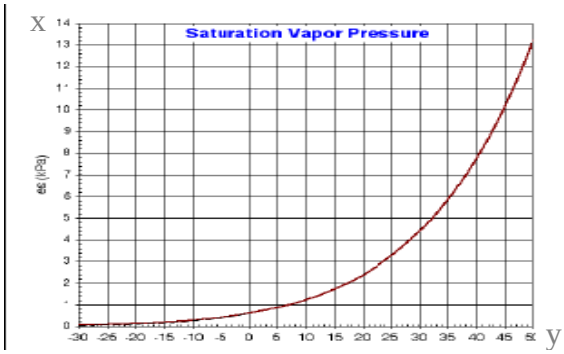
BAB II
LANDASAN TEORI

2.1 Kelembaban

Kelembaban merupakan suatu tingkat keadaan lingkungan udara basah yang disebabkan oleh adanya uap air. Tingkat kejenuhan sangat dipengaruhi oleh temperatur. Grafik tingkat kejenuhan tekanan uap air terhadap temperatur diperlihatkan pada Gambar 2.1.



Gambar 2.1. Grafik tingkat kejenuhan tekanan uap air terhadap temperatur

Jika tekanan uap parsial sama dengan tekanan uap air yang jenuh maka akan terjadi pemadatan. Secara matematis kelembaban relative (RH) didefinisikan sebagai prosentase perbandingan antara tekanan uap air parsial dengan tekanan uap air jenuh.

Kelembaban dapat diartikan dalam beberapa cara. *Relative Humidity* secara umum mampu mewakili pengertian kelembaban. Untuk mengerti *Relative Humidity* pertama harus diketahui *Absolut Humidity*. *Absolut Humidity* merupakan jumlah uap air pada volume udara tertentu yang dipengaruhi oleh temperatur dan tekanan.

$$a_h = 217 \left(\frac{e}{T} \right)$$

a_h : *absolute humidity*

e : tekanan oleh uap air

T : temperatur saat pengukuran

Relative Humidity merupakan persentase *rasio* dari jumlah uap air yang terkandung dalam volume tersebut dibandingkan dengan jumlah uap air maksimal yang dapat terkandung dalam volume tersebut (terjadi bila mengalami saturasi). *Relative Humidity* juga merupakan persentase *rasio* dari tekanan uap air saat dilakukan pengukuran dan tekanan uap air saat mengalami saturasi.

$$f = 100\left(\frac{a_h}{a_s}\right) = 100\left(\frac{e_h}{e_s}\right)$$

f : *relative humidity*

a_h : *absolute humidity* saat pengukuran

a_s : *absolute humidity* saat saturasi

e_h : tekanan uap air saat pengukuran

e_s : tekanan uap air saat saturasi

Pembacaan 100 %RH berarti udara telah saturasi (udara penuh dengan uap air). Berkeringat merupakan upaya tubuh untuk menjaga temperatur tubuh. Saat 100 %RH, keringat tidak menguap ke udara, sehingga tubuh terasa lebih panas. Sebaliknya bila RH rendah, maka tubuh akan merasa lebih dingin. Contoh: Saat temperatur udara 24 ^{o}C dan kelembaban 0%RH maka tubuh akan merasa temperatur udara seperti 21 ^{o}C, tetapi bila temperatur udara 24 ^{o}C dan kelembaban 100 %RH maka tubuh merasa temperatur udara seperti 27 ^{o}C. Biasanya besarnya RH yang dianggap nyaman sekitar 45 %RH.

2.2 Definisi Suhu

Suhu udara adalah ukuran energi kinetik rata – rata dari pergerakan molekul – molekul. Suhu suatu benda ialah keadaan yang menentukan kemampuan benda tersebut, untuk memindahkan (transfer) panas ke benda – benda lain atau menerima panas dari benda – benda lain tersebut. Dalam sistem dua benda, benda yang kehilangan panas dikatakan benda yang bersuhu lebih tinggi.

Selama 24 jam, suhu udara selalu mengalami perubahan-perubahan. Di atas lautan perubahan suhu berlangsung lebih banyak perlahan-lahan dari pada di atas daratan. Variasi suhu pada permukaan laut

kurang dari 1°C, dan dalam keadaan tenang variasi suhu udara dekat laut hampir sama. Sebaliknya diatas daerah pedalaman continental dan padang pasir perubahan suhu udara permukaan antara siang dan malam mencapai 20°C. Sedangkan pada daerah pantai variasinya tergantung dari arah angin yang bertiup. Variasinya besar bila angin bertiup dari atas daratan dan sebaliknya.

Suhu pada umumnya diartikan sebagai besaran yang menyatakan derajat panas dinginnya suatu benda. Skala suhu yang biasa digunakan diantaranya :

1. Celcius

   Skala Celsius adalah suatu skala suhu yang mendapatkan namanya dari ahli astronomi Anders Celsius (1701–1744), yang pertama kali mengusulkannya pada tahun 1742. Skala suhu Celsius didesain supaya titik beku air berada pada 0 derajat dan titik didih pada 100 derajat di tekanan atmosferik standar. Karena ada seratus tahapan antara kedua titik referensi ini, istilah asli untuk sistem ini adalah centigrade (100 bagian) atau centesimal. Pada 1948 nama sistem ini diganti secara resmi menjadi Celsius oleh Konferensi Umum tentang Berat dan Ukuran ke-9 (CR 64), sebagai bentuk penghargaan bagi Celsius dan untuk mencegah kerancuan yang timbul akibat konflik penggunaan awalan centi- (di Indonesia senti-) seperti yang digunakan satuan ukur SI. Meski angka-angka untuk saat beku dan mendidih untuk air tetap lumayan tepat, definisi aslinya tidak cocok digunakan sebagai standar formal: ia bergantung pada definisi tekanan atmosferik standar yang sendiri bergantung kepada definisi suhu. Definisi resmi Celsius saat ini menyatakan bahwa 0,01 °C berada pada triple point air dan satu derajat adalah 1/273,16 dari perbedaan suhu antara triple point air dan nol absolut.

   Definisi ini memastikan bahwa satu derajat Celsius merepresentasikan perbedaan suhu yang sama dengan satu kelvin. Anders Celsius awalnya mengusulkan titik beku berada pada 100

derajat dan titik didih pada 0 derajat. Suhu sebesar -40 derajat mempunyai nilai yang sama untuk Celsius dan Fahrenheit. Selain itu, sebuah cara untuk mengkonversi Celsius ke Fahrenheit adalah dengan menambah 40, dikalikan dengan 1,8, dan kemudian dikurangi 40. Sebaliknya, untuk mengkonversi dari Fahrenheit ke Celsius kita menambah 40, kemudian dibagikan 1,8 dan akhirnya dikurangi 40.

2. Fahrenheit

Skala Fahreheit adalah salah satu skala suhu selain Celsius dan Kelvin. Nama Fahrenheit diambil dari ilmuwan Jerman yang bernama Gabriel Fahrenheit (1686-1736). Skala ini dikemukakan pada tahun 1724. Dalam skala ini, titik beku air adalah 32 derajat Fahrenheit (ditulis 32°F) dan titik didih air adalah 212 derajat Fahrenheit. Negatif 40 derajat Fahreheit sama dengan negatif 40 derajat Celsius. Skala Fahrenheit banyak digunakan di Amerika Serikat.

3. Kelvin

Skala Kelvin (simbol: K) adalah skala suhu di mana nol absolut didefinisikan sebagai 0 K. Satuan untuk skala Kelvin adalah kelvin (lambang K), dan merupakan salah satu dari tujuh unit dasar SI. Satuan kelvin didefinisikan oleh dua fakta: nol kelvin adalah nol absolut (ketika gerakan molekuler berhenti), dan satu kelvin adalah pecahan 1/273,16 dari suhu termodinamik triple point air (0,01 °C). Skala suhu Celsius kini didefinisikan berdasarkan kelvin. Kelvin dinamakan berdasarkan seorang fisikawan dan insinyur Inggris, William Thomson, 1st Baron Kelvin.

Perkataan kelvin sebagai unit SI ditulis dengan huruf kecil k (kecuali pada awal kalimat), dan tidak pernah diikuti dengan kata derajat, atau simbol °, berbeda dengan Fahrenheit dan Celsius. Ini karena kedua skala yang disebut terakhir adalah skala ukuran sementara kelvin adalah unit ukuran. Ketika kelvin diperkenalkan pada tahun 1954 (di Konferensi Umum tentang Berat dan Ukuran

(CGPM) ke-10, Resolusi 3, CR 79), namanya adalah "derajat kelvin" dan ditulis °K; kata "derajat" dibuang pada 1967 (CPGM ke-13, Resolusi 3, CR 104). Perhatikan bahwa simbol unit kelvin selalu menggunakan huruf besar K dan tidak pernah dimiringkan. Tidak seperti skala suhu yang menggunakan simbol derajat, selalu ada spasi di antara angka dan huruf K-nya, sama seperti unit SI lainnya.

2.3 Definisi Curah Hujan

Hujan merupakan satu bentuk presipitasi yang berwujud cairan. Presipitasi sendiri dapat berwujud padat (misalnya salju dan hujan es) atau aerosol (seperti embun dan kabut). Hujan terbentuk apabila titik air yang terpisah jatuh ke bumi dari awan. Tidak semua air hujan sampai ke permukaan bumi karena sebagian menguap ketika jatuh melalui udara kering. Hujan jenis ini disebut sebagai virga.

Hujan memainkan peranan penting dalam siklus hidrologi. Lembaban dari laut menguap, berubah menjadi awan, terkumpul menjadi awan mendung, lalu turun kembali ke bumi, dan akhirnya kembali ke laut melalui sungai dan anak sungai untuk mengulangi daur ulang itu semula.

Intensitas curah hujan adalah jumlah curah hujan yang dinyatakan dalam tinggi hujan atau volume hujan tiap satuan waktu, yang terjadi pada satu kurun waktu air hujan terkonsentrasi. Besarnya intensitas curah hujan berbeda-beda tergantung dari lamanya curah hujan dan frekuensi kejadiannya. Intensitas curah hujan yang tinggi pada umumnya berlangsung dengan durasi pendek dan meliputi daerah yang tidak luas. Hujan yang meliputi daerah luas, jarang sekali dengan intensitas tinggi, tetapi dapat berlangsung dengan durasi cukup panjang. Kombinasi dari intensitas hujan yang tinggi dengan durasi panjang jarang terjadi, tetapi apabila terjadi berarti sejumlah besar volume air bagaikan ditumpahkan dari langit. Adapun jenis-jenis hujan berdasarkan besarnya curah hujan (definisi BMKG), diantaranya yaitu hujan kecil antara 0 – 21 mm per hari, hujan sedang antara 21 – 50 mm per hari dan hujan besar atau lebat di atas 50 mm per hari.

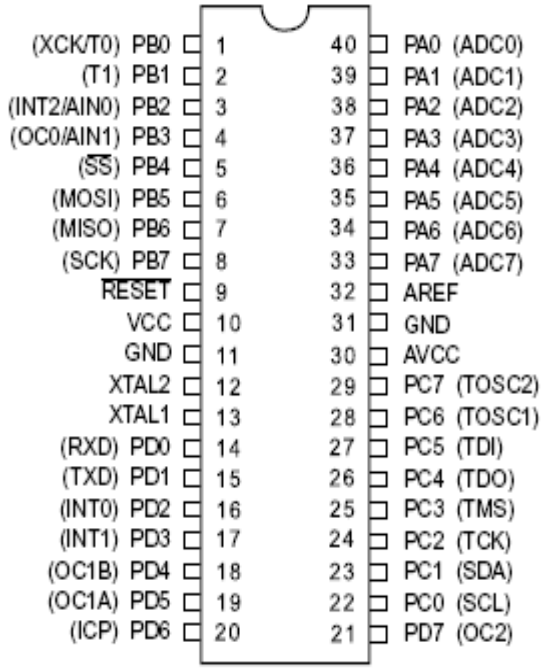
2.4 Mikrokontroler ATMega32

Mikrokontroler ATMega32 merupakan bagian dari keluarga mikrokontroller CMOS 8-bit buatan Atmel. ATMega32 memiliki arsitektur 8-bit, dimana semua instruksi dikemas dalam kode 16-bit dan sebagian besar instruksi dieksekusi dalam 1siklus clock. Mikrokontroler ATMega32 memiliki arsitektur *Havard,* yaitu memisahkan memori untuk kode program dan memori data. ATMega32 berteknologi RISC (*Reduced Instruction Set Computing*), sedangkan seri MCS51 berteknologi CISC (*Complex Instruction Set Computing)*. AVR dapat dikelompokkan menjadi empat kelas, yaitu keluarga ATtiny, keluarga AT 90Sxx, keluarga ATmega dan AT86RFxx. Pada dasarnya yang membedakan masing-masing kelas adalah memori, peripheral, dan fungsinya.

Mikrokontroler AVR ATmega32 adalah salah satu dari keluarga ATmega dengan populasi pengguna cukup besar. Memiliki memori flash 32k dan 32 jalur input output, serta dilengkapi dengan ADC 8 kanal dengan resolusi 10-bit dan 4 kanal PWM. Sebuah chip dengan fitur cukup lengkap untuk mendukung beragam aplikasi, termasuk robotik.

2.4.1 Deskripsi Pin

Mikrokontroler ATMega32 memiliki 40 kaki dan 32 kaki diantaranya merupakan port paralel yang terdiri dari port PA, PB, PC dan PD yang masing-masing memiliki 8 port dapat dilihat pada gambar 2.2.



Gambar 2.2 Deskripsi Pin Mikrokontroler ATMega32

Adapun nama dan fungsi dari setiap pin pada mikrokontroler ATMega32:

a. Port PA (pin 33 sampai dengan pin 40)

Port A berfungsi sebagai input analog ke A / D Converter. Port A juga berfungsi sebagai-directional I-bit bi 8 / O port, jika A / D Converter tidak digunakan. Port pin dapat menyediakan internal pull-up resistor (dipilih untuk setiap bit). Port A buffer output memiliki simbol berirama karakteristik drive dengan kedua tenggelam tinggi dan kemampuan sumber. Ketika pin PA0 untuk PA7 digunakan sebagai masukan dan secara eksternal ditarik rendah, mereka akan sumber arus jika internal pull-up Resistor diaktifkan. Port A pin yang tri-lain ketika kondisi reset menjadi aktif, bahkan jika jam tidak berjalan.

.

b. Port PB (pin 1 sampai dengan pin 8)

Port P1 berfungsi sebagai I/O biasa atau menerima *low order address bytes* pada saat *flash programming*. Port ini mempunyai internal pull up dan berfungsi sebagai input dengan memberikan output sink keempat buah input TTL.

c. Port PC (pin 22 sampai dengan pin 29)

Port PC berfungsi sebagai I/O biasa atau high order address, pada saat mengakses memori secara 16 bit (Movx @DPTR). Pada saat mengakses memori secara delapan bit, (Mov @Rn), port ini akan mengeluarkan isi dari PB *Special Function Register* (SFR). Port ini mempunyai internal pull up dan berfugsi sebagai input dengan memberikan logika 1. Sebagai output, port ini dapat memberikan output sink keempat buah input TTL.

d. Port PD (pin 14 sampai dengan pin 21)

Sebagai I/O biasa, Port P3 mempunyai sifat yang sama dengan port P1 dan port P2, sedangkan fungsi spesial dari port-port P2 ditunjukan pada tabel 2.1.

Tabel 2.1 Fungsi Port-port ATMega32

Pin	Port	Nama	Fungsi
Pin 14	PD.0	RXD	Port serial input
Pin 15	PD.1	TXD	Port serial output
Pin 16	PD.2	INT0	Port External Interupt 0
Pin 17	PD.3	INT1	Port External Interupt 1
Pin 11	PB.0	T0	Port external timer 0 input
Pin 12	PB.1	T1	Port external timer 1 input

e. Pin 9

Pin reset (RST) akan aktif dengan memberikan input high selama 2 *cycle.*

f. Pin 13

Pin 18 atau Pin XTAL1 untuk output oscillator.

g. Pin 12

Pin 19 atau Pin XTAL2 untuk input oscillator.

h. Pin 31

Pin 31 berfungsi sebagai ground dari mikrokontroler ATMega32.

i. Pin 10

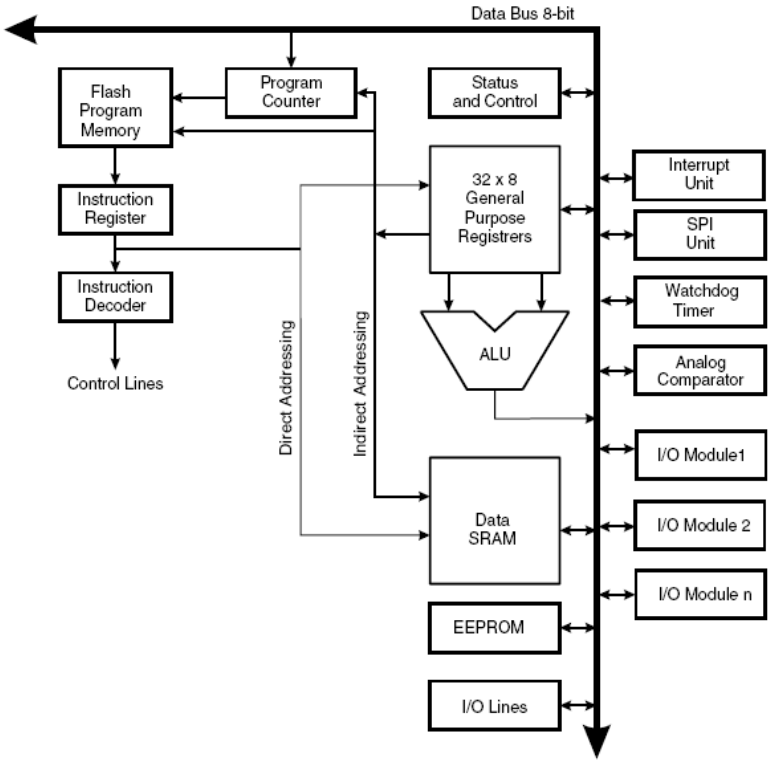
Pin 40 berfungsi sebagai VCC pada mikrokontroler ATMega32.

2.4.2 Arsitektur CPU ATMEGA32

Fungsi utama CPU adalah memastikan pengeksekusian instruksi dilakukan dengan benar. Oleh karena itu CPU harus dapat mengakses memori, melakukan kalkulasi, mengontrol peripheral, dan menangani interupsi.

Ada 32 buah General Purpose Register yang membantu ALU bekerja. Untuk operasi aritmatika dan logika, operand berasal dari dua buah general register dan hasil operasi ditulis kembali ke register. Status and Control berfungsi untuk menyimpan instruksi aritmatika yang baru saja dieksekusi. Informasi ini berguna untuk mengubah alur program saat mengeksekusi operasi kondisional. Instruksi di jemput dari flash memory. Setiap byte flash memory memiliki alamat masing-masing. Alamat instruksi yang akan dieksekusi senantiasa disimpan Program Counter. Ketika terjadi

interupsi atau pemanggilan rutin biasa, alamat di Program Counter disimpan terlebih dahulu di stack. Alamat interupsi atau rutin kemudian ditulis ke Program Counter, instruksi kemudian dijemput dan dieksekusi. Ketika CPU telah selesai mengeksekusi rutin interupsi atau rutin biasa, alamat yang ada di stack dibaca dan ditulis kembali ke Program Counter seperti pada gambar 2.3.



Gambar 2.3. Block Diagram of the AVR MCU Architecture

1. **Program Memori**

   ATMEGA32 memiliki 32 KiloByte flash memori untuk menyimpan program.Karena lebar intruksi 16 bit atau 32 bit maka flash memori dibuat berukuran 16K x 16. Artinya ada 16K alamat di flash memori yang bisa dipakai dimulai dari alamat 0 heksa sampai alamat 3FFF heksa dan setiap alamatnya menyimpan 16 bit instruksi.

2. **SRAM Data Memori**

ATMEGA32 memiliki 2 KiloByte SRAM. Memori ini dipakai untuk menyimpan variabel. Tempat khusus di SRAM yang senantiasa ditunjuk register SP disebut stack. Stack berfungsi untuk menyimpan nilai yang dipush.

3. **EEPROM Data Memori**

ATMEGA32 memiliki 1024 byte data EEPROM. Data di EEPROM tidak akan hilang walaupun catuan daya ke sistem mati. Parameter sistem yang penting disimpan di EEPROM. Saat sistem pertama kali menyala paramater tersebut dibaca dan system diinisialisasi sesuai dengan nilai parameter tersebut.

4. **Interupsi**

Sumber interupsi ATMEGA32 ada 21 buah. Tabel 2.1 hanya menunjukkan 10 buah interupsi pertama. Saat interupsi diaktifkan dan interupsi terjadi maka CPU menunda instruksi sekarang dan melompat ke alamat rutin interupsi yang terjadi. Setelah selesai mengeksekusi intruksi-instruksi yang ada di alamat rutin interupsi CPU kembali melanjutkan instruksi yang sempat tertunda.

5. **I/O Port**

ATMEGA32 memiliki 32 buah pin I/O. Melalui pin I/O inilah ATMEGA32 berinteraksi dengan sistem lain. Masing-masing pin I/O dapat dikonfigurasi tanpa mempengaruhi fungsi pin I/O yang lain. Setiap pin I/O memiliki tiga register yakni: DDxn, PORTxn, dan PINxn. Kombinasi nilai DDxn dan PORTxn menentukan arah pin I/O.

6. **Clear Timer on Compare Match (CTC)**

CTC adalah salah satu mode Timer/Counter1, selain itu ada Normal mode, FastPWM mode, Phase Correct PWM mode. Pada CTC mode maka nilai TCNT1 menjadi nol jika nilai

TCNT1 telah sama dengan OCR1A atau ICR1. Jika nilai *top* ditentukan OCR1A dan interupsi diaktifkan untuk Compare Match A maka saat nilai TCNT1 sama dengan nilai OCR1A interupsi terjadi. CPU melayani interupsi ini dan nilai TCNT1 menjadi nol.

7. **USARTH**

Selain untuk general I/O, pin PD1 dan PD0 ATMEGA32 berfungsi untuk mengirim dan menerima bit secara serial. Pengubahan fungsi ini dibuat dengan mengubah nilai beberapa register serial. Untuk menekankan fungsi ini, pin PD1 disebut TxD dan pin PD0 disebut RxD.

2.5 Komponen Instrumentasi

Komponen instrumentasi yang digunakan untuk membangun sistem alat monitoring cuaca mencakup IC *regulator*, kapasitor, dioda, kristal.

2.5.1 LM78xx (IC *Regulator* Tegangan)

Seperti diperlihatkan gambar 2.4, IC ini mempunyai tiga kaki yang digunakan sebagai komponen pendukung dari Vcc untuk menghasilkan tegangan 5, 6, 8, 9, 10, 12, 15, 18, 24Volt. Simbol ‘xx’ pada gambar 2.3 menandakan besar tegangan yang dihasilkan seperti untuk menghasilkan tegangan keluaran 5Volt maka nilai untuk menandakan simbol ‘xx’ tersebut adalah 05, yang berarti IC yang digunakan adalah LM7805. IC *regulator* ini berfungsi untuk menstabilkan tegangan. Penerapan IC ini mengharuskan Vi > Vo. IC *regulator* yang digunakan yaitu LM7805 untuk menghasilkan tegangan keluaran 5Volt dan LM78015 untuk menghasilkan tegangan keluaran 15Volt.



Gambar 2.4. IC LM78xx

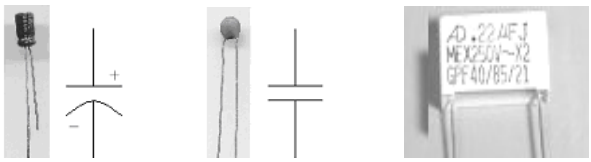
2.5.2 Kapasitor

Kapasitor disebut juga *Kondensator*, yaitu komponen yang berfungsi untuk menyimpan muatan/tegangan listrik atau menahan arus searah. Kapasitor ELCO (*Electrolit Capasitor*) terbuat dari keping aluminium dan elektrolit yang dikandung dalam lembaran kertas berpori. Plat aluminium bersifat sebagai isolator dan elektrolit berfungsi sebagai konduktor. Kapasitor ELCO memiliki kekutuban atau polaritas yaitu tanda positif dan tanda negatif. Jika dalam pemasangan kutub-kutub ELCO terbalik maka kapasitor akan rusak. Karena tidak terlalu akurat dan bersifat *elektronik marginal properties*, maka kapasitor jenis ini tidak baik digunakan dalam rangkaian yang berhubungan dengan transmisi sinyal HF. Jadi, kapasitor ELCO ini lebih baik digunakan untuk *filter ripple, timing circuit*.

Kapasitor keramik secara internal tidak dibangun sebagai koil, sehingga cocok untuk penggunaan aplikasi tinggi. Kapasitor ini bersifat non-polaritas atau tidak memiliki tanda positif dan tanda negatif sehingga dapat dipasang bolak-balik. Biasanya digunakan untuk *by-pass* sinyal frekuensi tinggi ke *ground*. Kapasitor keramik bergantung pada suhu lingkungan.

Metalized polyester capasitor dibuat dari *film dielectric* dan biasa disebut dengan Kapasitor Milar. Mempunyai kualitas yang baik, *low drift*, temperaturnya stabil. Secara fungsional, kapasitor milar ini sama dengan kapasitor non polaritas lain.

Untuk satuan dari ELCO adalah mikro, kapasitor keramik adalah piko dan kapasitor milar adalah nano. Simbol dan contoh dari kapasitor diperlihatkan oleh gambar 2.5.



(a) (b) (c)

Gambar 2.5. (a) ELCO, (b) Kapasitor Keramik dan (c) Kapasitor Milar

2.5.3 Dioda

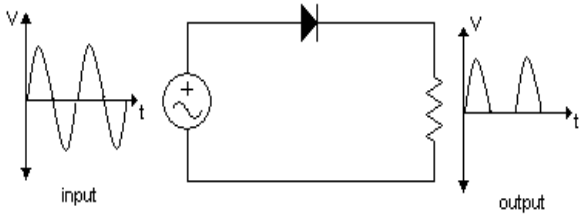
Dioda ialah suatu komponen semikonduktor yang memiliki sifat yang unik. Dioda hanya mengizinkan arus mengalir dalam satu arah saja, jika dipakai sebagai penyearah dengan kata lain dioda dapat mengubah sinyal ac menjadi sinyal dc.



(a) (b)

Gambar 2.6. Dioda (a).Fisik, (b).Simbol

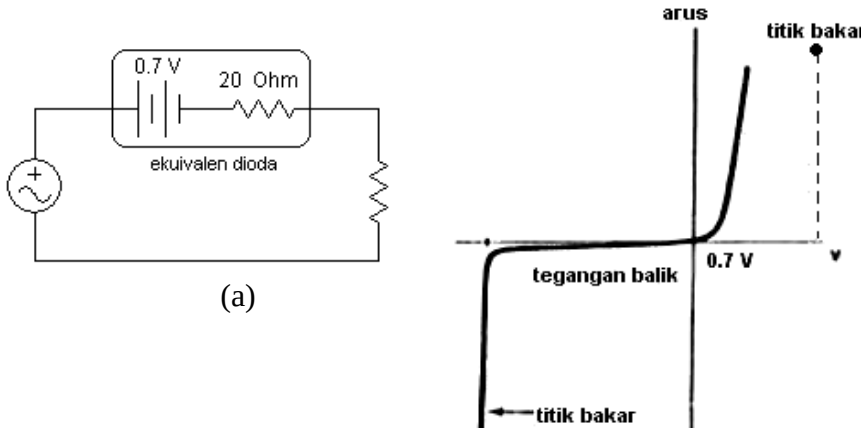
Gambar 2.6 menunjukkan sebuah dioda dan simbolnya. Bagian dioda terdiri dari 2 bagian yaitu anoda (positif) dan katoda (negatif). Seperti telah dijelaskan di atas, bahwa dioda hanya mengalirkan arus satu arah saja. ini berarti selama siklus negatif dari tegangan masukan, tidak akan ada arus yang melewati dioda. Seperti tampak pada gambar 2.7.



Gambar 2.7. Karakteristik Dioda

Hal-hal lain yang perlu diperhatikan ialah, pada saat tegangan sumber melewati dioda, terjadi penurunan tegangan sekitar 0.7 Volt. jadi bila tegangan input ialah 5 volt, tegangan keluarannya menjadi 4.3 Volt.

Selama tegangan input kurang dari 0.7 Volt, tidak akan ada arus yang dapat mengalir, dan setelah tegangan masukan melebihi 0.7 Volt, arus akan naik dengan cepat. Rangkaian *ekuivalen* dan *grafik* arusnya tampak seperti gambar 2.8.



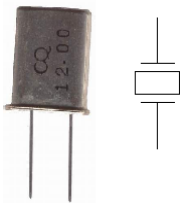
Gambar 2.8. Rangkaian Ekivalen Dioda dan Grafiknya

Sesaat setelah tegangan input melewati 0.7 Volt, arus akan mulai mengalir, tapi yang perlu diperhatikan ialah tegangan dan arus yang diberikan ke dioda tidak boleh terlalu tinggi karena akan menyebabkan kerusakan pada dioda atau terbakar, umumnya dioda dapat bertahan hingga 50 Volt dan arus 1 Ampere, tentunya tergantung pada jenis dioda yang kita pakai, itulah sebabnya mengapa dioda banyak digunakan sebagai penyearah tegangan.

2.5.4 Kristal

Kristal merupakan pembangkit *clock* internal yang menentukan rentetan kondisi-kondisi (*state*) yang membentuk

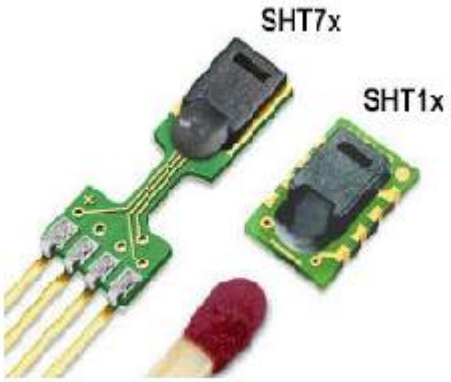
sebuah siklus mesin mikrokontroler. Siklus mesin tersebut diberi nomor S1 hingga S6, masing-masing kondisi panjangnya 2 *periode* osilator, dengan demikian satu siklus mesin paling lama dikerjakan dalam 12 *periode* osilator. Satuan kristal biasanya dalam skala mega yaitu antara 4MHz sampai 24MHz dengan bentuk dan simbol seperti yang diperlihatkan oleh gambar 2.9. Pada pembuatan sistem pemantauan cuaca ini menggunakan kristal dengan frekuensi 12MHz untuk mempermudah dalam perhitungan *timer*.



Gambar 2.9. Kristal

2.6 Sensor SHT1x/SHT1x

SHT1x/SHT1x merupakan multi sensor untuk kelembaban dan temperatur secara *digital.* Produk ini mulai dipasarkan Februari 2002 yang diproduksi oleh SENSIRION Company di Zurich (Switzerland) sensor dapat dilihat pada gambar 2.10. detailnya dijelaskan pada bab berikutnya.



Gambar 2.10. SHT11/71

Sensor tipe SHT11 / SHT7x merupakan multi sensor untuk suhu dan kelembaban secara digital. Sensor tipe SHT1x / SHT7x menggunakan

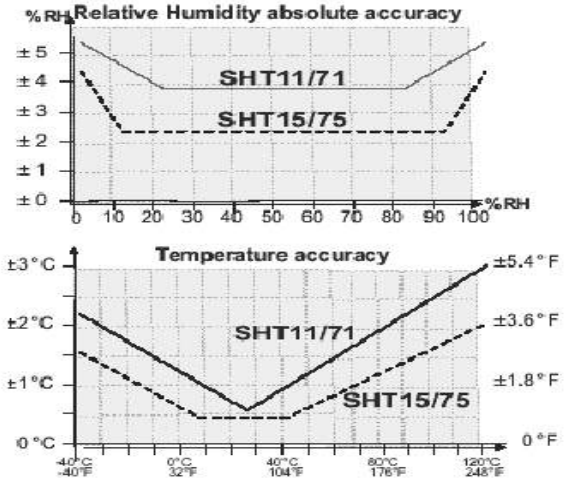
teknologi CMOS yang telah dipatenkan sehingga menjamin kestabilan dan *reliability* yang tinggi. Dalam chip ini terdiri dari *capacitive polymer sensing element* untuk *relative humidity* sensor dan bandgap temperatur sensor. Keduanya dihubungkan pada 14 bit ADC *(Analog to Digital Convertion*) dan *interface serial*, di dalam chip itu sendiri. Output yang dihasilkannya berupa kualitas sinyal yang superior, waktu respon yang cepat, tidak sensitif terhadap *external disturbace*, dan dengan harga yang kompetitif.

Antarmuka 2-wire *serial interface* dan *internal voltage regulation* membuat sistem integrasi yang mudah dan cepat. Juga karena bentuknya yang kecil dan konsumsi powernya yang hemat, sensor ini merupakan pilihan yang terbaik. Sensor ini tersedia dalam tipe bentuk yaitu *surface-mountable* LCC (*Lealess Chip Carrier*) dan *pluggable 4-pin single-in-line*.

Sensor tipe SHTxx dapat diaplikasikan dalam bermacam-macam bidang seperti: automotif, medis, *weather stations*, penyimpanan barang, HVAC (*Ventilation and air conditioning system*), *data logging*, alat ukur dapat dilihat pada gambar 2.10.

Sensor ini terdiri dari elemen *polimer kapasitif* (digunakan untuk mengukur kelembaban), sensor temperatur, 14 *bit* ADC (*Analog to Digital Converter*), dan *interface serial* 2 kabel. Didalamnya juga terdapat *memory* kalibrasi yang digunakan untuk menyimpan koefisien kalibrasi hasil pengukuran sensor. Data hasil pengukuran dari SHT11/71 ini berupa *digital logic* yang diakses secara *serial.*

Pada *microcontroller* data dikirim dan diterima pada PA.0 sedangkan PA.1 untuk *clock* yang akan dikirim ke sensor. Pada PA.0 dan PA.1 diberikan *pull-up* 10K untuk memastikan komunikasi agar *valid.* Hubungan *microcontroller* dengan SHT11.



Gambar 2.11. Grafik Akurasi Temperatur & Kelembaban

Pada gambar 2.11 di atas terlihat akurasi sensor SHT1x / SHT7x untuk pengukuran temperatur dan kelembaban.

Akurasi pengukuran temperatur:

Tipe SHT11 / SHT71:

Untuk -40^{o}C < T 28^{o}C *error* pengukuran ± 2.25^{o}C

Untuk 28^{o}C <T 123.8^{o}C *error* pengukuran ± 3^{o}C

Tipe SHT15 / SHT75:

Untuk -40^{o}C < T 8^{o}C *error* pengukuran ± 1. 5^{o}C

Untuk 28^{o}C < T 45^{o}C *error* pengukuran ± 0.5^{o}C

Untuk 45^{o}C< T 123.8^{o}C *error* pengukuran ± 2^{o}C

Akurasi pengukuran kelembaban:

Tipe SHT11 / SHT71:

Untuk 0%RH< H< 23%RH *error* pengukuran ± 5 %RH

Untuk 23%RH< H< 85%RH *error* pengukuran ± 4 %RH

Untuk 95%RH< H< 100%RH *error* pengukuran ± 5%RH

Tipe SHT15 / SHT75:

Untuk 0%RH< H< 13%RH *error* pengukuran ±3%RH

Untuk 13%RH< H< 95%RH *error* pengukuran ±2.5%RH

Untuk 95%RH< H< 100%RH *error* pengukuran ±3%RH

Di bawah ini tabel dari karakteristik SHT11 /SHT71

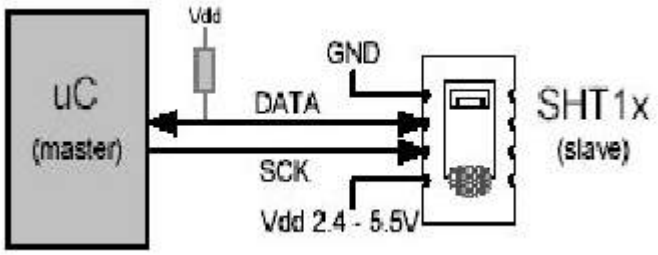
Tabel 2.2 Karakteristik Sensor SHT11/SHT71

Parameter	Kondisi	Min.	Typ.	Max.	Units.
Kelembaban					
Resolusi (2)		0.5	0.03	0.03	%RH
		8	12	12	hit
Repeatability			±0.1		%RH
Akurasi (1) Keraguan	linear	Lihat figure 1			
Interchangeabilitiy		Fullyn Interchangeabilitiy			
Nonlinearity	Raw data linearized		±3 <<1		%RH %RH
Range		0		100	%RH
Waktu Response	1/s(63%) slowly movirg air		4		s
Hysteresis			±1		%RH
Stabilitas	typical		<1		%RH
Temperatur					
Resolusi (2)		0.04	0.01	0.01	^{o}C
		0.07	0.02	0.02	^{o}F
		12	14	14	bit
Repeatability			±0.1		^{o}C
			±0.2		^{o}F
Akurasi		Lihat figure 1			
Range		-40		123.8	^{o}C
		-40		254.9	^{o}F
Waktu Response	1/s(63%)	5		30	s

Sumber : SHT1x / SHT7x Datasheet (telah ditolah kembali)

SHT11/71 mempunyai karakteristik antara lain : *range* kelembaban 0%RH – 100%RH, ketelitian ± 4%RH untuk range kerja 23%RH – 84%RH sedangkan untuk RH < 23% atau RH > 84% ketelitiannya ± 5%RH, *range* temperatur -40 ^{o}C – 123.8 ^{o}C (untuk jelasnya karakteristik dan akurasi kelembaban dan temperatur dapat dilihat pada gambar 2.12), waktu respon normalnya 4s, untuk jelasnya respon waktu dilihat pada tabel 2.3.

Spesifikasi Interface



Sumber : SHT1x/SHT7x Datasheet

Gambar 2.12. Aplikasi Sirkuit

Power Pin

SHTxx memerlukan *supply* antara 2,4V – 5,5V. Setiap kali *power-up, chip* ini memerlukan 11ms untuk mencapai keadaan stabil, sebelum keadaan ini tercapai tidak diperbolehkan adanya pengiriman "Command". *Power supply pin* (VCC dan GRD) disarankan di-*kople* dengan kapasitor sebesar 100nF.

Serial interface

Serial Interface dari SHTxx dioptimalkan untuk pembacaan sensor dan konsumsi *power*, dan tidak kompatibel dengan 12C *interface.*

- *Serial* Clock *input* (SCH)

  Digunakan untuk men-sinkronisasi komunikasi antara *microcontroller* dan SHTxx. Karena *interface* ini terdiri dari *static logic* sepenuhnya maka tidak ada batasan frekwensi minimum dari SCK.

- *Serial* Data (DATA)

  Data pin merupakan *tri-state pin* yang digunakan untuk transfer data *in* dan data *out.* DATA berubah setelah transisi turun, dan *valid* pada transisi naik dari *serial clock* SCK. Selama transmisi, DATA *line* harus stabil selama SCK *high.*

Untuk menghindari adanya *signal contetion, microcontroller* hanya diperbolehkan men-*drive* DATA dengan *low*. Eksternal *pull-up* resistor (10K) diperlukan untuk memastikan *logic high.*

2.6.1 Mengirim Command

Untuk memulai transmisi dikirimkan "Transmisi Start" yang berupa men-*drive low* DATA *line* ketika SCK *high*, diikuti pulsa *low* pada SCK dan men-*drive high* DATA *line* ketika SCK masih ber-*logic high* tampak pada gambar 2.13.



Sumber : SHT1x/SHT7x Datasheet

Gambar 2.13. Transmisi start

Selanjutnya bagian "Command" terdiri dari 3 *bit addres* (yang mendukung hanya 000) dan 5 *bit* "Command". SHTxx mengindikasikan penerimaan "Command" dengan men-*drive* DATA *low* (ACK *bit*) setelah transisi *low* ke-8 dari *clock* SCK. Kontrol DATA *line* dilepas (sehingga menjadi *high* dikarenakan *pull-up*)setelah transisi turun ke-9 *clock* SCK.

Tabel 2.3. Command SHTxx

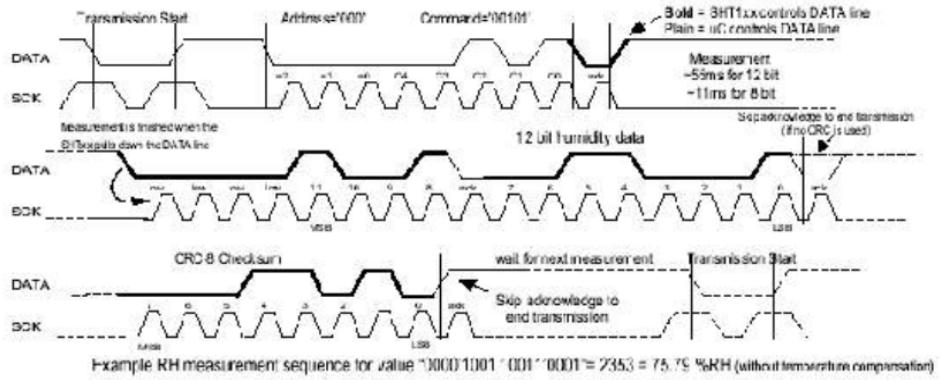
Command	Code
Reserved	0000x
Measure temperature	00011
Measure Humidity	00101
Read Status Register	00111
Write Status registerReserved	00110
Reserved	0101x-1110x
Soft reset, resets the interface, clears, clears the status register to default values wait minimum 11 ms before next command	11110

Sumber : SHT1x / SHT7x Datasheet

2.6.2 Pengukuran Sensor

Setelah mengirimkan “Command Measure”, *microcontroller* menunggu proses pengukuran selesai, ini memakan waktu kurang lebih 11/55/210 ms untuk pengukuran 8/12/14 *bit.* Waktu sesungguhnya bervariasi sampai ±15% kecepatan dari internal *oscillator.* Untuk menandakan bahwa pengukuran selesai, STHxx men-*drive low* DATA *line. Microcontroller* harus menunggu tanda ini sebelum menjalankan *clock* SCK lagi.

Kemudian 2 *byte* hasil pengukuran dan 1 *byte* CRC ditransmisikan, *microcontroller* harus memberi *signal acknowlidge* untuk tiap *byte* dengan men-*drive* DATA *line low.* Semua nilai *output* dimulai dengan MSB atau *right justified,* (misal: SCK ke-5 adalah MSB untuk *output 12 bit;* sedangkan untuk *output* 8 *bit, byte* pertama tidak digunakan). Komunikasi berhenti setelah *acknowledge bit* dari CRC *output.* Bila CRC tidak diperlukan, maka *mikrokontroller* dapat menghentikan komunikasi setelah *output* pengukuran LSB (dengan membiarkan *anknowladge high*). *Chip* otomatis masuk dalam *mode “sleep”* setelah pengukuran dan komunikasi berakhir. Beda waktu antar pengukuran sekitar 1 detik.

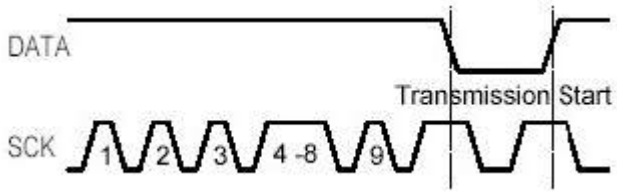


Sumber : SHT1x/SHT7x Datasheet

Gambar 2.14. Timing Diagram Pengukuran

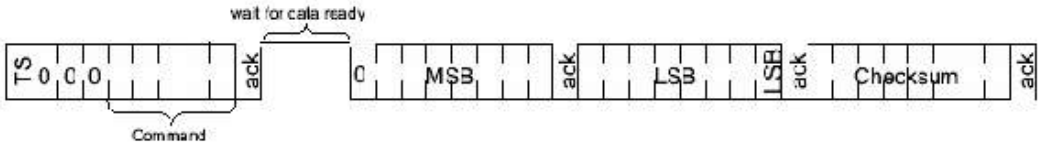
Bila komunikasi dengan *chip* hilang maka diperlukan *reset serial interface* (dengan menjalankan *clock*

SCK lebih dari 9 kali dengan menjaga DATA tetap *high* yang kemudian diikuti dengan "Transmision Start" dan "Command"). *Reset* ini tidak berpengaruh pada isi *status register* dapat dilihat pada gambar 2.15.



Sumber : SHT1x/SHT7x Datasheet

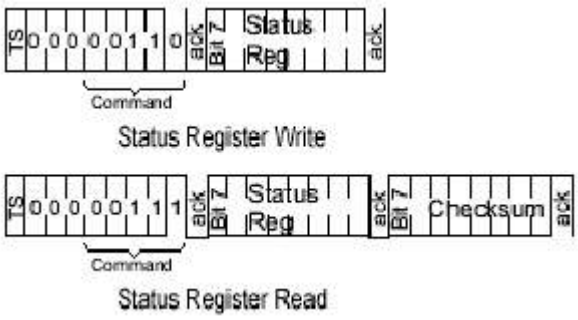
Gambar 2.15. Reset



Gambar 2.16. Skematik Pengukuran

Status Register

Beberapa fungsi dari SHTxx terdapat pada "Status Register", dibawah ini akan dideskripsikan lebih lanjut.



Sumber : SHT1x/SHT7x Datasheet

Gambar 2.17. Skematik Status Register

Tabel 2.4. Status Register

Bit	Type	Description	Defoult	
7		reserved	0	
6	R	End of Battery (low voltage detection) '0' for Void > 2.47 '1' for Void < 2.47	X	No defoult value bit is only updated after a measurement
5		Reserved	0	
4		Reserved	0	
3		For testing only do not use	0	
2	R/W	Heater	0	off
1	R/W	No reload from OTP	0	reload
0	R/W	'1' = 8 bit RH/11 bit Temperature resolution '1' = 8 bit RH/11 bit Temperature resolution	0	12 bit RH 14 bit temp

Sumber : SHT1x/SHT7x Datasheet

Resolusi Pengukuran

Defoult resolusi pengukuran adalah 14 *bit* (temperatur) dan 12 *bit* (RH). Resolusi ini dapat diubah menjadi 12 *bit* (temperatur) dan 8 *bit* (RH) untuk kegunaan transfer data kecepatan tinggi dan *low power application.*

End of Battery

Fungsi dari *End of Battery* untuk mendeteksi V_{DD} dibawah 2,47 V. Tingkat akurasi ±0,05V. *Bit* ini hanya di-*update* bila terjadi pengukuran.

Heater

Di dalam *chip* SHTxx terdapat elemen *heater* yang dapat dinyalakan. *Heater* ini bila dinyalakan akan meningkatkan temperatur dari sensor ±5 ^{o}C (9^{o}F), dan konsumsi *power* naik

~8mA @ 5V. Dengan membandingkan hasil pengukuran temperatur dan RH sebelum dan sesudah penggunaan *Heater* maka akan diketahui berfungsi tidaknya sensor tersebut.

Dalam lingkungan dengan kelembaban tinggi (RH>95%), penggunaan *Heater* akan menghambat terjadinya kondensasi, meningkatkan waktu respon dan tingkat akurasi. Bila SHTxx mengalami panas, pengukuran menunjukan hasil pengukuran temperatur yang lebih tinggi dan RH yang lebih rendah dibanding pengukuran pada kondisi normal.

Kalibrasi Reload Sebelum Pengukuran

Untuk menghemat *power* dan meningkatkan kecepatan pengukuran maka OTP *reload* tiap pengukuran dapat di-*bypass*, ini menghemat ~8,2ms tiap pengukuran.

2.6.3 Konversi Output SHTxx ke Nilai Fisik

Relative Humidity

Sensor kelembaban tidak dipengaruhi secara signifikan oleh besarnya *voltage.* Untuk kompensasi ke-tidak linear-an dari sensor kelembaban dan untuk memperoleh akurasi yang tinggi, maka untuk temperatur 25^{o}C disarankan menggunakan rumus sesuai tabel 2.5, sedangkan untuk temperatur selain 25^{o}C, konpensasi RH menggunakan rumus sesuai tabel 2.6.

Tabel 2.5 Koefesien Konversi Kelembaban

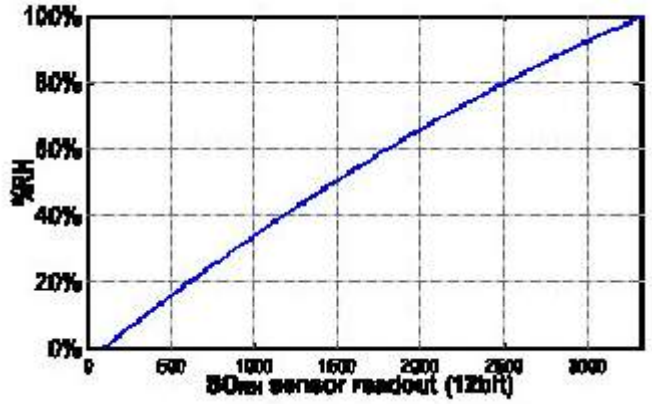
$$RH_{linear} = c_1 + c_2 \bullet SO_{RH} + c_3 \bullet SO_{RH}^{\ 2}$$

SO_{RH}	c_1	c_2	c_3
12 bit	-4	0.0405	$-2.8 * 10^{-6}$
8 bit	-4	0.648	$-7.2 * 10^{-4}$

Tabel 2.6 Koefisien Konversi oleh Temperatur

$$RH_{true} = (T_C - 25) \bullet (t_1 + t_2 \bullet SO_{RH}) + RH_{linear}$$

SO_{RH}	t_1	t_2
12 bit	0.01	0.00008
8 bit	0.01	0.00128



Gambar 2.18. Konversi Output Sensor Kelembaban Terhadap RH

Temperature

Sensor temperatur PPAT (Proportional To Absolute temperature) merupakan sensor yang linear, konversi *digital output* menggunakan rumus sesuai tabel 2.7.

Tabel 2.7. Koefesien Konversi Temperatur

$$Temperature = d_1 + d_2 \bullet SO_T$$

VDD	d_1 [°C]	d_1 [°f]
5V	-40.00	-40.00
4V	-39.75	-39.50
3.5V	-39.66	-39.35
3V	-39.60	-39.28
2.5V	-39.55	-39.23

SO_T	d_2 [°C]	d_2 [°f]
14bit	0.01	0.018
12bit	0.04	0.072

2.7 Sensor Curah Hujan Dengan Metode PUSH Bottom

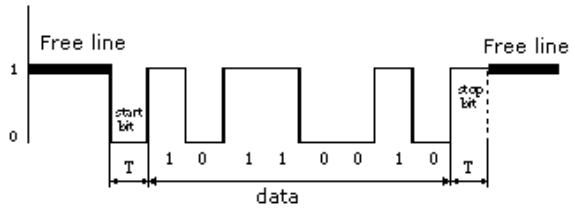
Sensor Curah Hujan Dengan Metode *PUSH Bottom* adalah sensor untuk melakukan pengukuran tingkat curah hujan dengan cara menangkar air hujan dengan cara menampung air hujan hingga memenuhi nilai tertentu (0,2 mm atau 0,5 mm). Bila air hujan telah memenuhi nilai

tersebut maka sensor akan terjungkit. Setiap kali terjungkit/tipping menyebabkan tertekanya *switch* sebagai signal output berupa pulsa.

2.8 Sistem komunikasi Serial

Port serial adalah port yang paling populer digunakan untuk keperluan koneksi ke piranti luar. Kata “Serial”, menggambarkan prinsip kerja *port* ini yang memberikan (*Serialize*) data. Cara kerjanya adalah diawali dengan mengambil sebuah byte data lalu kemudian mengirimkan perdelapan bit dalam byte tersebut satu persatu dalam satu jalur data. Keuntungannya adalah bahwa port ini hanya membutuhkan satu kabel untuk mengirimkan kedelapan bit tadi (dibandingkan port paralel yang membutuhkan delapan kabel). Keuntungan lainnya adlah efisiensi dalam biaya dan tentunya ukuran kabel yang kecil. Kerugiannya yakni bahwa port serial membutuhkan delapan kali lebih lama untuk mengirimkan data dibanding dengan proses pengiriman dengan delapan kabel.

Terdapat dua cara dalam komunikasi data secara serial, yaitu komunikasi data serial secara sinkron dan komunikasi data serial secara asinkron. Pada komunikasi data serial sinkron, clock dikirimkan bersama-sama dengan data serial, sedangkan pada komunikasi data asinkron clock tidak dikirimkan bersama data serial, tetapi dibangkitkan secara terpisah baik pada bagian pemancar maupun pada bagian penerima. Kecepatan pengiriman data dan fase clock pada bagian pemancar dan bagian penerima harus sinkron, untuk itu diperlukan sinkronisasi antara dua bagian tersebut. Salah satu caranya adalah dengan mengirimkan bit ‘start’ dan bit ‘stop’. Untuk bit ‘start’ adalah data biner 0 dan untuk bit ‘stop’ adalah data biner 1. Setelah pengiriman bit ‘start’ maka akan diikuti oleh data yang akan dikirim , selanjutnya diakhiri dengan bit ’stop’. Berikut adalah contoh pengiriman karakter B2 heksa atau 10110010 biner tanpa bit paritas. Dapat terlihat pengiriman data diawali dengan bit ‘ start’ lalu data B2 heksa dan diakhiri dengan bit ’stop’ sebagai akhir dari pengiriman dapat dilihat pada gambar 2.19.



Gambar 2.19. Pengiriman data serial

Kecepatan pengiriman data *(baud rate)* bervariasi, mulai dari 110,135,150,300,600,1200,2400,4800, 9600, 19200, 38400, 57600, 115200, 230400, 460800 dan 921600 (bit/detik). Pada komunikasi data serial *baut rate* dari kedua bagian harus diatur pada kecepatan yang sama. Setelah itu harus ditentukan panjang datanya, apakah 6,7 atau 8 bit., juga apakah data disertai dengan paritas genap, paritas ganjil atau tidak menggunakan paritas. Untuk menentukan *baud rate* dapat dilihat pada persamaan dibawah ini:

Misalkan XTAL yang digunakan adalah = 11.0592 MHz

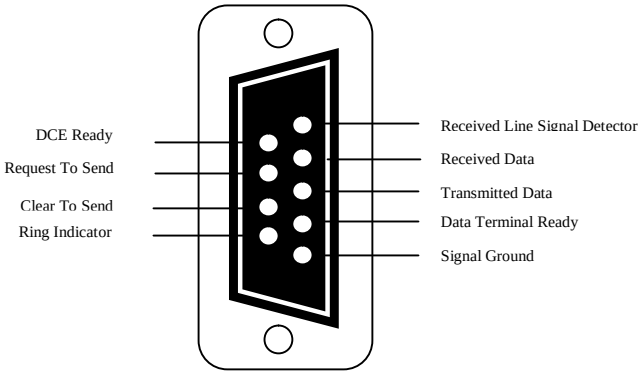
Maka:

Machine Cycle Frequency = $\frac{11.0592 \text{ MHz}}{(\textit{Machine Cycle})\ 12} = 921.6$ kHz

Machine Cycle Frequency Mode 1 = $\frac{921.6 \text{ kHz}}{(\text{Bit } \textit{counter} \text{ Mode 1})\ 16} = 57.600$ Hz

Baud rate = $\frac{57.600 \text{ Hz}}{3} = 19.200$ *Baud rate* (2^n = 8 bit data)

Konektor port serial atau yang biasa disebut DB-9 (COM1 dan COM2) dapat dilihat pada bagian belakang komputer (CPU) memiliki kaki sejumlah 9 pin seperti pada gambar 2.20.



Gambar 2.20. Konektor Serial DB-9

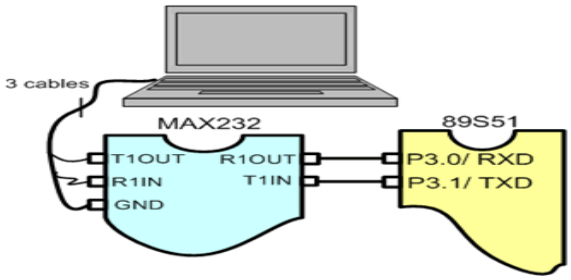
Tabel 2.8. Konfigurasi pin dan nama sinyal konektor serial DB-9

Nomor Pin	Nama Sinyal	Arah In/Out	Keterangan
1	DCD	In	*Data Carrier Detect/ Receive Line Signal Detect*
2	RxD	In	*Receive Data*
3	TxD	Out	*Transmit Data*
4	DTR	Out	*Data Terminal Ready*
5	GND	-	*Ground*
6	DSR	In	*Data Set Ready*
7	RTS	Out	*Request To Send*
8	CTS	In	*Clear To Send*
9	RI	In	*Ring Indikator*

2.8.1 MAX232

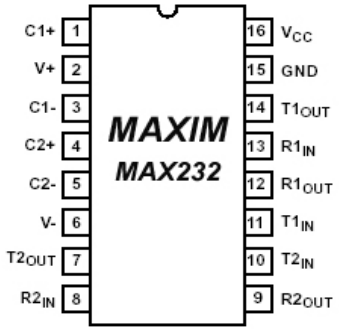
Untuk dapat berhubungan dengan PC, mikrokontroler harus membutuhkan komponen tambahan baik komunikasi paralel maupun serial. Pada pembuatan tugas akhir ini yang digunakan adalah komunikasi serial. Pada mikrokontroler sendiri terdapat *buffer* yang dapat digunakan sebagai pendukung proses komunikasi tersebut. Pada saat ini banyak komponen yang dapat digunakan untuk pendukung proses komunikasi tersebut, salah satu contohnya adalah maxim232.

Maxim232 berfungsi sebagai perantara antara mikrokontroler dengan port serial, karena mikrokontroler tidak dapat mengirim data begitu saja maka diperlukan maxim232. di dalam IC terdapat charge pump yang akan membangkitkan +10 Volt dan -10 Volt dari sumber +5 Volt tunggal dalam IC DIP (*Dual in-line Package*) 16 pin (8 pin x 2baris) ini terdapat 2 buah transmiter dan dua buah receiver. Jadi IC ini berfungsi sebagai perantara karena maxim232 hanya menerima data dari mikrokontroler untuk kemudian dikirim ke pc melalui DB9 terlihat pada gambar 2.21.



Gambar 2.21. Interface MAX232

Maxim232 mempunyai 16 kaki yang terdiri untuk keperluan port serial, komunikasi mikrokontroler dengan maxim. Letak dari masing-masing port diperlihatkan pada gambar 2.22.



Gambar 2.22. Konfigurasi pin MAXIM232

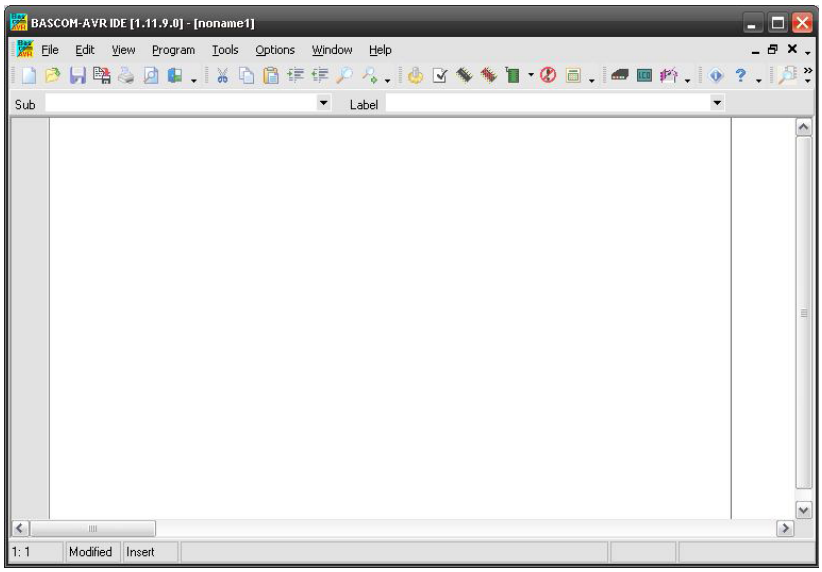
Adapun nama dan fungsi dari kaki-kaki pin pada Maxim232 adalah sebagai berikut:

1. VCC (pin 16) : Power supply
2. GND (pin 15) : Ground

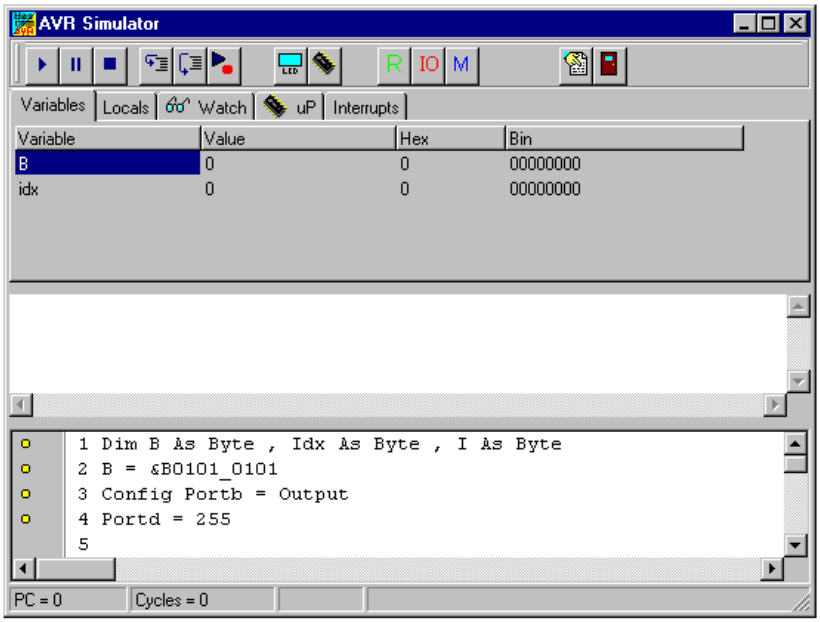
3. T_{1IN} dan R_{1OUT} (pin 11 dan 12) : Pin ini terhubung dengan pin 11 mikrokontroler ATMega32.
4. R_{1IN} dan T_{1OUT} (pin 13 dan 14) : Pin ini terhubung dengan pin 2 dan 3 DB9.
5. C_{1+} dan C_{1-} : Kapasitor 1
6. C_{2+} dan C_{2-} : Kapasitor 2
7. V_+ dan V_- : Tegangan referensi dari Maxim232.

2.9 AVR

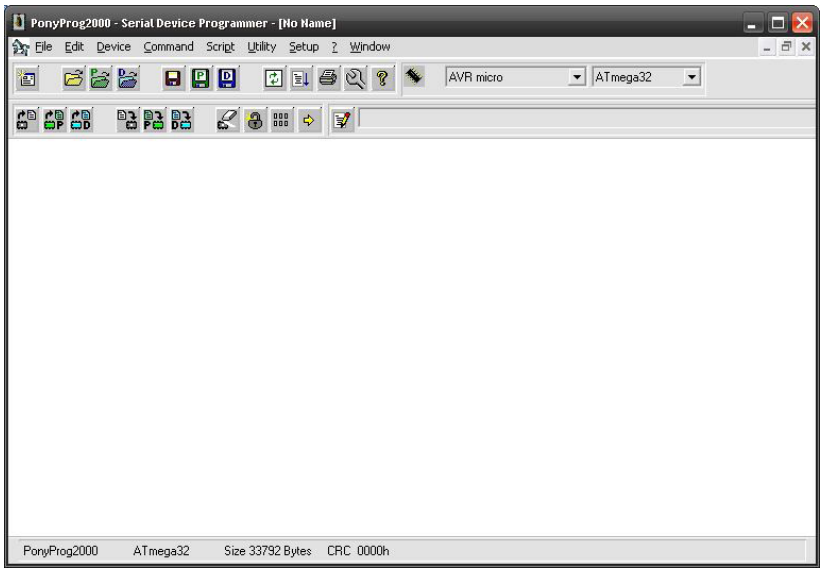
AVR merupakan pemrograman perangkat lunak pendukung mikrokontroler Atmega32 yang dilengkapi dengan simulator. Mulai dari penulisan sumber program yang menggunakan bahasa basic, sampai compiling program ke hex. Selanjutnya untuk mendownload file .hex ke flash memori Atmega32 digunakan software PonyProg2000. Tampilan software BASCOM AVR, Tampilan Simulator BASCOM AVR dan PonyProg2000 masing-masing ditunjukkan oleh Gambar 2.23, Gambar 2.24 dan Gambar 2.25.



Gambar 2.23. Tampilan Utama Bascom AVR



Gambar 2.24. Tampilan Simulator Bascom AVR



Gambar 2.25. Tampilan Ponyprog2000

Code AVR pada dasarnya merupakan perangkat lunak pemrograman microcontroller keluarga AVR berbasis bahasa C. Ada tiga komponen penting yang telah diintegrasikan dalam perangkat lunak ini: Compiler C, IDE dan Program generator.

Berdasarkan spesifikasi yang dikeluarkan oleh perusahaan pengembangnya, Compiler C yang digunakan hampir

mengimplementasikan semua komponen standar yang ada pada bahasa C standar ANSI (seperti struktur program, jenis tipe data, jenis operator, dan library fungsi standar-berikut penamaannya). Tetapi walaupun demikian, dibandingkan bahasa C untuk aplikasi komputer, compiler C untuk microcontroller ini memiliki sedikit perbedaan yang disesuaikan dengan arsitektur AVR tempat program C tersebut ditanamkan (embedded).

Khusus untuk library fungsi, disamping library standar (seperti fungsi-fungsi matematik, manipulasi String, pengaksesan memori dan sebagainya), Code AVR juga menyediakan fungsi-fungsi tambahan yang sangat bermanfaat dalam pemrograman antarmuka AVR dengan perangkat luar yang umum digunakan dalam aplikasi kontrol. Beberapa fungsi library yang penting diantaranya adalah fungsi-fungsi untuk pengaksesan LCD, komunikasi I^2C, IC RTC (*Real time Clock*), sensor suhu dan kelembaban SHT11, SPI (*Serial Peripheral Interface*) dan lain sebagainya. Untuk memudahkan pengembangan program aplikasi, Code AVR juga dilengkapi IDE yang sangat *user friendly*. Selain menu-menu pilihan yang umum dijumpai pada setiap perangkat lunak berbasis Windows, Code AVR ini telah mengintegrasikan perangkat lunak *downloader* (*in system programmer*) yang dapat digunakan untuk mentransfer kode mesin hasil kompilasi kedalam system memori microcontroller AVR yang sedang diprogram.

Selain itu, Code AVR juga menyediakan sebuah *tool* yang dinamakan dengan *Code Generator* atau Code AVR. Secara praktis, t*ool* ini sangat bermanfaat membentuk sebuah kerangka program (*template*), dan juga memberi kemudahan bagi programmer dalam peng-inisialisasian register-register yang terdapat pada microcontroller AVR yang sedang diprogram. Dinamakan Code Generator, karena perangkat lunak Code ini akan membangkitkan kode-kode program secara otomatis setelah fase inisialisasi pada jendela Code AVR selesai dilakukan. berikut memperlihatkan beberapa penggal baris kode program yang dibangkitkan secara otomatis oleh Code AVR. Secara teknis, penggunaan *tool* ini pada dasarnya hampir sama dengan *application wizard* pada bahasa-bahasa

pemrograman Visual untuk komputer (seperti Visual C, Borland Delphi, dan sebagainya).

2.10 DFD (*Data Flow Diagram*)

Data Flow Diagram (DFD) adalah alat pembuatan model yang memungkinkan profesional sistem untuk menggambarkan sistem sebagai suatu jaringan proses fungsional yang dihubungkan satu sama lain dengan alur data, baik secara manual maupun komputerisasi. DFD ini sering disebut juga dengan nama *Bubble chart, Bubble* diagram, model proses, diagram alur kerja, atau model fungsi.

DFD ini adalah salah satu alat pembuatan model yang sering digunakan, khususnya bila fungsi-fungsi sistem merupakan bagian yang lebih penting dan kompleks dari pada data yang dimanipulasi oleh sistem. Dengan kata lain, DFD adalah alat pembuatan model yang memberikan penekanan hanya pada fungsi sistem.

DFD ini merupakan alat perancangan sistem yang berorientasi pada alur data dengan konsep dekomposisi dapat digunakan untuk penggambaran analisa maupun rancangan sistem yang mudah dikomunikasikan oleh profesional sistem kepada pemakai maupun pembuat program. Adapun simbol-simbol dari DFD yang dapat dilihat pada lampiran.

Penggambaran DFD

Tidak ada aturan baku untuk menggambarkan DFD. Tapi dari berbagai referensi yang ada, secara garis besar langkah untuk membuat DFD adalah :

1. Identifikasi terlebih dahulu semua entitas luar yang terlibat di sistem.
2. Identifikasi semua input dan output yang terlibat dengan entitas luar.
3. Buat Diagram Konteks (*diagram context*)

Diagram ini adalah diagram level tertinggi dari DFD yang menggambarkan hubungan sistem dengan lingkungan luarnya. Caranya :

- Tentukan nama sistemnya.
- Tentukan batasan sistemnya.
- Tentukan terminator apa saja yang ada dalam sistem.
- Tentukan apa yang diterima/diberikan terminator dari/ke sistem.
- Gambarkan diagram konteks.

4. Buat Diagram *Level Zero*

   Diagram ini adalah dekomposisi dari diagram konteks. Caranya :

   - Tentukan proses utama yang ada pada sistem.
   - Tentukan apa yang diberikan/diterima masing-masing proses ke/dari sistem sambil memperhatikan konsep keseimbangan (alur data yang keluar/masuk dari suatu level harus sama dengan alur data yang masuk/keluar pada level berikutnya).
   - Apabila diperlukan, munculkan data store (master) sebagai sumber maupun tujuan alur data.
   - Gambarkan diagram *level zero*.
     - Hindari perpotongan arus data.
     - Beri nomor pada proses utama (nomor tidak menunjukkan urutan proses).

5. Buat Diagram *Level* Satu

   Diagram ini merupakan dekomposisi dari diagram level zero. Caranya :

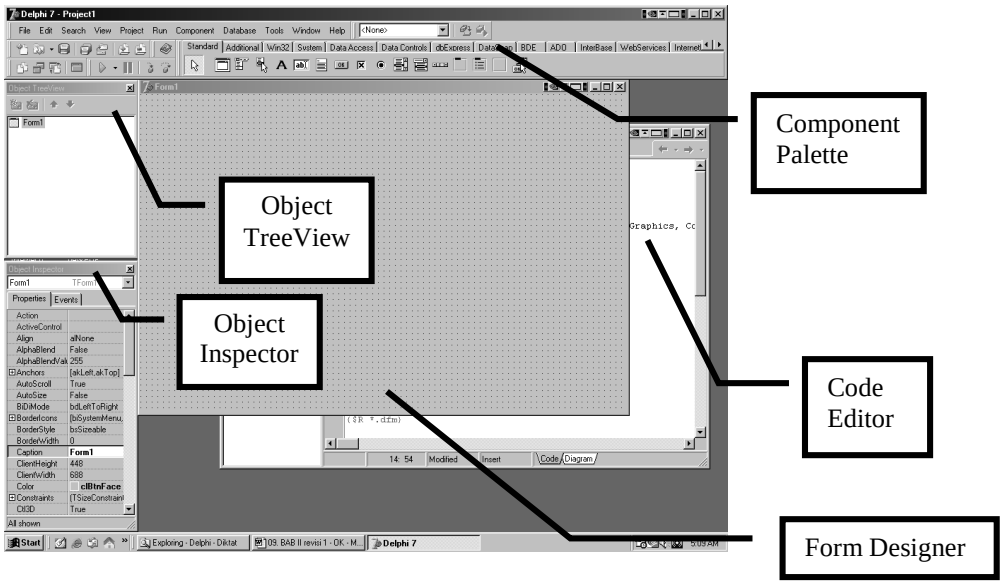
   - Tentukan proses yang lebih kecil (sub-proses) dari proses utama yang ada di *level* zero.
   - Tentukan apa yang diberikan/diterima masing-masing sub-proses ke/dari sistem dan perhatikan konsep keseimbangan.
   - Apabila diperlukan, munculkan data store (transaksi) sebagai sumber maupun tujuan alur data.
   - Gambarkan DFD *level* Satu
     - Hindari perpotongan arus data.
     - Beri nomor pada masing-masing sub-proses yang menunjukkan dekomposisi dari proses sebelumnya.

6. DFD *Level* Dua, Tiga, …

Diagram ini merupakan dekomposisi dari *level* sebelumnya. Proses dekomposisi dilakukan sampai dengan proses siap dituangkan ke dalam program. Aturan yang digunakan sama dengan *level* satu.

2.11 Delphi 7

Borland Delphi atau yang biasa disebut *Delphi* saja, merupakan sarana pemrograman aplikasi *visual*. Bahasa pemrograman yang digunakan adalah bahasa pemrograman *pascal* atau yang kemudian juga disebut bahasa pemrograman *Delphi*. *Delphi* adalah suatu bahasa pemrograman yang telah memanfaatkan metoda pemrograman *Object Oriented Programming* (OOP). Adapun tampilan program *delphi* dapat dilihat pada gambar 2.26.

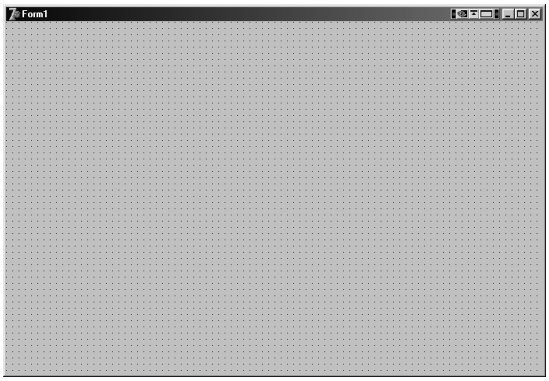


Gambar 2.26. Program Borland Delphi

Tampilan sarana pengembangan aplikasi yang terdapat pada lingkungan kerja *Delphi* dapat dilihat pada gambar 2.26. Berikut penjelasan masing-masing bagian tersebut:

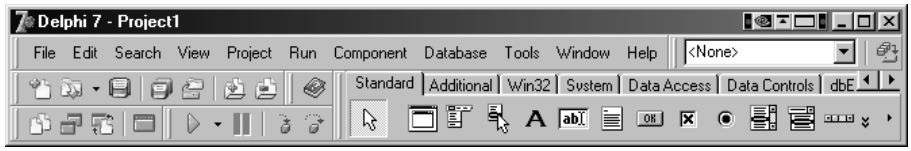
- *Form Designer* atau *form* adalah *windows* kosong tempat merancang antarmuka pemakai (*user interface*) aplikasi. Tampilan awalnya

seperti pada gambar 2.27. pada form inilah ditempatkan komponen-konponen sehingga aplikasi dapat berinteraksi dengan pemakainya.



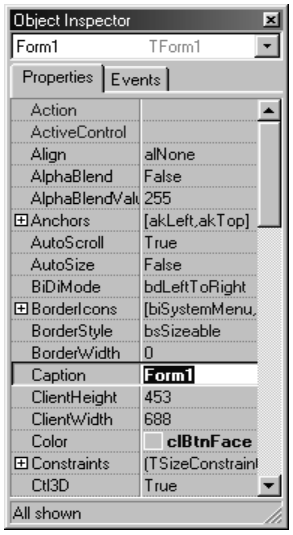
Gambar 2.27. *Form Designer* pada *Delphi*

- *Componen Palette*, berisi ikon-ikon komponen *visual* dan *nonvisual* yang dapat digunakan untuk merancang antarmuka bagi pemakai aplikasi. Komponen *palette* terdiri atas beberapa *page* yang dipakai sebagai pengelompok jenis komponen, misalnya yang tampak pada gambar 2.28 adalah *page standard*.



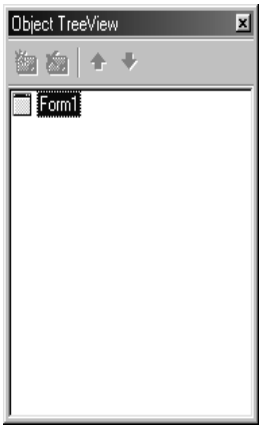
Gambar 2.28. *Component Palette*

- *Object Inspector*, untuk menentukan dan mengubah *property* (atribut) dan *event object*. Selain itu dapat juga dipilih komponen melalui *object inspector*. Tampilan *object inspector* adalah seperti yang terlihat pada gambar 2.29.



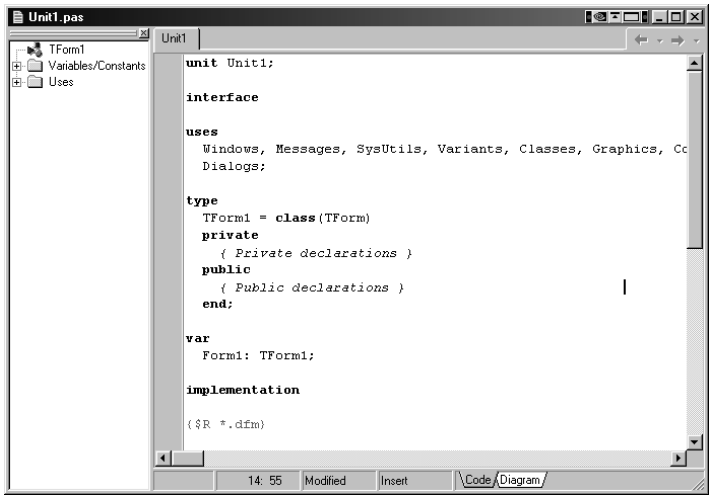
Gambar 2.29. *Windows Object Inspector*

- *Object TreeView* untuk menampilkan dan mengubah hubungan logis antar komponen di dalam projek. Contoh tampilan *object treeview* dapat dilihat pada gambar 2.30.



Gambar 2.30. *Windows Object TreeView*

- *Code Editor*, berfungsi untuk menulis dan menyunting kode program. Lokasi kode editor ada di belakang form. Pada gambar 2.31 adalah contoh tampilan kode editor.



Gambar 2.31. *Windows Code Editor*