Sutejo dan Michael AN, 2004 : 46)

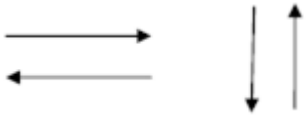
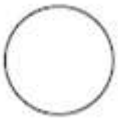
II.8.1. Simbol-simbol *Flowchart*

Simbol-simbol yang di pakai dalam *flowchart* dibagi menjadi 3 kelompok

1) *Flow direction symbols*

Digunakan untuk menghubungkan simbol satu dengan yang lain. Disebut juga *connecting line*. Lihat tabel II.2

Tabel II.2. Simbol *Flow Direction*

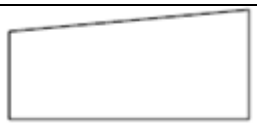
Simbol	Keterangan
	Simbol arus/*flow*, yaitu menyatakan jalannya arus suatu proses.
	Simbol *communication link*, yaitu menyatakan transmisi data dari suatu lokasi ke lokasi lain
	Simbol *connector*, berfungsi menyatakan sambungan dari proses ke proses lainnya dalam halaman yang sama
	Simbol *offline connector*, meyatakan sambungan dari proses ke proses lain dalam halaman yang berbeda

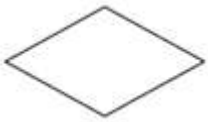
Sumber : (rama.staff.gunadarma.ac.id/.../2+definisi+dan+simbol+Flowchart.pdf)

2) *Processing symbols*

Menunjukan jenis operasi pengolahan dalam suatu proses / prosedur . (lihat tabel II.3)

Tabel II.3. Simbol *Flow* Proses

Simbol	Keterangan
	Simbol proses, yaitu menyatakan suatu tindakan (proses) yang dilakukan oleh komputer
	Simbol manual, yaitu menyatakan suatu tindakan (proses) yang tidak dilakukan oleh komputer
	Simbol *offline storage*, menyatakan data dalam simbol ini akan disimpan ke suatu media tertentu.
	Simbol manual *input*, memasukkan data secara manual menggunakan online keyboard

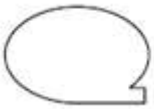
	Simbol *decision*, yaitu menunjukkan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban : ya atau tidak
	Simbol *predefined process* yaitu menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberi harga awal
	Simbol terminal, menyatakan awal atau akhir suatu program
	Simbol *keying operation* menyatakan segala jenis operasi yang diproses menggunakan suatu mesin yang mempunyai keyboard

Sumber : (rama.staff.gunadarma.ac.id/.../2+definisi+dan+simbol+Flowchart.pdf)

3) *Input* / *Output symbols*

Menunjukkan jenis peralatan yang digunakan sebagai media *input* atau *output*. (lihat tabel II.4)

Tabel II.4. Simbol *Input*/*Output*

Simbol	Keterangan
	Simbol *input*/*output*, menyatakan *input* atau *output* yang tidak tergantung pada jenis peralatannya
	Simbol *punched card*, menyatakan *input* berasal dari kartu atau *output* ditulis pada kartu
	Simbol *magnetic tape*, menyatakan *input* berasal dari pita megnetis atau *output* di simpan ke dalam pita magentis
	Simbol *disk storage*, menyatakan *input* berasal dari *disk* atau *output* disimpan ke *disk*

	Simbol *documen*, mencetak keluaran dalam bentuk dokumen (melalui printer)
	Simbol *display*, mencetak keluaran dalam layar monitor

Sumber : (rama.staff.gunadarma.ac.id/.../2+definisi+dan+simbol+Flowchart.pdf)