

BAB II

KAJIAN TEORITIK DAN KERANGKA BERPIKIR

2.1. Kajian Teoritik

2.1.1. Definisi Rancang Bangun

Rancang merupakan serangkaian prosedur untuk menerjemahkan hasil analisa dari sebuah sistem ke dalam bahasa pemrograman untuk mendeskripsikan dengan *detail* bagaimana komponen-komponen sistem diimplementasikan. Sedangkan pengertian bangun atau pembangunan sistem adalah kegiatan menciptakan sistem baru maupun mengganti atau memperbaiki sistem yang telah ada baik secara keseluruhan maupun sebagian (Pressman, 2002).

Dengan demikian pengertian rancang bangun merupakan kegiatan menerjemahkan hasil analisa kedalam bentuk paket perangkat lunak kemudian menciptakan sistem tersebut ataupun memperbaiki sistem yang sudah ada³.

2.1.2. Pengertian Model

Model adalah rencana, representasi, atau deskripsi yang menjelaskan suatu objek, sistem, atau konsep yang seringkali berupa penyederhanaan atau idealisasi. Bentuknya dapat berupa model fisik (maket, bentuk prototipe), model citra (gambar rancangan, citra komputer), atau rumusan

³<http://indahpermata6.blogspot.com/2013/06/pengertian-rancang-bangun-dan-konsep.html>, diakses pada 14 Juli 2014 pukul 22:35 WIB.

matematis. Model biasa dibangun agar membantu proses identifikasi, penggambaran atau kategorisasi komponen-komponen yang relevan dari suatu proses. Sebuah model dapat dikatakan sempurna, jika ia mampu memperlihatkan semua aspek yang mendukung terjadinya suatu proses. Misalnya, dapat menunjukkan keterkaitan antara satu komponen dengan komponen lainnya dalam suatu proses dan keberadaannya dapat ditunjukkan secara nyata⁴.

2.1.3. Menejemen Pengelolaan Laboratorium

2.1.3.1. Pengertian Menejemen

Manejemen adalah suatu usaha bersama sekelompok manusia untuk mencapai tujuan. Organisasi secara efektif dan efisien menggunakan segala upaya dan daya yang ada. Manajemen fasilitas laboratorium sangat penting artinya bagi sebuah organisasi pendidikan sebagai usaha untuk mencapai tujuan.

Luther M. Gullick (1993 : 31) menyatakan fungsi-fungsi manajemen yang penting adalah perencanaan, pengorganisasian, pengadaan tenaga kerja, pemberian bimbingan, pengkoordinasian, pelaporan, dan penganggaran.

2.1.3.2. Definisi Pengelolaan

Pengelolaan merupakan suatu proses pendayagunaan sumber daya secara efektif dan efisien untuk mencapai suatu sasaran yang diharapkan secara

⁴ <https://rumakom.wordpress.com/2010/10/31/pengertian-konsep-dan-model-komunikasi/>, diakses pada 1 Februari 2015 pukul 1:20 WIB.

optimal dengan memperhatikan keberlanjutan fungsi sumber daya⁵. Henri Fayol (1996:86) menyatakan bahwa pengelolaan hendaknya dijalankan berkaitan dengan unsur atau fungsi-fungsi *manager*, yakni perencanaan, pengorganisasian, pemberian komando, pengkoordinasian, dan pengendalian.

2.1.3.3. Definisi Laboratorium

Laboratorium merupakan perangkat kelengkapan akademik dalam menunjang kegiatan proses belajar mengajar. Selain itu, laboratorium juga merupakan tempat melakukan aktifitas praktikum untuk mengaplikasikan teori ke dalam praktek⁶. Menurut Konsorsium Ilmu Pendidikan (Moh. Amien, 1988:1), laboratorium diartikan sebagai sarana, prasarana dan mekanisme kerja yang menunjang secara unik satu atau lebih dharma perguruan tinggi melalui pengalaman langsung dalam membentuk keterampilan, pemahaman, dan wawasan dalam pendidikan dan pengajaran serta dalam pengembangan ilmu dan teknologi dan pengabdian pada masyarakat. Sedangkan menurut PP No.25/1980, pasal 27, laboratorium/studio adalah sarana penunjang jurusan dalam satu atau seni tertentu sesuai dengan keperluan bidang studi yang bersangkutan⁷.

⁵<http://suciamelya.blogspot.com/2013/02/pengelolaan-laboratorium.html>, diakses pada 14 Juli 2014 pukul 22:51 WIB

⁶ Anonim. (2010). *Pengertian Laboratorium*. dari : <http://smileboys.blogspot.com/2008/05/pengertian-laboratorium.html>. diakses pada 14 Juli 2014 pukul 22:51 WIB.

⁷ Soemardjo, dan Sumardjito.(1996). *Aturan Perundangan Bangunan dan Sarana/Prasarana Sekolah*.Makalah, FPTK IKIP Yogyakarta. Hal 5

Demi kelancaran dan kenyamanan dalam penggunaan dan pemanfaatan laboratorium, maka perlu adanya pengelolaan dan penataan yang baik secara berkala yang dilakukan oleh penanggung jawab laboratorium. Pengelolaan laboratorium berkaitan dengan pengelola dan pengguna, fasilitas laboratorium (bangunan, peralatan laboratorium, spesimen biologi, bahan kimia), dan aktivitas yang dilaksanakan di laboratorium yang menjaga keberlanjutan fungsinya. Dalam pengelolaan laboratorium, pengelolannya meliputi beberapa aspek yaitu sebagai berikut:

1. Perencanaan

Perencanaan merupakan sebuah proses pemikiran yang sistematis, analitis, logis tentang kegiatan yang harus dilakukan, langkah-langkah, metode, SDM, tenaga dan dana yang dibutuhkan untuk mencapai tujuan yang telah ditentukan secara efektif dan efisien⁸.

2. Penataan

Tata letak peralatan adalah suatu bentuk usaha pengaturan penempatan peralatan di laboratorium, sehingga laboratorium tersebut berwujud dan memenuhi persyaratan untuk beroperasi⁹.

3. Pengadministrasian

Pengadministrasian sering juga disebut sebagai kegiatan menginventaris. Inventaris adalah suatu kegiatan dan usaha untuk

⁸ Anonim. (2010). *Pengelolaan lab.* dari : <http://www.psb-psma.org/content/blog/pengelolaan-lab-bagian-2-tata-letak-peralatan-laboratorium>. diakses pada 14 Juli 2014 pukul 22:51 WIB.

⁹<http://sulistyok.blogspot.com/2010/12/pengelolaan-dan-penataan-laboratorium.html>. diakses pada 14 Juli 2014 pukul 22:51 WIB.

menyediakan catatan tentang keadaan semua fasilitas, barang-barang yang dimiliki sekolah. Bagi SMA yang mempunyai beberapa laboratorium sangat penting untuk mendata fasilitas/menginventaris alat dan bahan laboratorium untuk kegiatan pembelajaran siswa. Dengan kegiatan inventarisasi yang memadai akan dapat diperoleh pedoman untuk mempersiapkan anggaran atau mempersiapkan kegiatan pada tahun yang akan datang¹⁰.

1. Pengamanan, perawatan, dan pengawasan

Pada dasarnya pengamanan, perawatan dan pengawasan laboratorium merupakan tanggung jawab bersama baik pengelola maupun pengguna. Mengatur dan memelihara laboratorium merupakan upaya agar laboratorium selalu tetap berfungsi sebagaimana mestinya. Sedangkan upaya menjaga keselamatan kerja mencakup usaha untuk selalu mencegah kemungkinan terjadinya kecelakaan sewaktu bekerja di laboratorium dan penanganannya bila terjadi kecelakaan¹¹.

2.1.4. RFID (*Radio Frequency Identification*)

2.1.4.1. Sejarah RFID

Pada tahun 1946, Léon Theremin menemukan alat mata-mata untuk pemerintah Uni Soviet yang dapat memancarkan kembali gelombang radio dengan informasi suara. Gelombang suara menggetarkan sebuah diafragma (*diaphragm*) yang mengubah sedikit bentuk resonator, yang kemudian memodulasi frekuensi radio yang terpantul. Walaupun alat

¹⁰ Arikunto, Suharsimi. (1993). *Organisasi dan Administrasi Pendidikan Teknologi dan Kejuruan*. Jakarta: Raja Grafindo Persada.

¹¹ Soenarto, dan Satunggalno.(1996). *Strategi Implementasi, Motivasi dan Evaluasi Kebijakan dalam Perawatan Sarana dan Prasarana Pendidikan*. Makalah, FPTK IKIP Yogyakarta. Hal 10

ini adalah sebuah alat pendengar mata-mata yang pasif dan bukan sebuah kartu/label identitas, alat ini diakui sebagai benda pertama dan salah satu nenek-moyang teknologi RFID. Beberapa publikasi menyatakan bahwa teknologi yang digunakan RFID telah ada semenjak awal era 1920-an, sementara beberapa sumber lainnya menyatakan bahwa sistem RFID baru muncul sekitar akhir era 1960-an¹².

Sebuah teknologi yang lebih mirip, IFF Transponder, ditemukan oleh Inggris pada tahun 1939, dan secara rutin digunakan oleh tentara sekutu di Perang Dunia II untuk mengidentifikasi pesawat tempur kawan atau lawan. Transponder semacam itu masih digunakan oleh pihak militer dan maskapai penerbangan hingga hari ini.

Paten Amerika Serikat nomor 3,713,148 atas nama Mario Cardullo pada tahun 1973 adalah nenek moyang pertama dari RFID modern; sebuah transponder radio pasif dengan memori ingatan. Alat pantulan tenaga pasif pertama didemonstrasikan pada tahun 1971 kepada Perusahaan Pelabuhan New York (*New York Port Authority*) dan pengguna potensial lainnya. Alat ini terdiri dari sebuah transponder dengan memori 16 bit untuk digunakan sebagai alat pembayaran bea.

Pada dasarnya, paten Cardullo meliputi penggunaan frekuensi radio, suara dan cahaya sebagai media transmisi. Rencana bisnis pertama yang diajukan kepada para investor pada tahun 1969 menampilkan

¹² <http://www.pekalongankab.go.id/fasilitas-web/artikel/teknologi/1615-rfid-radio-frequency-identification-sebagai-teknologi-sistem-pengidentifikasi-objek-otomatis.html>, diakses pada 27 Januari 2014 pukul 2:35 WIB.

penggunaan teknologi ini di bidang transportasi (identifikasi kendaraan otomotif, sistem pembayaran tol otomatis, plat nomor elektronik, manifest (daftar barang) elektronik, pendata rute kendaraan, pengawas kelaikan kendaraan), bidang perbankan (buku cek elektronik, kartu kredit elektronik), bidang keamanan (tanda pengenalan pegawai, pintu gerbang otomatis, pengawas akses) dan bidang kesehatan (identifikasi dan sejarah medis pasien).

Demonstrasi label RFID dengan teknologi tenaga pantulan, baik yang pasif maupun yang aktif, dilakukan di Laboratorium Sains Los Alamos pada tahun 1973. Alat ini diperasikan pada gelombang 915 MHz dan menggunakan label yang berkapasitas 12 bit.

Paten pertama yang menggunakan kata RFID diberikan kepada Charles Walton pada tahun 1983 (Paten Amerika Serikat nomor 4,384,288)¹³.

2.1.4.2. Definisi RFID

RFID adalah proses identifikasi seseorang atau objek dengan menggunakan frekuensi transmisi radio. RFID menggunakan frekuensi radio untuk membaca informasi dari sebuah *device* kecil yang disebut *tag* atau transponder (*Transmitter + Responder*). *Tag* RFID akan mengenali diri sendiri ketika mendeteksi sinyal dari *device* yang kompatibel, yaitu pembaca RFID (*RFID Reader*). RFID adalah teknologi identifikasi yang

¹³ <http://elfri.wordpress.com/2009/02/20/sekilas-teknologi-rfid/>, diakses pada 7 April 2014 pukul 0:40 WIB.

fleksibel, mudah digunakan, dan sangat cocok untuk operasi otomatis. RFID mengkombinasikan keunggulan yang tidak tersedia pada teknologi identifikasi yang lain.

RFID dapat disediakan dalam *device* yang hanya dapat dibaca saja (*Read Only*) atau dapat dibaca dan ditulis (*Read/Write*), tidak memerlukan kontak langsung maupun jalur cahaya untuk dapat beroperasi, dapat berfungsi pada berbagai variasi kondisi lingkungan, dan menyediakan tingkat integritas data yang tinggi. Sebagai tambahan, karena teknologi ini sulit untuk dipalsukan, maka RFID dapat menyediakan tingkat keamanan yang tinggi¹⁴.

2.1.4.3. Tag RFID

Tag RFID adalah *device* yang dibuat dari rangkaian elektronika dan antena yang terintegrasi di dalam rangkaian tersebut. Rangkaian elektronik dari *tag* RFID umumnya memiliki memori sehingga *tag* ini mempunyai kemampuan untuk menyimpan data. Memori pada *tag* secara dibagi menjadi sel-sel. Beberapa sel menyimpan data *Read Only*, misalnya *serial number* yang unik yang disimpan pada saat *tag* tersebut diproduksi. Sel lain pada RFID mungkin juga dapat ditulis dan dibaca secara berulang. Berdasarkan catu daya *tag*, *tag* RFID dapat digolongkan menjadi:

- *Tag* Aktif: yaitu *tag* yang catu dayanya diperoleh dari *batere*, sehingga akan mengurangi daya yang diperlukan oleh pembaca

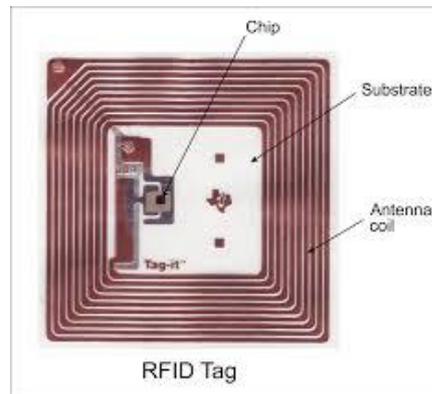
¹⁴ <https://mhs.blog.ui.ac.id/dony/2011/10/03/rfid-radio-frequency-identification/>, diakses pada 27 Januari 2014 pukul 2:55 WIB.

RFID dan *tag* dapat mengirimkan informasi dalam jarak yang lebih jauh. Kelemahan dari tipe *tag* ini adalah harganya yang mahal dan ukurannya yang lebih besar karena lebih kompleks. Semakin banyak fungsi yang dapat dilakukan oleh *tag* RFID maka rangkaianannya akan semakin kompleks dan ukurannya akan semakin besar.

- *Tag* Pasif: yaitu *tag* yang catu dayanya diperoleh dari medan yang dihasilkan oleh pembaca RFID. Rangkaianannya lebih sederhana, harganya jauh lebih murah, ukurannya kecil, dan lebih ringan. Kelemahannya adalah *tag* hanya dapat mengirimkan informasi dalam jarak yang dekat dan pembaca RFID harus menyediakan daya tambahan untuk *tag* RFID.

Tag RFID telah sering dipertimbangkan untuk digunakan sebagai *barcode* pada masa yang akan datang. Pembacaan informasi pada *tag* RFID tidak memerlukan kontak sama sekali. Karena kemampuan rangkaian terintegrasi yang *modern*, maka *tag* RFID dapat menyimpan jauh lebih banyak informasi dibandingkan dengan *barcode*. Fitur pembacaan jamak pada teknologi RFID sering disebut sebagai anti collision¹⁵. Contoh *Tag* RFID ada pada gambar 2.1 berikut ini.

¹⁵ Ibid, hal 14

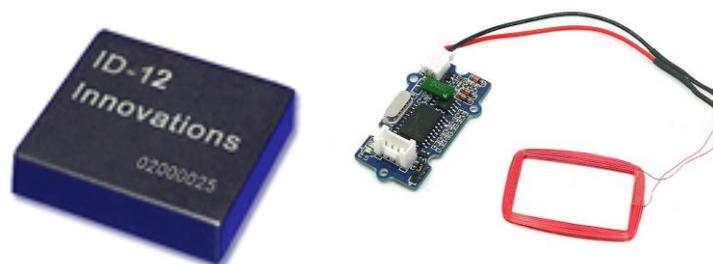


Gambar 2.1. Tag RFID

Sumber : <http://endtimetruth.com/wp-content/uploads/2014/01/RFID-chip-and-antenna-3.png>

2.1.4.4. RFID Reader

Untuk berfungsinya sistem RFID diperlukan sebuah *reader* atau alat *scanning-device* yang dapat membaca *tag* dengan benar dan mengomunikasikan hasilnya ke suatu basis data. Gambar 2.2. memperlihatkan RFID *reader*.



Gambar 2.2. RFID Reader

Sumber : <http://s3.electrodragon.com/wp-content/uploads/2012/04/125Khz-EM4100-RFID-Card-Read-Module-RDM630-UART-Output-Arduino-shield.jpg>

RFID *reader* merupakan penghubung antara program aplikasi yang ada pada komputer dengan antenna yang akan meradiasikan gelombang radio ke *tag* RFID. Gelombang radio yang diemisikan oleh

antena pada RFID *reader* akan berpropagasi pada ruangan disekitarnya. Akibatnya data dapat berpindah secara *wireless* dari tag ke RFID *reader*¹⁶.

2.1.4.5. Sistem Kerja RFID

Pada sistem RFID umumnya, *tag* atau *transponder* ditempelkan pada suatu objek. Setiap *tag* dapat membawa informasi yang unik, di antaranya: *serial number*, model, warna, tempat perakitan, dan data lain dari objek tersebut. Ketika *tag* ini melalui medan yang dihasilkan oleh pembaca RFID yang kompatibel, *tag* akan mentransmisikan informasi yang ada pada *tag* kepada pembaca RFID, sehingga proses identifikasi objek dapat dilakukan.

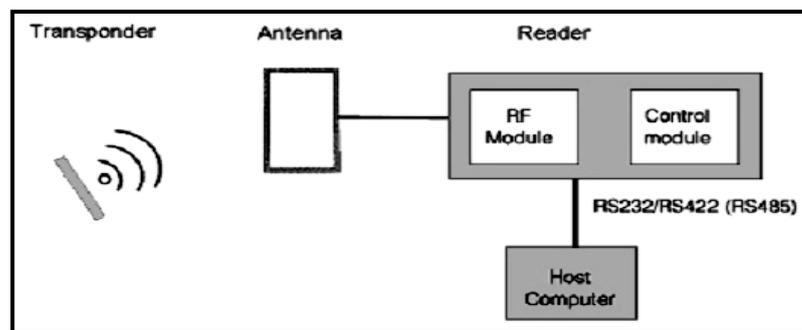
Sistem RFID terdiri dari empat komponen, di antaranya seperti berikut ini :

- *Tag* : Ini adalah *device* yang menyimpan informasi untuk identifikasi objek. *Tag* RFID sering juga disebut sebagai *transponder*.
- Antena : untuk mentransmisikan sinyal frekuensi radio antara pembaca RFID dengan *tag* RFID.
- Pembaca RFID : adalah *device* yang kompatibel dengan *tag* RFID yang akan berkomunikasi secara *wireless* dengan *tag*.
- *Software* Aplikasi : adalah aplikasi pada sebuah *workstation* atau PC yang dapat membaca data dari *tag* melalui pembaca RFID.

¹⁶ [www.id-innovations.com/Modules\(non write\).htm](http://www.id-innovations.com/Modules(non%20write).htm), diakses pada 27 Januari 2014 pukul 3:15 WIB

Baik *tag* dan pembaca RFID dilengkapi dengan antenna sehingga dapat menerima dan memancarkan gelombang elektromagnetik.

Sistem kerja RFID terdapat pada gambar 2.3. dibawah ini.



Gambar 2.3. Sistem Kerja RFID

Sumber : <http://elektronika-dasar.web.id/wp-content/uploads/2012/08/Sistem-RFID.jpg>

Sebuah pembaca RFID harus menyelesaikan dua buah tugas, yaitu:

1. Menerima perintah dari *software* aplikasi.
2. Berkomunikasi dengan *tag* RFID

Pembaca RFID adalah merupakan penghubung antara *software* aplikasi dengan antenna yang akan meradiasikan gelombang radio ke *tag* RFID. Gelombang radio yang diemisikan oleh antenna berpropagasi pada ruangan di sekitarnya. Akibatnya data dapat berpindah secara *wireless* ke *tag* RFID yang berada berdekatan dengan antenna.¹⁷

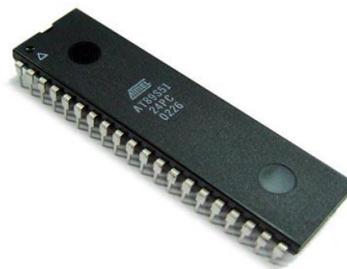
2.1.5. Mikrokontroler AT89S52

Mikrokontroler adalah mikroprocessor yang dirancang khusus untuk aplikasi kontrol, dan dilengkapi dengan ROM, RAM dan fasilitas

¹⁷ Artikel elektronika, sistem kerja rfid,dari ; <http://elektronika-dasar.web.id/2013/12/22/artikel-elektronika/pengertian-dan-komponen-radio-frequency-identification-rfid>, 27 Januari 2014 pukul 3:25 WIB.

I/O pada satu *chip*. AT89S52 adalah salah satu anggota dari keluarga MCS-51/52. AT89S52 dirancang oleh Atmel sesuai dengan instruksi standar dan susunan pin 80C5.

Mikrokontroler 89S52 merupakan versi terbaru dibandingkan mikrokontroler AT89C51 yang telah banyak digunakan saat ini. Mikrokontroler AT89S52 ialah mikrokomputer CMOS 8 bit dengan *8Kbyte Flash Programmable* dan *Erasable Read Only Memory (PEROM)*, yang memungkinkan memori program untuk dapat diprogram kembali. Mikrokontroler berteknologi memori *non volatile* kerapatan tinggi dari Atmel ini kompatibel dengan mikrokontroler standar industri MCS-51 baik pin kaki IC maupun set instruksinya serta harganya yang cukup murah. Oleh karena itu, sangatlah tepat jika mempelajari mikrokontroler jenis ini bagi pemula. AT89S52 mempunyai kelebihan yaitu mempunyai *flash* memori sebesar *8Kbyte*, RAM *256 byte* serta 2 buah data *pointer* 16 bit¹⁸. Bentuk mikrokontroler AT89S52 seperti gambar 2.4 berikut ini.



Gambar 2.4. Mikrokontroler AT89S52

Sumber : <http://i01.i.aliimg.com/wsphoto/v0/2047573805/10pcs-original-DIP-40-MPU-font-b-Microcontroller-b-font-font-b-Atmel-b-font-AT89S51.jpg>

¹⁸ <http://id.scribd.com/doc/60615459/MIKROKONTROLER-AT89S52>, diakses pada 26 Maret 2014 pukul 20:05 WIB.

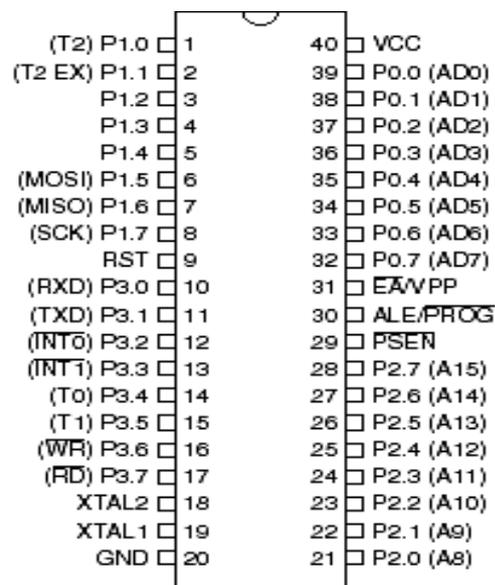
2.1.5.1. Spesifikasi dari AT89S52

- Sebuah CPU (*Central Processing Unit*) 8 Bit.
- Empat buah port I/O, yang masing masing terdiri dari 8 bit
- 8 *Kbytes In system Programmable* (ISP) *flash* memori dengan kemampuan 1000 kali baca/tulis tegangan kerja 4V-5.0V
- 256x8 bit RAM (*Random Acces Memory*) *internal*
- 32 Jalur I/O dapat diprogram
- 3 Buah 16 bit *Timer/Counter* (T0, T1, dan T2)
- 8 Sumber *interrupt*
- Sebuah *port serial* dengan *full duplex* UART (*Universal Asynchronous Receiver Transmitter*).
- *Watchdog timer*
- *Dual data pointer*
- Mode pemrograman ISP yang fleksibel (*Byte dan Page Mode*)
- Bekerja dengan rentang 0 – 33MHz
- Kecepatan maksimum pelaksanaan instruksi per siklus adalah 0,5 μ s pada frekuensi *clock* 24 MHz. Apabila frekuensi *clock* mikrokontroler yang digunakan adalah 12 MHz, maka kecepatan pelaksanaan instruksi adalah 1 μ s.¹⁹

¹⁹<http://ilmubawang.blogspot.com/2013/12/28/mengenal-mikrokontroler-at89s52.html>, diakses pada 26 Maret 2014 pukul 20:08 WIB.

2.1.5.2. Konfigurasi Pin Mikrokontroler AT89S52

Mikrokontroler AT89S52 terdiri atas 40 pin, yaitu 32 pin I/O, 2 pin *timer*, 2 pin I/O osilator, 2 pin *serial input* dan *output*, serta 2 pin *external input* dan *output*. Konfigurasi pin mikrokontroler AT89S52 di perlihatkan pada gambar 2.5. di bawah ini :



Gambar 2.5. Konfigurasi Mikrokontroler AT89S52

Sumber : http://1.bp.blogspot.com/-Zbz75HBnkXs/UQ8SOQ6-ycI/AAAAAAAAAEc/Wc_vPy871gE/s1600/mikrokontroler.JPG

Fungsi tiap-tiap pin adalah sebagai berikut:

- Port 0,

Merupakan *dual-purpose port*. Digunakan sebagai port *I/O* dan *Bus* alamat rendah yang berjumlah 8 pin, yaitu dimulai dari *port 0.0* sampai *port 0.7* (PO.1 sampai PO.7 atau AD0 sampai AD7).

- Port 1,

Hanya berfungsi sebagai *port I/O*. jumlahnya 8 pin, yaitu *port 1.0* sampai *port 1.7* (P1.1 sampai P1.7).

- Port 2,

Merupakan *dual-purpose port*. Digunakan sebagai *port I/O* dan *Bus* alamat tinggi yang berjumlah 8 pin, yaitu dimulai dari *port 2.0* sampai *port 2.7* (P2.1 sampai P2.7 atau A8 sampai A15).

- Port 3,

Merupakan *dual-purpose port*. Digunakan sebagai *port I/O* dan fungsi khusus. Jumlahnya 8 pin, yaitu dimulai dari *port 3.0* sampai *port 3.7* (P3.1 sampai P3.7).

Fungsi khusus pin 10-17 dari port 3 adalah seperti tabel 2.1 dibawah ini :

Tabel 2.1. Data Port 3 Pin 10 -17

No. Pin	Port. Pin	Nama Port	Fungsi Alternatif
10	P3.0	RXD	Menerima data untuk port serial
11	P3.1	TXD	Mengirim data untuk port serial
12	P3.2	$\overline{\text{INT 0}}$	Interrupt 0 eksternal
13	P3.3	$\overline{\text{INT 1}}$	Interrupt 1 eksternal
14	P3.4	T0	Timer 0 input eksternal
15	P3.5	T1	Timer 1 input eksternal
16	P3.6	$\overline{\text{WR}}$	Memori data eksternal write strobe
17	P3.7	$\overline{\text{RD}}$	Memori data eksternal read strobe

- ALE/PROG

Address Latch Enable (ALE) merupakan pin untuk menerima pulsa *output* yang akan mengunci *byte low* dari alamat tertentu, selama terjadi

akses memori *eksternal*. Pin ini juga merupakan *input* pulsa pemrograman selama pemrograman *flash* (paralel). Pada operasi normal, ALE mengeluarkan suatu laju *clock* konstan sebesar 1/6 dari frekuensi *oscillator*, dan dapat digunakan untuk pewaktu eksternal.

- PSEN

Program Store Enable. Merupakan sinyal keluaran yang mengaktifkan EPROM luar.

- (XTAL 1)

Pin masukan ke rangkaian osilator internal. Sebuah osilator kristal atau sumber osilator luar dapat digunakan.

- (XTAL 2)

Pin keluaran ke rangkaian osilator internal. Pin ini dipakai bila menggunakan osilator kristal.

- EA/ VPP

Eksternal Access Enable. Kondisi *high* untuk menjalankan program memori internal dan *low* untuk menjalankan program memori eksternal.

- RST (*Reset*)

Masukan reset aktif *high*. Pulsa transisi dari rendah ke tinggi akan mereset AT89S52. Pin ini dihubungkan dengan rangkaian *power on reset* yang terdiri dari sebuah kapasitor dan sebuah resistor yang berfungsi sebagai pembangkit frekuensi.

- Vcc, GND

Berfungsi untuk koneksi *power* atau catu daya yang dibutuhkan untuk mengoperasikan mikrokontroler AT89S52. Catu daya yang dibutuhkan adalah tegangan DC 5 Volt.

- Vpp atau *External Acces Enable*.

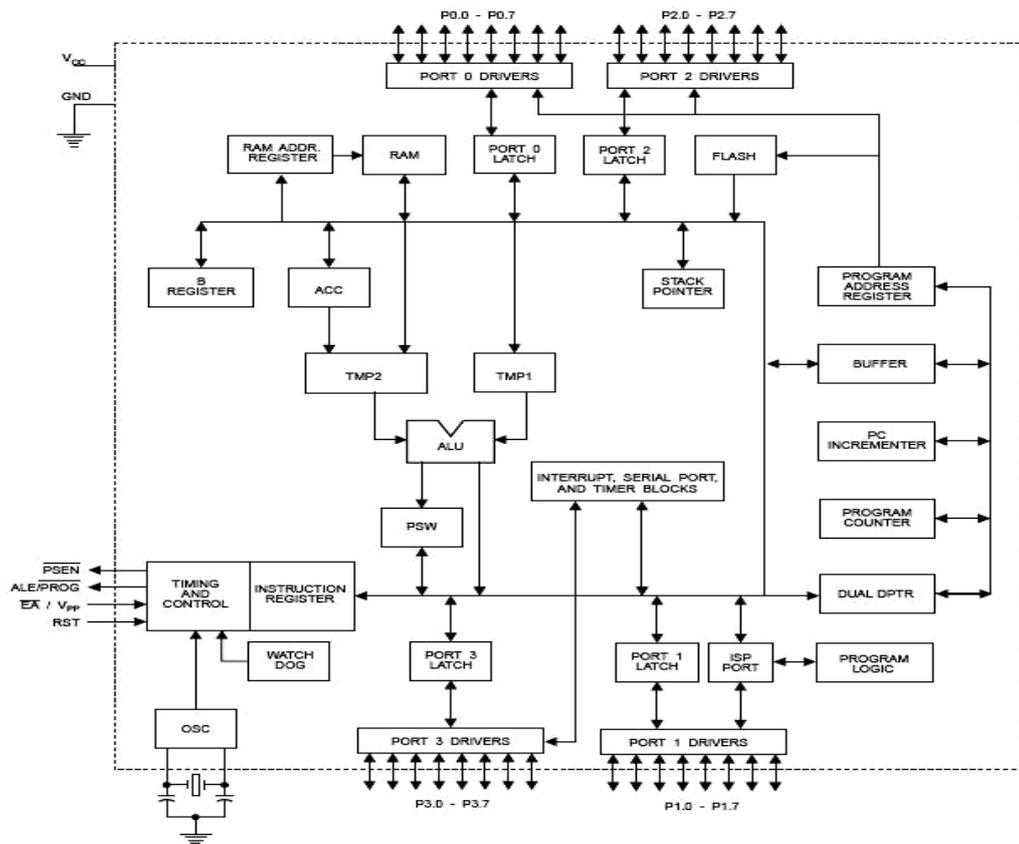
Dihubungkan ke *ground*, jika mikrokontroler akan mengeksekusi program dari memori eksternal 0000h hingga FFFFh. Selain itu, dihubungkan pula ke Vcc agar mikrokontroler mengakses program secara *internal*.²⁰

2.1.5.3. Blok Diagram Mikrokontroler

Blok Diagram Mikrokontroller AT89S52 dibangun berdasarkan arsitektur seperti di tunjukkan pada gambar 2.7 Seluruh bagian yang digambarkan pada gambar tersebut saling berhubungan melalui *internal bus* - 8bit menelusuri seluruh bagian keping.

Bus tersebut kemudian dihubungkan keluar melalui *port input/output* apabila memori atau ekspansi diperlukan. Setiap blok diagram memiliki hubungan fungsi yang erat dengan blok fungsi lainnya, yang bekerja secara sinkron sesuai dengan akses pemrograman. Adapun blok diagram Mikrokontroler AT89S52 dapat dilihat pada gambar 2.6 berikut ini.

²⁰ Iswanto. 2011. Belajar Mikrokontroler AT89S52 dengan bahasa C. Yogyakarta: CV.Andi Offset. hal. 4



Gambar 2.6. Blok Diagram Mikrokontroler AT89S52

Sumber : http://www.toko-elektronika.com/tutorial/uc1_fichiers/image004.jpg

a. ALU (*Aritmatic Logic Unit*)

ALU adalah suatu unit yang melaksanakan proses aritmatik dan logika seperti penjumlahan, pengurangan, pembagian, AND, OR, X – OR, rotasi, clear dan komplemen operasi percabangan.

b. Akumulator

Akumulator adalah merupakan register aritmatika yang berfungsi sebagai penampung data sebelum dan sesudah proses. Sebagian besar instruksi pemrosesan pada AT89S2051 menggunakan akumulator sebagai operand sumber atau tujuan pengiriman data dan ke *port*.

c. Register B

Register B digunakan selama operasi perkalian/pembagian 8 bit dan dapat juga digunakan sebagai register operand sumber atau operand tujuan.

d. *Stack pointer*

Stack pointer digunakan sebagai tempat penyimpanan variabel data yang ditindih dalam memori atau sebagai register petunjuk.

e. RAM (*Random Acces Memory*)

RAM adalah memori yang dapat dibaca atau ditulis. Data dalam RAM akan terhapus (*volatile*) bila catu daya dihilangkan. Karena sifat RAM yang *volatile* ini, maka program mikrokontroler tidak disimpan dalam RAM. RAM pada IC ini mempunyai kapasitas sebesar 128 *byte* x 8 bit.

f. TMP1/TMP2 (*Timer/Counter*)

TMP1/TMP2 berfungsi sebagai *timer/counter* 16 bit yang terangkai secara *internal*.

g. *Program Address Register*

Program address register merupakan alamat register dari program.

h. *Buffer*

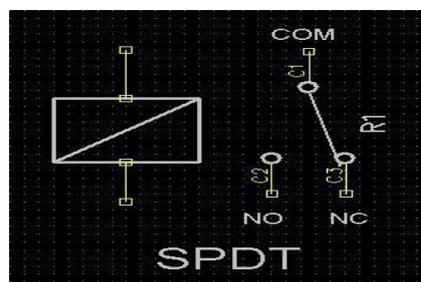
Buffer pada IC ini merupakan penyangga agar data yang dipindahkan dari suatu register ke register lain tetap atau tidak berantakan.

2.1.6. Relai

Dalam suatu sistem kontrol elektronik relai menjadi komponen yang sering dipakai, karena relai mudah dalam pengoperasiannya dan dapat di

kendalikan dari jarak yang jauh. Relai adalah komponen yang terdiri dari sebuah kumparan berinti besi yang akan menghasilkan elektromagnet ketika kumparannya dialiri oleh arus listrik. Elektromagnet ini kemudian menarik mekanisme kontak yang akan menghubungkan kontak *Normally-Open* (NO) dan membuka kontak *Normally-Closed* (NC). Sedikit menjelaskan, kata *Normally* disini berarti relai dalam keadaan *non-aktif* atau *non-energized*, atau gambangnya kumparan relai tidak dialiri arus. Jadi kontak *Normally-Open* (NO) adalah kontak yang pada saat **Normal** tidak terhubung, dan kontak *Normally-Closed* (NC) adalah kontak yang pada saat **Normal** terhubung²¹.

Gambar 2.7 dibawah ini adalah simbol dari komponen relai **SPDT** (*Single-Pole-Double-Throw*) yang berarti memiliki sebuah kontak NO dan sebuah kontak NC dengan sebuah *COMMON*. Pada saat kumparan tidak dialiri arus, maka kontak NC akan terhubung dengan COM. Jika kumparan dialiri arus, maka kontak akan bergerak dari NC ke NO, sehingga NO akan terhubung dengan *COM*.



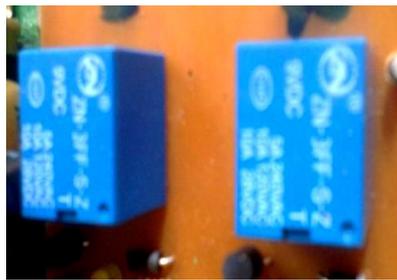
Gambar 2.7. Simbol Relai SPDT

Sumber : <https://telinks.files.wordpress.com/2010/04/dpdt.jpg?w=474>

²¹ George loveday, intisari elektronik (Jakarta : PT Elexmedia Komputindo, 1992) , hal 291

Karakteristik relai antara lain adalah tegangan kerja koil/kumparan. Ada yang 5Vdc, 12Vdc, 24Vdc, 36Vdc, hingga 48Vdc. Tegangan kerja adalah tegangan yang harus diberikan kepada koil agar relai dapat bekerja. Selain itu ada karakteristik kemampuan kontak relai. Bisa 3A, 5A, 10A, atau lebih. Maksudnya adalah arus maksimal yang mampu dialirkan oleh kontak relai adalah sesuai dengan karakteristiknya, jadi bisa 3A, 5A, atau 10A. Memang meskipun dipaksa untuk mengalirkan arus lebih besar juga tidak langsung rusak, tapi itu bukanlah praktek yang benar.²²

Berikut gambar 2.8 dibawah ini Relai yang digunakan.



Gambar 2.8. Relai
Sumber : Dokumentasi

2.1.6.1. Jenis-jenis Kontak Relai

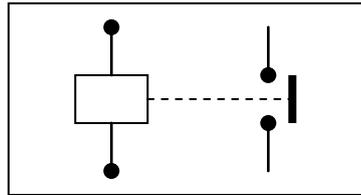
Kondisi atau posisi-posisi kontak relai terdiri dari :

1. Kontak NO (*Normaly Open*)

Pada posisi NO, kontak relai berada pada keadaan terbuka dari hubungan kontak dengan terminal kutub kontak. Jadi dapat dikatakan, pada posisi kumparan tidak aktif (tidak bertegangan)

²² <http://telinks.wordpress.com/2013/12/29/tag/relay-spdt>, diakses pada 26 Maret 2014 pukul 20:35 WIB.

kontak akan selalu terbuka, akan tetapi jika kumparan dialiri tegangan maka kontak NO akan menutup dan menjadi NC (*normaly close*). Contoh kontak relai posisi NO terdapat pada gambar 2.9.

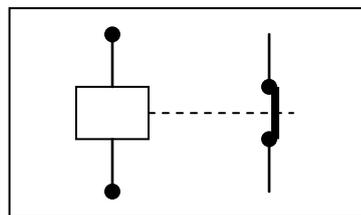


Gambar 2.9. Kontak Relai Posisi NO (*normaly open*)

Sumber : Dokumentasi

2. Kontak NC (*Normaly Close*)

Pada posisi NC, kontak-kontak relai berlawanan keadaan dengan kondisi kontak NO, pada keadaan normal (kumparan tidak dialiri arus listrik) posisi kontak sudah dalam keadaan terhubung (kontak), namun ketika kumparan aktif (dialiri arus listrik) maka posisi kontak akan berubah menjadi NO. Contoh kontak relai posisi NC terdapat pada gambar 2.10.

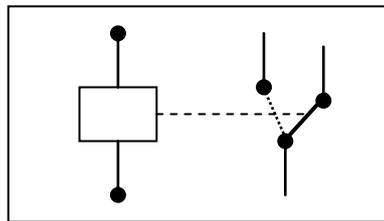


Gambar 2.10. Kontak relai posisi NC (*normaly close*)

Sumber : Dokumentasi

3. Kontak Tukar Sambung

Relai dengan karakteristik kontak seperti ini, mempunyai kontak tengah yang normalnya tertutup tetapi melepaskan diri dari posisi ini (awal) dan membuat kontak dengan yang lain bila kumparannya diberikan arus listrik²³. Contoh kontak relai posisi tukar sambung terdapat pada gambar 2.11.



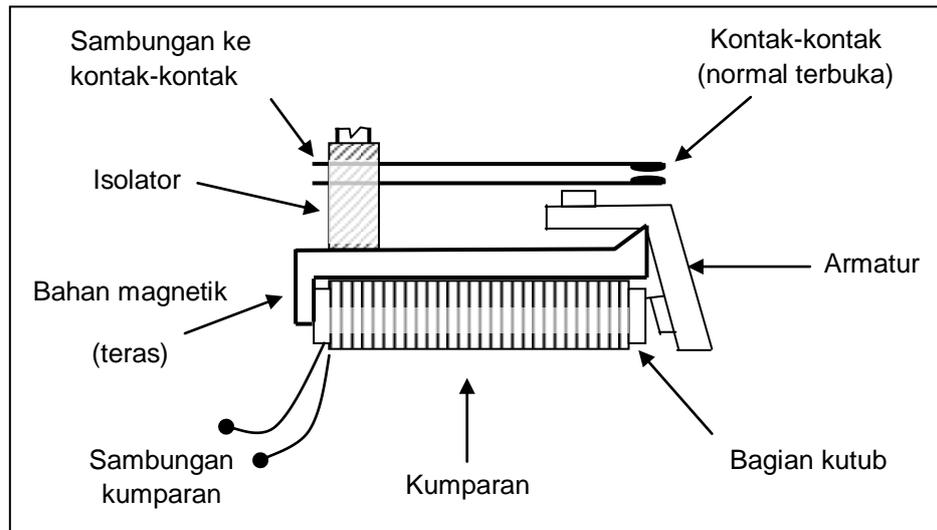
Gambar 2.11. Kontak relai posisi tukar sambung
Sumber : Dokumentasi

2.1.6.2. Prinsip Kerja Relai

Bila kumparan itu dienergikan oleh arus (biasanya jenis DC akan tetapi jenis AC juga ada), medan magnet yang terbangkitkan menarik armatur berporos, memaksanya bergerak cepat ke arah teras. Gerakan armatur ini dipakai melalui pengungkit, untuk menutup atau membuka kontak-kontak. Beberapa susunan kontak dapat dipakai, semuanya itu secara listrik terisolasi dari rangkaian kumparan²⁴. Prinsip kerja relai terdapat pada gambar 2.12 berikut ini.

²³ Brian Scaddan, sistem pengawatan dan pencarian kesalahan untuk instalator listrik (jakarta : Erlangga, 2005). hal.280-281

²⁴ Ibid, Hal 30



Gambar 2.12. Sebuah relai sederhana

Sumber : <http://www.glolab.com/relays/NO%20RELAY.jpg>

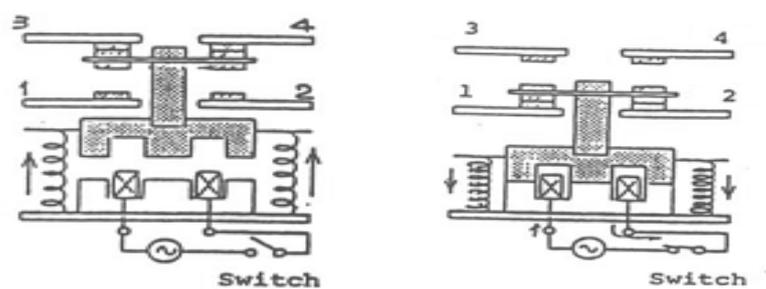
2.1.7. Kontaktor Magnet

Kontaktor magnet yaitu suatu alat penghubung listrik yang bekerja atas dasar magnet yang dapat menghubungkan antara sumber arus dengan muatan. Bila inti koil pada kontaktor diberikan arus, maka koil akan menjadi magnet dan menarik kontak sehingga kontaknya menjadi terhubung dan dapat mengalirkan arus listrik.

Kontaktor magnet atau saklar magnet merupakan saklar yang bekerja berdasarkan prinsip kemagnetan. Artinya saklar ini bekerja jika ada gaya kemagnetan pada penarik kontakannya. Magnet berfungsi sebagai penarik dan sebagai pelepas kontak-kontaknya dengan bantuan pegas pendorong. Sebuah kontaktor harus mampu mengalirkan dan memutuskan arus dalam keadaan kerja normal. Arus kerja normal ialah arus yang mengalir selama pemutusan tidak terjadi. Sebuah kontaktor dapat memiliki

koil yang bekerja pada tegangan DC atau AC. Pada tegangan AC, tegangan minimal adalah 85% tegangan kerja, apabila kurang maka kontaktor akan bergetar.

Ukuran dari kontaktor ditentukan oleh batas kemampuan arusnya. Biasanya pada kontaktor terdapat beberapa kontak, yaitu kontak normal membuka (*Normally Open* = NO) dan kontak normal menutup (*Normally Close* = NC). Kontak NO berarti saat kontaktor magnet belum bekerja kedudukannya membuka dan bila kontaktor bekerja kontak itu menutup/menghubung. Sedangkan kontak NC berarti saat kontaktor belum bekerja kedudukan kontakannya menutup dan bila kontaktor bekerja kontak itu membuka. Jadi fungsi kerja kontak NO dan NC berlawanan. Kontak NO dan NC bekerja membuka sesaat lebih cepat sebelum kontak NO menutup²⁵. Agar lebih jelas lihat gambar 2.13 mengenai konstruksi kontaktor magnet.

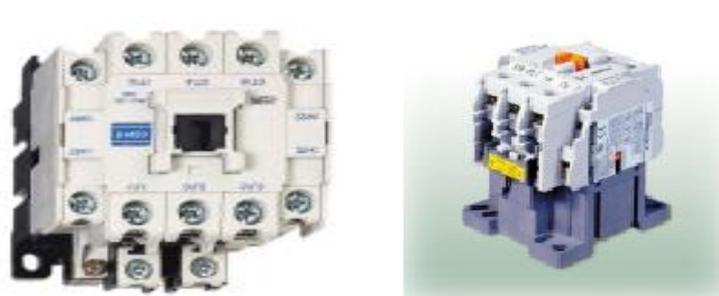


Gambar 2.13. Konstruksi Kontaktor Magnet

Sumber : <https://suriptoinstalasi.files.wordpress.com/2012/07/kontaktor-11.jpg>

²⁵ <http://ahmadaminudin311.blogspot.com/2012/02/pengertian-kontaktor-magnetik.html>, diakses pada 15 Januari 2014 pukul 16:15 WIB.

Pada gambar diatas, kontak 3 dan 4 adalah NC sedangkan kontak 1 dan 2 adalah NO. Apabila tidak ada arus maka kontak akan tetap diam. Tetapi apabila arus dialirkan dengan menutup *switch* maka kontak 3 dan 4 akan menjadi NO sedangkan kontak 1 dan 2 menjadi NC. Bentuk kontaktor magnet dapat dilihat seperti gambar 2.14 berikut ini.



Gambar 2.14. Kontaktor Magnet

Sumber : http://202.67.224.132/pdimage/51/4199151_contactorsn.jpg

Fungsi dari kontak-kontak dibuat untuk kontak utama dan kontak bantu. Kontak utama terdiri dari kontak NO dan kontak bantu terdiri dari kontak NO dan NC. Konstruksi dari kontak utama berbeda dengan kontak bantu, yang kontak utamanya mempunyai luas permukaan yang luas dan tebal. Kontak bantu luas permukaannya kecil dan tipis.

Kontaktor pada umumnya memiliki kontak utama untuk aliran 3 fasa. Dan juga memiliki beberapa kontak bantu untuk berbagai keperluan. Kontak utama digunakan untuk mengalirkan arus utama, yaitu arus yang diperlukan untuk beban, misalnya motor listrik, pesawat pemanas dan sebagainya. Sedangkan kontak bantu digunakan untuk mengalirkan arus bantu yaitu arus yang diperlukan untuk kumparan magnet, alat bantu

rangkaian, lampu lampu indikator, dan lain-lain. Notasi dan penomoran kontak-kontak kontaktor terdapat pada tabel 2.2 berikut ini :

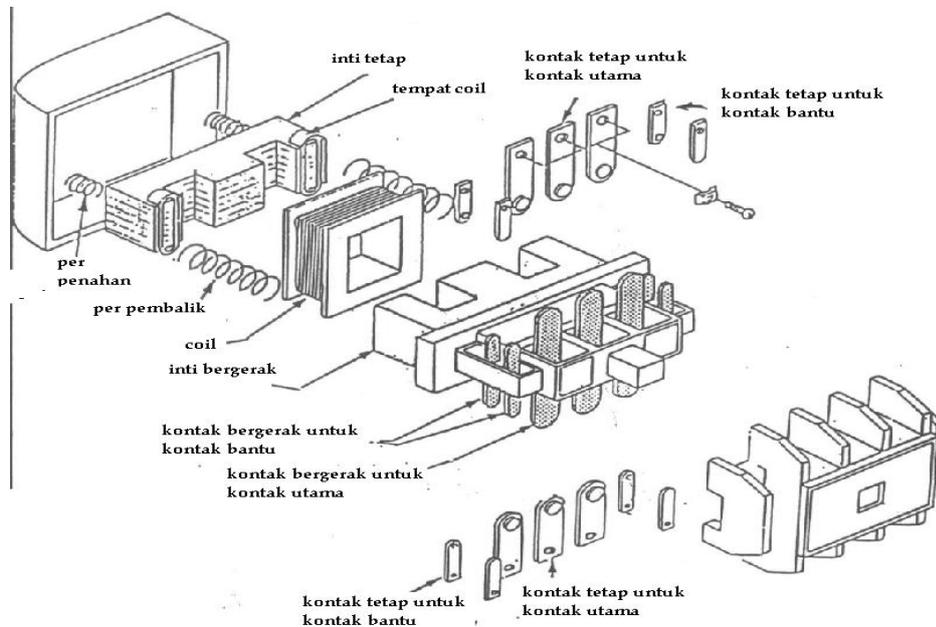
Tabel 2.2. Notasi dan Penomoran Kontak-Kontak Kontaktor

Kontak	Notasi		Jenis Kontak	Penggunaan
	Huruf	Angka		
Utama	L ₁ L ₂ L ₃	1 3 5	NO	Ke Jala-jala
	R S T	2 4 6	NO	Ke Motor
	U V W			
Bantu	-	13 14	NO	Pengunci
		19 20	NO	Fungsi Lain
		31 32		
		Dsb		
		21 22 41 42 Dsb	NC	Pengaman dan Fungsi lain
Kumparan Magnet (COIL)		Notasi Huruf		a - b A ₁ - A ₂

Dewasa ini kontaktor magnet lebih banyak digunakan di bidang industri dan laboratonium. Hal ini karena kontaktor mudah dikendalikan dari jarak jauh. Selain itu, dengan perlengkapan elektronik dapat mengamankan rangkaian listrik.

Tidak seperti sakelar mekanis, dalam merakit dan menggunakan kontaktor harus dipahami rangkaian pengendali (*control*) dan rangkaian utama. Rangkaian pengendali ialah rangkaian yang hanya menggambarkan bekerjanya kontaktor dengan kontak-kontak bantuannya. Sedangkan rangkaian utama ialah rangkaian yang khusus memberikan hubungan beban dengan sumber tegangan (jala-jala) 1 fasa atau 3 fasa. Bila kedua rangkaian itu dipadu akan menjadi rangkaian pengawatan (*circuit*

diagram). Agar lebih jelas mengenai konstruksi kontaktor magnet secara umum lihat gambar 2.15 dibawah ini.



Gambar 2.15. Konstruksi Kontaktor Magnet

Sumber : <https://suriptoinstalasi.files.wordpress.com/2012/07/kontaktor-6.jpg>

Konstruksi umum sebuah kontaktor dapat dilihat pada gambar diatas. Kontaktor memiliki kontak diam dan kontak - kontak yang bergerak apabila koil mendapat arus dari sumber. Kontaktor akan bekerja selama koil mendapat arus. Apabila arus terputus maka kontaktor akan kembali ke posisi semula.

2.1.8. Regulator Tegangan

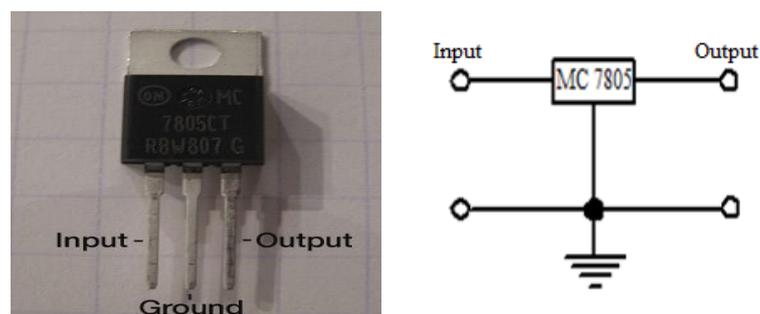
Regulator tegangan adalah bagian *power supply* yang berfungsi untuk memberikan stabilitas output pada suatu power supply. Regulator tegangan yang digunakan pada alat penuang kopi susu ini adalah regulator

tiga terminal. Regulator tiga terminal adalah “*Integrated Voltage Regulator Circuit*” yang dirancang untuk mempertahankan tegangan outputnya tetap dan mudah untuk dirangkai²⁶.

Keuntungannya adalah:

- Membutuhkan penambahan komponen luar yang sangat sedikit, ukuran kecil.
- Mempunyai proteksi terhadap arus hubung singkat.
- Mempunyai *automatic thermal shutdown*.
- Mempunyai tegangan *output* yang sangat konstan.
- Mempunyai arus rendah.
- Mempunyai *ripple output* yang sangat kecil.
- Pembiayaan rendah.

Gambar 2.16 berikut ini memperlihatkan contoh IC regulator tegangan positif tiga terminal.



Gambar 2.16. Regulator 7805 dan Simbol Rangka
Sumber : Dokumentasi

²⁶ Daryanto. 2010. Keterampilan Kejuruan Teknik Elektronika. Bandung: Satu Nusa. hal. 81

Seri LM 78XX adalah regulator dengan tiga terminal, dapat diperoleh berbagai tegangan tetap.

Beberapa IC regulator mempunyai kode yang dibuat oleh pabrik pembuat komponen, sebagai contoh : IC LM. 7805 AC Z, yang artinya sebagai berikut.

LM : *Linier Monolithic*

78 : Bagian nomor dasar yang menyatakan tegangan yang positif.

05 : Tegangan *Output*.

AC : Standar ketepatan

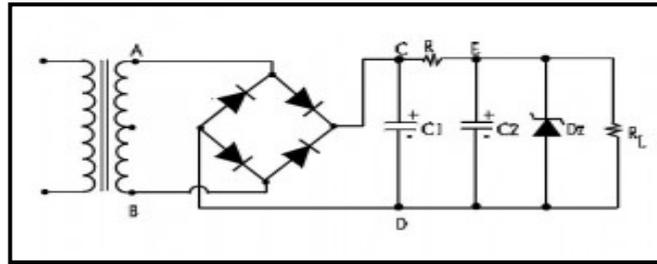
Z : Tipe pembungkus, ZTO-92 Plastik.

Seri LM 78XX dapat diperoleh dalam kemasan TO-3 aluminium, arus keluaran (*output*) 1A, boleh lebih asalkan IC regulator dilengkapi dengan pendingin (*Heatshick*). Regulator LM 78XXC mudah dipakai dan tambahan komponen-komponen ekstern tidak banyak.

Sifat-sifat IC regulator LM 78XX adalah sebagai berikut:

- Arus keluaran melebihi 1A.
- Pengamanan pembebanan lebih ternik.
- Tidak diperlukan komponen tambahan.
- Ada pengaman untuk transistor keluaran (*output*).
- Dapat diperoleh dalam kemasan TO-3 aluminium.

Contoh rangkaian lengkap catu daya menggunakan regulator tiga terminal IC 7805 untuk tegangan output 5 volt konstan ditunjukkan pada gambar 2.17 berikut.



Gambar 2.17. Rangkaian Catudaya dengan IC Regulator

Sumber: <http://elektronikadasar.web.id/wpcontent/uploads/2012/05/Regulator-Tegangan-Pada-Power-Supply.jpg>

Arus maksimum regulator IC yang dikirim ke beban tergantung pada tiga faktor, yaitu :

- Temperatur
- Perbedaan antara tegangan *input* dan *output* atau disebut diferensial *input output*.
- Arus beban.

Uraian lengkap mengenai parameter IC regulator dapat dilihat dari *data sheet* yang telah dibuat dari pabrik pembuat komponen. Contohnya IC 7805 C mempunyai *output* nominal 5 volt. Dari *data sheet* motorolla didapat *temperature junction* 250°C ($T_j+250^{\circ}\text{C}$), tegangan *output* antara *low* 4,8volt atau *high* 5,2 volt ; arus *output* $> 100\text{mA}$.²⁷

2.1.9. Pemrograman Delphi

Delphi adalah sebuah bahasa pemrograman tingkat tinggi sekelas dengan basic C. Bahasa pemrograman delphi disebut bahasa prosedural, artinya bahasa atau sintaknya mengikuti urutan tertentu atau prosedur.

²⁷ Ibid, hal. 36

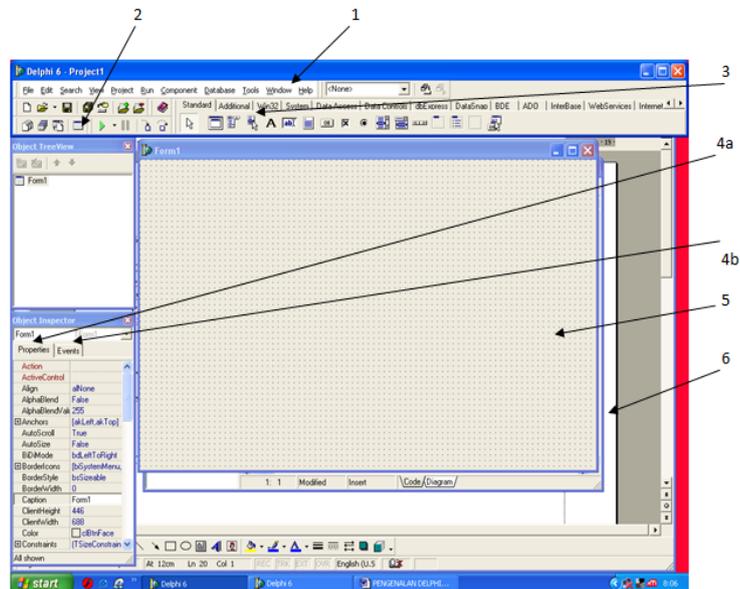
Delphi adalah sebuah perangkat lunak (bahasa pemrograman) untuk membuat program atau aplikasi komputer berbasis windows.

Delphi merupakan bahasa pemrograman berbasis objek, artinya semua komponen yang ada merupakan objek-objek. Ciri sebuah objek adalah memiliki nama, *properties*, dan *method/prosedure* dan dikemas menjadi satu kemasan (*encapsulate*). Delphi disebut juga visual programming, artinya komponen-komponen yang ada tidak hanya berupa teks (yang sebenarnya program kecil) tetapi juga muncul dengan berupa gambar-gambar.

2.1.9.1. Tampilan Borland Delphi 7.0

Tampilan pertama *Borland Delphi* adalah Proyek Baru yang diberi nama *Project1*. Pada pengoperasian berikutnya nama yang muncul adalah *Project2* dan seterusnya. Di dalam *Project1* terdapat *form* yang bernama *Form1*. Formulir yang pertama bernama *Form1*. Formulir berikutnya bernama *Form2* dan seterusnya. Aplikasi-aplikasi dibangun pada *Project1* ini. Nama *Project1* maupun *Form1* dapat diubah. Cara pengubahan akan diberikan pada keterangan lebih lanjut²⁸. Untuk lebih jelas mengenai tampilan borland delphi 7.0 bisa dilihat seperti gambar 2.18 berikut ini.

²⁸ Marcus Zakaria, Teddy. 2003. Pemrograman Delphi untuk Pemula. Hal 3



Gambar 2.18. Tampilan Borland Delphi

Sumber : Dokumentasi

Keterangan:

1. **Menu Bar** : Berisi menu perintah yang dikelompokkan dalam perintah-perintah sejenis (Misanya *File, Edit,* dsb.)
2. **Speed menu** : Disebut juga *Tool Bar*, yang dapat dipilih untuk mempercepat perintah tanpa melalui *Menu Bar*
3. **Component Palette** : Kumpulan obyek yang siap digunakan. Disebut *VCL (Visual Component Library)* atau biasa disingkat *Component*.

4. *Object Inspector* : Berisi properti dan *event* untuk dipilih

a. *Properties*

Bagian kiri adalah nama properti,
bagian kanan adalah nilai properti

b. *Event*

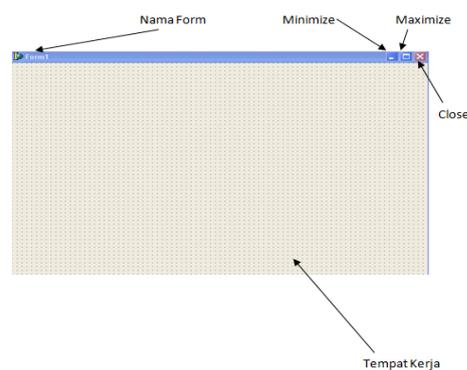
Bagian kiri adalah nama *event*,
bagian kanan adalah nilai *event*.

5. *Form Designer* : Tempat meletakkan komponen yang telah dipilih.

6. *Code Editor* : Tempat mengetikkan kode program.

A. *Form*

Form yang muncul pada layar Delphi adalah seperti gambar 2.19 berikut:



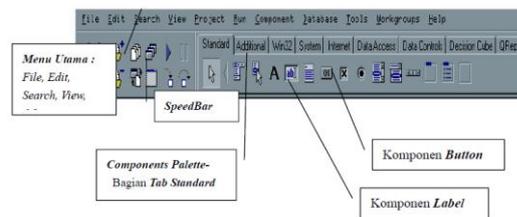
Gambar 2.19. Tampilan *Form* pada Program Delphi²⁹

Sumber : Dokumentasi

²⁹ Ibid, hal 40

B. *Components* pada *Form* Program Delphi

Komponen yang tersedia dapat dilihat pada *Components Palette*. Ada beberapa komponen yang dapat dipilih, seperti pada gambar 2.20 dibawah ini.



Gambar 2.20. Kumpulan *Components* pada Program Delphi³⁰

Sumber : Dokumentasi

a. *Menu File*

Menu ini berhubungan dengan *file* seperti membuat, menyimpan dan mengakhiri sebuah pekerjaan.

b. *Menu Edit*

Menu ini berhubungan dengan penyuntingan apa yang dikerjakan seperti *Undo*, *Redo*, *Cut*, *Copy*, *Paste*.

c. *Menu Search*

Menu ini berhubungan dengan pencarian dan penggantian data.

d. *Menu View*

Menu ini berhubungan dengan penampilan atau apa yang akan ditampilkan.

³⁰ Ibid, hal 40

e. *Menu Project*

Menu ini berhubungan dengan *project* yang sedang dibuat, misal unit yang akan ditambahkan ke *project* ini, unit apa yang akan dihapus.

f. *Menu Run*

Menu ini berhubungan dengan menjalankan program, mencari kesalahan (*debug*).

g. *Menu Component*

Menu ini berhubungan dengan komponen, misal menambah komponen baru dan menghapus komponen yang ada.

h. *Menu Database*

Menu ini berhubungan dengan Database, Database Form Wizard dan Database Explorer.

i. *Menu Tools*

Menu ini berhubungan dengan pengaturan/konfigurasi, tool-tool pembantu Delphi.

j. *Menu Help*

Menu ini berhubungan dengan informasi mengenai Delphi, Help / bantuan

C. *Object*

Berbasis objek, artinya pendekatan pembuatan program melalui objek-objek yang ada. Setiap objek akan memiliki *property* (*atribut*) dan *method* yang diaktifkan / dipicu oleh *event*.

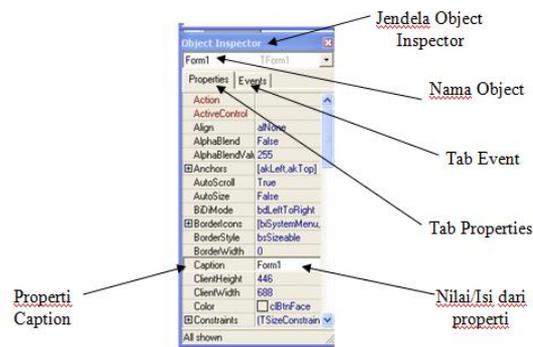
Sebuah komponen adalah sebuah objek pada *Palette* :

- Sebuah Objek, adalah sebuah komponen dalam *Component Palette*.
- Atau sesuatu yang dibuat melalui kode-kode / bahasa pemrograman.

Jadi sebuah objek adalah secara umum kelas dari kumpulan sesuatu. komponen pasti objek namun tidak selalu merupakan komponen, misal *TstringList* adalah sebuah objek (kumpulan karakter) dan bukan sebuah komponen.

D. *Properties*

Tiap-tiap *form* maupun komponen memiliki karakteristik tertentu dan dapat dipilih. *Properties* dapat dipilih melalui *Object Inspector*, pada halaman *Properties*. Jika tidak muncul *Object Inspector*, pilih menu *view*, kemudian *Object Inspector* atau tekan tombol F11. Pada *Object Inspector* terdapat dua halaman (*tab*) yaitu *Properties* dan *Event*. Berikut gambar 2.21 adalah tampilan *object inspector* pada program delphi.



Gambar 2.21. Tampilan *Object Inspector* pada Program Delphi³¹

Sumber : Dokumentasi

Properties digunakan untuk mengganti *property* sebuah obyek/komponen. *Event* digunakan untuk membuat prosedur yang diaktifkan (*triggerred*) lewat sebuah *event*. Berikut dibawah ini tabel 2.3 yang berisi nama-nama *property* pada tampilan delphi.

Tabel 2.3. Nama Nama *Property*³²

<i>Procedure</i>	Kumpulan perintah yang melakukan proses tertentu
<i>Function</i>	Hampir sama dengan <i>procedure</i> , tetapi proses tersebut dapat mengembalikan suatu hasil
<i>Method</i>	Fungsi yang telah tergabung pada sebuah komponen

E. Method

Sebuah ***method*** adalah sebuah ***function*** / fungsi yang tergabung dalam sebuah ***object***. Contoh ***ListBox*** (dapat berarti sebuah ***array of strings***) yang memiliki ***method*** (***Clear***) yang membuat ***Listbox*** tersebut menjadi kosong. ***Clear*** adalah sebuah ***method*** pada ***ListBox*** tersebut.

³¹ Ibid, hal 40

³² Ibid, hal 40

Contoh:

```

Begin
    ListBox1.Clear; // Mengosongkan isi ListBox
    ListBox1.Items.LoadFromFile('c:\Data1.txt');
    //properti Items (bertipe string) memiliki
    method untuk LoadFromFile
end;
```

F. Events

Sebuah *Event* adalah sebuah aksi pengguna (*User Action*) misal *Mouse Click*, *KeyPressed*. *SetiapEvents* diawali dengan kata 'On'. Berikut ini adalah tabel 2.4 yang berisi contoh jenis-jenis *event* pada program delphi.

Tabel 2.4. Contoh Jenis Jenis Events³³

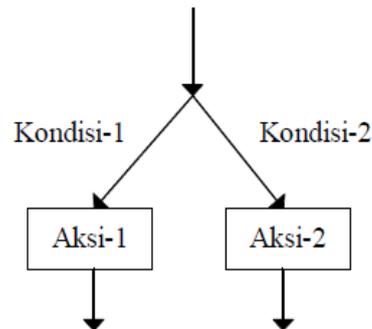
Nama event	Nama method
OnClick ..	Button1Click(Sender : TObject)
OnKeyDown ..	Button1KeyDown(Sender : TObject)
OnMouseMove ..	Button1MouseMove(Sender : TObject)

2.1.9.2. Perintah Dasar pada Program Delphi

A. Struktur Pemilihan

Pada beberapa kasus terkadang kita menginginkan komputer melakukan suatu aksi tertentu bila suatu kondisi terpenuhi. Keberadaan perintah bersyarat pada suatu program memberikan pencabangan proses seperti gambar 2.22 berikut ini.

³³ Ibid, hal 40



Gambar 2.22. Percabangan pada program³⁴

Sumber : Dokumentasi

Bahasa Pascal menyediakan beberapa perintah bersyarat yaitu :

- **IF .. THEN (Pemilihan 1 kasus)**

Perintah bersyarat If – then digunakan hanya melakukan 1 aksi bila kondisi dipenuhi. Bentuk sintaks dari perintah bersyarat ini adalah:

```

If <kondisi_percabangan> then
Begin
    ... {aksi-1}
end;
  
```

- **IF ... THEN ... ELSE (Pemilihan 2 kasus)**

Perintah bersyarat If umumnya digunakan untuk melakukan percabangan sederhana (antara 2 atau 3 cabang). Atau untuk percabangan yang banyak, dimana kondisi yang menjadi prasyaratnya melibatkan lebih dari satu parameter. Bentuk sintaks dari perintah bersyarat ini adalah:

³⁴ Ibid, hal 40

```
If <kondisi_percabangan> then
Begin
    ... {aksi-1}
end
    else
Begin
    ... {aksi-2}
end;
end;
```

- **Percabangan bersarang (*Nested*)**

Percabangan bersarang terjadi apabila pada cabang terjadi kondisi yang menyebabkan terjadinya percabangan kembali³⁵. Jumlah dari kondisi cabang bisa lebih dari satu. Bentuk sintaks dari perintah bersyarat ini adalah:

```
If <kondisi_percabangan> then
Begin
    ... {aksi-1}
end
    else
Begin
    ... {aksi-2}
end
    else
Begin
    ... {aksi-3}
end;
end;
```

³⁵ Teknik Antarmuka Mikrokontroler dengan Komputer Berbasis Delphi hal. 52

- **CASE ... OF ... : Pemilihan N kasus**

Perintah bersyarat Case umumnya digunakan untuk kondisi dengan banyak percabangan. Syarat percabangan pada bentuk ini hanya boleh melibatkan satu buah parameter dengan tipe data bukan Real. Pemeriksaan kondisi di sini lebih tepat disebutkan dalam hubungan relasi sama dengan (=). Dengan demikian bila parameter bernilai tertentu maka dilakukan suatu aksi terkait, bila bernilai lain maka dilakukan aksi yang lain juga, demikian seterusnya. Bentuk sintaks dari perintah bersyarat ini adalah:

```

If <kondisi_percabangan1> then
Begin
    ... {aksi-1}
end
    else
If <kondisi_percabangan2> then
Begin
    ... {aksi-2}
end
    else
Begin
    ... {aksi-3}
End;

```

B. Stuktur Pengulangan

Dalam menyelesaikan masalah, terkadang kita harus melaku suatu proses yang sama lebih dari satu kali. Untuk itu perlu dibuat suatu

algoritma pengulangan. Pascal memberikan tiga alternatif pengulangan, yaitu dengan For, While, atau Repeat. Masing-masing memiliki karakteristik, yang akan dipelajari pada modul ini. Ada dua hal yang penting dalam melakukan merancang perintah pengulangan, yaitu:

- Inisialisasi awal.
- Nilai akhir pengulangan atau kondisi berhenti

Bahasa Pascal menyediakan beberapa perintah pengulangan yaitu :

▪ **FOR – TO - DO**

Pada pengulangan dengan For, inisialisasi awal dan kondisi akhir ditentukan dengan menggunakan suatu variable kendali yang nilainya dibatasi dalam suatu range tertentu.

```
For <variable_kendali> := <nilai_awal> to
<nilai_akhir> do

Begin

    ... {aksi}

End ;
```

Atau dengan perintah

```
For <variable_kendali> := <nilai_awal> downto
<nilai_akhir> do

Begin

    ... {aksi}

End ;
```

▪ **WHILE - DO**

Pada metoda pengulangan ini aksi hanya akan diproses bila kondisi pengulangan terpenuhi.

```
While <kondisi_ pengulangan> do  
Begin  
    ... {aksi}  
End ;
```

- **REPEAT – UNTIL**

Metoda pengulangan ini juga melakukan pengulangan berdasarkan pemeriksaan kondisi pengulangan. Hanya saja natur dari pengulangan ini adalah sistem seakan-akan memaksa untuk melakukan pengulangan, sampai di ketahui adanya kondisi berhenti. Berlawanan dengan While, yang akan memproses aksi hanya bila kondisi_pengulangan bernilai true, pada pengulangan Repeat, sistem akan memproses aksi selama kondisi_berhenti bernilai false. Dengan demikian aksi pasti akan selalu diproses (minimal satu kali). Pada tipe ini, pengulangan dapat terjadi terus-menerus (tidak pernah berhenti), yaitu bila kondisi berhenti tidak pernah bernilai true.

```
Repeat  
    ... {aksi}  
Until <kondisi_ berhenti>
```

2.1.9.3. Jenis Data pada Program Delphi³⁶

A. Jenis data *Integer / Ordinal*

Tipe data ini adalah tipe data yang berupa bilangan bulat yang akan digunakan untuk operasi matematika. Ada beberapa tipe data yang termasuk integer, yaitu antara lain:

1) Jenis data *Boolean*

Tipe data ini hanya terdiri dari nilai logika *True* dan *False*.

2) Jenis data *Floating Point /Real*

Tipe data ini meliputi bentuk bilangan desimal. Beberapa bentuk dari tipe data ini adalah:

3) Jenis Data *Enumerated*

Tipe data ini adalah berupa nilai urutan dari data yang telah dibuat.

4) Jenis data *Sub Range*

Tipe data ini memungkinkan kita membuat range sendiri. Jika kita mendeklarasikan 5 adalah a maka proses *compile* akan terus berjalan jika kita mendeklarasikan 15 adalah a maka proses *compile* tidak akan berjalan atau berhenti, sehingga terdapat *error* pada proses *compile* sehingga tidak dapat di *run*.

5) Jenis data *String dan Character*

Tipe data *String* adalah tipe data yang merupakan rangkaian dari huruf, angka atau *symbol* yang tidak mengandung harga (nilai). Tipe

³⁶ Marcus Zakaria, Teddy. 2004. Berlatih Tipe Data di Delphi. Hal 5

data *Character* adalah tipe data yang berkaitan dengan karakter (huruf, angka, simbol).

2.1.9.4. Menyimpan Form

Delphi ada 3 buah file utama (*.dpr, *.pas dan *.dfm).

1. *.dpr adalah file proyek yang dibuat berisi program kecil untuk, mendefinisikan Unit yang ada dalam file *project* menginisialisasi data, membangun *Form* dan menjalankan aplikasi
2. *.pas adalah unit-unit (*pascal code file*), bisa terdiri satu atau banyak file.
- 3 *.dfm adalah file definisi *Form* (*special pseudo code file*), bisa terdiri satu atau banyak file.

Setiap *Form* (.dfm) harus memiliki sebuah Unit (.pas), tetapi dapat memiliki Unit tanpa sebuah *Form* (hanya kode saja). Jika ingin melihat kode tersebut dapat mengklik kanan mouse, lalu pilih **VIEW AS TEXT** atau tekan tombol **Alt- F12**. Sebaiknya anda tidak mengubah isi kode tersebut, karena akan menyebabkan masalah serius. Untuk kembali ke bentuk *Form*, pilih **VIEW AS FORM** atau tekan tombol **Alt- F12** kembali.

2.1.10. Teori Dasar *Database*

2.1.10.1. Pengertian *Database Microsoft Access*

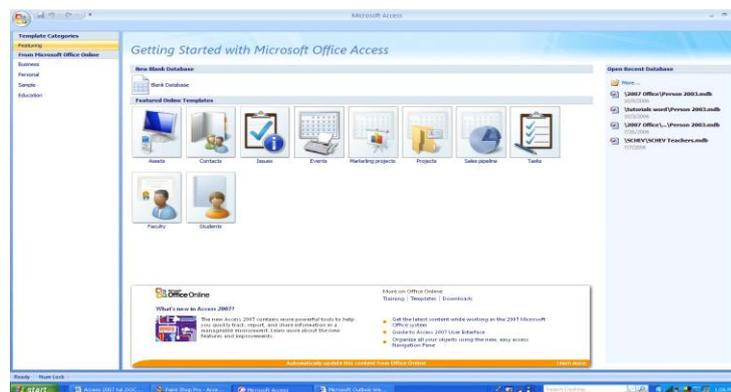
Microsoft Access adalah sebuah program aplikasi basis data komputer relasional yang ditujukan untuk kalangan rumahan dan perusahaan kecil hingga menengah. Aplikasi ini menggunakan mesin basis data *Microsoft Jet Database Engine*, dan juga menggunakan tampilan grafis yang *intuitif* sehingga memudahkan pengguna.

Istilah-istilah yang sering digunakan dalam *Microsoft Access* adalah sebagai berikut :

- ***Database*** adalah kumpulan informasi yang saling berkaitan
- ***object*** adalah sesuatu yang ada dalam database seperti *tabel*, *query*, *form* atau *macro*
- ***Table*** pengelompokan dari hubungan data yang berhubungan dalam field dan arsip pada datasheet. Dengan menggunakan *field* umum dalam 2 tabel, data bisa dikombinasikan. Banyak tabel bisa disimpan dalam *database* tunggal.
- ***Field*** adalah kolom dalam *datasheet* dan artinya data hasil ketikan yang disimpan dalam bentuk tabel. . Untuk tabel mailing list *table* juga termasuk *fileld* untuk nama awal, nama akhir, alamat, kota, negara, kode pos dan nomor telepon
- ***Record*** dalam baris dalam dan nilainya didefinisikan oleh *field*. dalam tabel mailing list, masing masing arsip akan berisi data untuk satu orang yang diatur oleh pembagian *field*.

- **Design View** tool yang disediakan untuk membuat *fields* dalam tabel.
- **Datasheet View** Mengijinkan anda untuk meng*update*, mengedit dan menghapus informasi dalam tabel.

Dibawah ini adalah gambar 2.23 yang berisi kotak dialog *microsoft access 2007*.



Gambar 2.23. Kotak Dialog *Microsoft Access 2007*³⁷
Sumber : Dokumentasi

2.1.10.2. *Microsoft Access 2007*

A. **Komponen *Microsoft Access***

Komponen-komponen Access 2007 seperti yang terlihat pada gambar adalah sebagai berikut :

1. *Tittle Bar*

Bagian ini terletak dibagian atas jendela *Microsoft Access*. Pada bagian ini tetulis *Microsoft Access* dan berguna menunjukkan jendela program aplikasi dan dokumen atau file database yang sedang dibuka.

³⁷ Office. Office tutorial Microsoft Access. Hal 1

2. *Menu Bar*

Pada bagian ini terdapat menu – menu yang digunakan untuk mengaktifkan perangkat –perangkat dalam membuat sebuah dokumen.

3. *Toolbar*

Merupakan sebuah balok berisi tombol – tombol yang digunakan untuk menjalankan perintah – perintah.

4. *Jendela Kerja*

Merupakan ruang kosong pada jendela aplikasi Access 2007. Di bagian ini nantinya berbagai objek database ditampilkan untuk diolah.

5. *Task Pane*

Merupakan sebuah panel yang menyediakan berbagai fasilitas bantuan maupun informasi – informasi tertentu seputar proses pengerjaan *database*.

6. *Status Bar*

Merupakan balok yang berisi informasi. Informasi tersebut antara lain adalah mode penyuntingan dan aktif atau tidaknya tombol *Num Lock*.

Dibawah ini adalah gambar 2.24 yang berisi komponen *microsoft access 2007*.



Gambar 2.24. Komponen Microsoft Access 2007³⁸
 Sumber : Dokumentasi

³⁸ Ibid, hal 55

B. Membuat Database Baru

Untuk memulai pekerjaan, terlebih dahulu sebuah *database* harus dibuat. Pembuatan sebuah *database* dilakukan dengan mengklik *Blank Database* pada tampilan utama *Microsoft Access 2007* seperti gambar 2.25.



Gambar 2.25. *Blank Database*³⁹
Sumber : Dokumentasi

Dengan melakukan hal tersebut, Kotak dialog akan muncul dengan permintaan untuk memasukkan nama *database* yang akan dibuat seperti terlihat pada gambar 2.26 berikut ini.



Gambar 2.26. Kotak Dialog *Blank Database*⁴⁰
Sumber : Dokumentasi

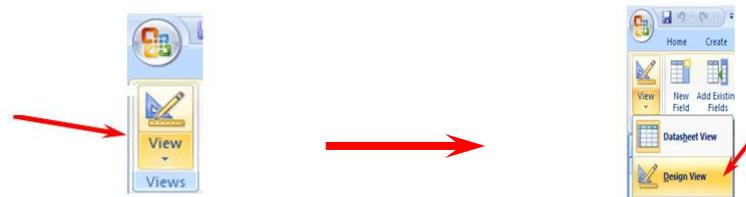
³⁹ Ibid, hal 55

⁴⁰ Ibid, hal 55

C. Pembuatan Tabel

Selanjutnya adalah membuat tabel, sebuah tabel terdiri dari baris dan kolom. Bagian baris merupakan data yang disimpan yang biasa disebut dengan *record*. Sedangkan bagian kolom merupakan pengelompokan *record-record* dan disebut *field*.

Perancangan tabel dengan *Design View* adalah perancangan tabel dengan terlebih dahulu menetapkan struktur suatu tabel. Perancangan tabel dengan *Design View* dilakukan dengan klik *icon view* pada *toolbar window database* dan pilih *Design View* seperti terlihat pada gambar 2.27.



Gambar 2.27. a.) Icon view b.) Design View⁴¹

Sumber : Dokumentasi

Setelah memilih dengan tampilan *Design view*, maka akan ada permintaan untuk menyimpan tabel yang akan dibuat dengan nama yang dapat dimasukkan. Seperti gambar 2.28.



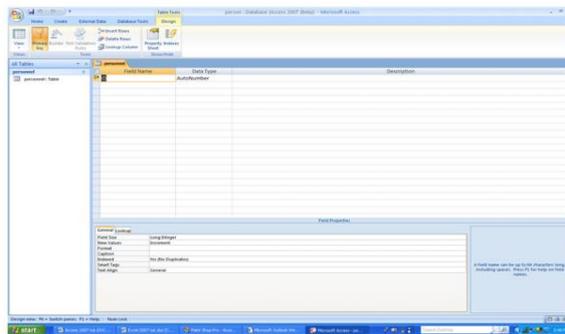
Gambar 2.28. Kotak dialog untuk menyimpan tabel⁴²

Sumber : Dokumentasi

⁴¹ Ibid, hal 55

⁴² Ibid, hal 55

Pada perencanaan tabel dengan *Design View* terdapat jendela *Table Designer* yang terdiri atas dua bagian, yaitu bagian atas dan bawah seperti terlihat pada gambar 2.29. Bagian atas digunakan untuk menetapkan struktur tabel, yaitu nama *field*, tipe data dan deskripsi dari *field* yang bersangkutan. Bagian bawah digunakan untuk mengatur *field properties*.



Gambar 2.29. Jendela *Design View*⁴³
Sumber : Dokumentasi

Sebuah *field* pada tabel akan memiliki tipe data yang sama. Tipe data yang dikenal oleh *Access 2007* adalah :

- ***Text***

Data ini berupa huruf, angka, spasi, atau tanda baca. *Field* data teks dalam tabel dapat menampung data teks dengan panjang sampai 255 karakter.

- ***Memo***

Tipe data *Memo* pada dasarnya sama dengan tipe data teks, namun memiliki ukuran data yang jauh lebih besar, yaitu mencapai 65536 karakter.

⁴³ Ibid, hal 55

- ***Number***

Tipe data *number* adalah tipe data berupa bilangan numeris. Tipe data ini dapat digunakan untuk perhitungan matematis.

- ***Date / Time***

Tipe *Date / Time* digunakan untuk menampung data berupa tanggal atau waktu.

- ***Currency***

Tipe data *Currency* menampung bilangan numeris yang berhubungan dengan mata uang. Bilangan yang digunakan untuk ketelitian hingga 15 digit didepan koma dan 4 digit di belakang koma.

- ***Auto Number***

Data angka yang berurutan dan unik(tidak ada yang sama) yang secara otomatis diberikan oleh *Access* jika *Record* data baru ditambahkan kedalam tabel.

- ***Yes/No***

Tipe data *Yes / No* adalah tipe data yang hanya berisi salah satu dari nilai *Yes* atau *No*, *True* atau *False*, *On* atau *Off*. Data yang tersimpan untuk *Yes* sebenarnya adalah -1 dan untuk *No* adalah 0.

- ***OLE Object***

Tipe data *OLE Object* adalah object – object seperti Microsoft Excel spreadsheet, Microsoft Word document, gambar, *sound*, atau data *binary* lain yang dikaitkan ke dalam tabel Access.

- ***Hyperlink***

Tipe data Hyperlink adalah sebuah link ke alamat tertentu.

- ***Lookup Wizard***

Tipe data ini berguna untuk memudahkan dalam memasukan data ke dalam tabel. Dilakukan dengan hanya memilih data dari tabel lain atau daftar yang didefinisikan sendiri.

D. **Pengaturan *Field Properties***

Properties untuk masing-masing *field* yang diset dari panel bawah dari *Design View window*. *Field Size* digunakan untuk mengeset jumlah karakter yang dibutuhkan dalam teks atau nomor *field*. *Default field* ukuran untuk teks adalah 50 karakter. Jika arsip dalam *field* hanya memiliki dua atau tiga karakter, anda bisa mengubah ukuran *field* untuk menghemat ruang disk atau or mencegah masukkan *error* dengan batas dari angka karakter yang diijinkan. Demikian juga, jika *field* membutuhkan lebih dari 50 karakter, masukkan angka lebih dari 255. Ukuran *field* diset dalam karakter angka untuk teks ketikan, tapi pilihan diberikan untuk jumlah :

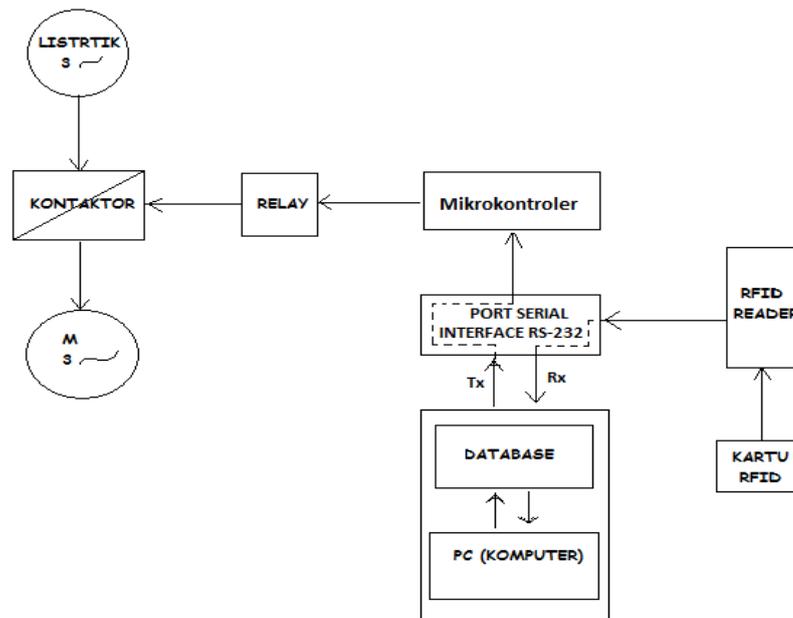
1. ***Byte - Positive integers*** antara 1 dan 255

2. **Integer** - *Positive dan negative integers* antara -32,768 dan 32,768
3. **Long Integer (default)** - *Larger positive and negative integers between -2 billion and 2 billion.*
4. **Single** - *Single-precision floating-point number*
5. **Double** - *Double-precision floating-point number*
6. **Decimal** - *Allows for Precision and Scale property control*
7. **Format** menyesuaikan data dalam *field* untuk format sama ketika dimasukkan ke dalam *datasheet*. Untuk teks dan memo fields, ini memiliki dua bagian yang dipisahkan dengan titik koma. Bagian pertama *property* digunakan untuk digunakan dalam *field* dan yang kedua untuk *field* kosong.

2.2. Kerangka Berpikir

Penelitian dalam pembuatan *prototype* absensi perkuliahan dan sistem peminjaman peralatan praktikum menggunakan RFID didasari pada penggunaan RFID sebagai alat untuk mengidentifikasi seseorang (mahasiswa) dengan menggunakan frekuensi transmisi radio yang lebih handal, praktis dan aman digunakan. Dengan menggunakan RFID ini penulis dapat membuat program absensi perkuliahan dan sistem peminjaman peralatan praktikum guna mengefesiensi waktu yang terbuang pada saat dosen mendata kehadiran mahasiswa dan saat mahasiswa melakukan peminjaman peralatan perkuliahan praktikum berlangsung.

Sistem alat ini menggunakan konsep dari cara kerja RFID sebagai absensi perkuliahan. Adapun sistem kerja alat dapat dilihat pada gambar 2.30.



Gambar 2.30. Blok Diagram Kerja Alat

Sumber : Dokumentasi

Penjelasan langkah-langkah untuk gambar diagram blok alat diatas yaitu : langkah awal melakukan absensi dengan tap kartu RFID mahasiswa ke RFID *reader*, maka RFID reader akan membaca *serial number tag* tersebut. Proses pembacaan *reader* terhadap *tag* ini dilakukan secara *wireless* dengan menggunakan frekuensi radio. Jenis RFID *reader* yang digunakan ialah *reader* tipe ID-12 dengan frekuensi kerja 125 kHz dengan kemampuan jarak baca yang tidak jauh. Kemudian *serial number* oleh RFID *reader* dikirimkan ke komputer secara serial dengan antarmuka serial-usb menggunakan RS232BM. Pada komputer, data yang berasal dari RFID *reader* tersebut dibandingkan dengan data yang ada pada *database* dengan menggunakan *borland delphi* dan pemrograman *database Microsoft Access 2007*. Data-data yang ada pada *database* mewakili identitas pengguna berupa nama mahasiswa dan nomor registrasi mahasiswa serta foto mahasiswa.

Jika data pada kartu RFID yang dimiliki pengguna (mahasiswa) ada pada *database* komputer, maka komputer akan menampilkannya sesuai dengan data yang telah teregistrasi dengan sebenar-benarnya secara otomatis, kemudian jika mahasiswa akan meminjam peralatan besar (bor duduk, gerinda duduk) yang sifatnya permanen maka petugas laboratorium cukup melakukan klik pada komputer yang telah berisi *icon* pilihan yang tersedia pada monitor komputer yang sudah terprogram didalamnya selanjutnya komputer akan mengirimkan instruksi kepada mikrokontroler

untuk menghidupkan relay (ON) selanjutnya menghidupkan kontaktor (ON) kemudian peralatan besar (sifatnya permanen) itu bisa digunakan. Komputer akan me-*record* data tersebut dan disimpan di *database*. Untuk kelompok praktikum selanjutnya juga seperti itu.

Namun jika data pada kartu RFID yang dimiliki mahasiswa tidak terdapat pada database komputer, maka komputer tidak akan me-*record* proses kehadiran mahasiswa dan komputer akan menampilkan notifikasi bahwa kartu tidak terdaftar dalam database komputer. Selanjutnya pengguna kartu RFID tersebut harus melakukan registrasi terlebih dahulu.

Data-data yang ditampilkan berasal dari komputer dan dikontrol oleh mikrokontroler AT89S52. Instruksi yang dikirimkan kepada mikrokontroler oleh komputer dikirimkan secara serial dengan antarmuka *serial-usb* menggunakan RS232BM. Bahasa pemrograman yang digunakan untuk mikrokontroler ialah bahasa pemrograman assembler.

Setelah perancangan dilakukan, maka langkah selanjutnya adalah melakukan pembuatan alat. *Prototype* ini dibuat berdasarkan perancangan sebelumnya dengan diawali menyiapkan alat dan komponen yang dibutuhkan lalu membuatnya sesuai dengan sketsa dan tujuan yang telah dibuat pada proses perancangan.

Alat dikatakan telah siap apabila telah berfungsi sesuai tujuan pembuatannya dan ketika diuji alat tersebut dapat bekerja seperti yang dikehendaki. Kriteria pengujian yaitu menguji apakah RFID dapat dapat bekerja sebagaimana mestinya berfungsi mengidentifikasi seseorang

(mahasiswa) dan program (delphi7) yang mampu mengintruksikan mikrokontroler untuk menghidupkan relai yang kemudian diteruskan menghidupkan kontaktor untuk peralatan listrik 3 phasa.