

**RANCANG BANGUN SISTEM PERINGATAN DINI BENCANA BANJIR
MELALUI SMS BERBASIS MIKROKONTROLER PIC16F877A**

TUGAS AKHIR

Diajukan sebagai salah satu syarat
untuk memperoleh Gelar Sarjana Teknik pada
Jurusan Teknik Elektro



oleh :

RAJES NANDA ZETRI
10655004551

FAKULTAS SAINS & TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SULTHAN SYARIF KASIM RIAU
PEKANBARU

2013

RANCANG BANGUN SISTEM PERINGATAN DINI BENCANA BANJIR MELALUI SMS BERBASIS MIKROKONTROLER PIC16F877A

RAJES NANDA ZETRI

NIM : 10655004551

Tanggal Sidang : 26 Juni 2013

Tanggal Wisuda : November 2013

Jurusan Teknik Elektro

Fakultas Sains dan Teknologi

Universitas Islam Negeri Sultan Syarif Kasim Riau

Jl. Soebrantas No. 155 Pekanbaru

ABSTRAK

Banjir di beberapa daerah baru-baru ini telah menimbulkan kerugian yang cukup signifikan. Sementara sistem peringatan banjir yang sudah ada hanya memberitahukan bahwa banjir akan datang tanpa adanya prediksi kapan musibah tersebut akan terjadi. Penelitian ini menggunakan metode kuantitatif dengan memperhatikan variabel ketinggian air yang diukur menggunakan sensor ultrasonik untuk mengetahui status ketinggian air dan variabel debit air yang dihitung menggunakan flowmeter untuk memprediksi kapan banjir akan terjadi. Informasi peringatan musibah banjir diberikan melalui layanan sms dan bunyi sirine. Perancangan tersebut akhirnya menghasilkan suatu sistem peringatan dini bencana banjir yang dilengkapi prediksi kapan banjir akan terjadi. Hasil perancangan sistem yang diimplementasikan dalam bentuk contoh asli sistem peringatan dini telah bekerja dengan baik sesuai dengan perancangan yang direncanakan. Peringatan bencana diberikan dalam bentuk pesan singkat peringatan bencana banjir kepada masyarakat di daerah rawan banjir dan buzzer sebagai media peringatan lain yang berfungsi memberikan peringatan langsung jika pesan peringatan bencana tidak terkirim ke nomor tujuan.

Kata Kunci : sistem peringatan dini, flowmeter, sensor ultrasonik, layanan sms, sirine

ENGINEERING OF FLOODING DISASTER EARLY WARNING SYSTEM VIA SMS MICROCONTROLLER PIC16F877A BASED

RAJES NANDA ZETRI

NIM : 10655004551

Date of Final Exam : June, 26, 2013

Department of Electrical Engineering

Faculty of Science and Technology

State Islamic University of Sultan Syarif Kasim Riau

Soebrantas St. No. 155 Pekanbaru - Indonesia

ABSTRACT

Flood at severally region of late have evoked loss that adequately significantly. While existing flooding warning system just informs that upcoming flood without marks sense to predict when that accident will happen. This research utilize quantitative method by notices water high variable that measured by use the ultrasonic sensor to know water high state and water debit variable that accounting to utilize flowmeter to predict when flood will happen. Accidents reprovred information flooding be given service thru sms and siren sound. Design of that finally results a flooding disaster early warning system that furnished predicts when flood will happen. System scheme result that implemented in forms early warning system prototype was working with every consideration in accordance with scheme that is plotted. Disaster warning is given in form orders to abbreviate flooding disaster warning to society at flooding gristle region and buzzer as media of functioning other warning give direct warning if send away for warning disaster not most ship goes to aim number.

Keywords :*Early Warning System, flowmeter, ultrasonic sensor, sms service, siren*

KATA PENGANTAR

Puji syukur yang sedalam-dalamnya penulis ucapkan kehadirat Allah SWT karena atas berkat rahmat dan hidayahnyalah sampai saat ini kita masih dapat menikmati indahnya kehidupan yang fana di muka Bumi-Nya ini. Shalawat berring salam tak lupa penulis sampaikan kepada junjungan alam Nabi Besar Muhammad SAW yang telah membawa kita dari zaman jahiliyah yang dipenuhi keburukan kepada zaman yang terang benderang dan penuh ilmu pengetahuan seperti yang kita rasakan sekarang ini sehingga pada saat ini penulis dapat menyelesaikan tugas akhir ini.

Pada kesempatan ini, penulis juga mengucapkan terima kasih yang sedalam-dalamnya kepada :

1. Orang tua dan keluarga besar Penulis yang telah memberikan dukungan dan apresiasi yang sangat besar dalam setiap kegiatan yang penulis laksanakan selama penelitian tugas akhir ini.
2. Bapak Prof. DR. H. M. Nazir Karim M.A selaku Rektor Universitas Islam Negeri Sulthan Syarif Kasim Riau.
3. Ibu Dra. Hj. Yenita Morena, M.Si selaku Dekan Fakultas Sains & Teknologi Universitas Islam Negeri Sulthan Syarif Kasim Riau.
4. Bapak Kunaifi, PgDipEnSt, M.Sc selaku Ketua Jurusan Teknik Elektro Fakultas Sains & Teknologi Universitas Islam Negeri Sulthan Syarif Kasim Riau sekaligus Penasehat Akademis.
5. Bapak DR. Alex Wenda dan bapak Abdillah, S.Si., M.IT selaku pembimbing tugas akhir, terima kasih yang sedalam-dalamnya penulis sampaikan atas bimbingannya selama penyusunan tugas akhir ini.
6. Bapak / Ibu Dosen Jurusan Teknik Elektro Fakultas Sains & Teknologi Universitas Islam Negeri Sulthan Syarif Kasim Riau.
7. Rekan-rekan seperjuangan mahasiswa Teknik Elektro dan pihak-pihak yang telah membantu penulis dalam penyusunan tugas akhir ini.

Penulis amatlah menyadari bahwa penyusunan laporan Tugas akhir ini masih sangat jauh dari kesempurnaan. Karena itu penulis sangat mengharapkan kritik, saran dan masukan yang dapat menyempurnakan tugas akhir ini pada masa yang akan datang.

Pekanbaru, Juni 2013

Rajes Nanda Zetri

DAFTAR ISI

LEMBAR PERSETUJUAN	ii
LEMBAR PENGESAHAN	iii
LEMBAR HAK ATAS KEKAYAAN INTELEKTUAL	iv
LEMBAR PERNYATAAN	v
LEMBAR PERSEMBAHAN	vi
ABSTRAK	vii
ABSTRACT	viii
KATA PENGANTAR	ix
DAFTAR ISI	xi
DAFTAR GAMBAR	xiv
DAFTAR TABEL	xvi
DAFTAR RUMUS	xvii
DAFTAR LAMBANG	xviii
DAFTAR LAMPIRAN	xix

BAB I PENDAHULUAN

1.1 Latar Belakang	I-1
1.2 Rumusan Masalah	I-2
1.3 Tujuan	I-2
1.4 Batasan Masalah	I-2

BAB II TINJAUAN PUSTAKA

2.1 Penelitian Terkait	II-1
2.2 Sensor Ultrasonik	II-3
2.3 Flow Meter	II-5
2.4 Mikrokontroler	II-8
2.4.1. Central Processing Unit (CPU)	II-8
2.4.2. Memori	II-10
2.4.3. Sistem Bus	II-11
2.4.4. I/O (Input-Output)	II-11

2.4.5.	Unit Pewaktuan (Timer)	II-12
2.4.6.	WDT (Watchdog Timer)	II-13
2.4.7.	ADC (Analog to Digital Converter).....	II-13
2.4.8.	Komunikasi Serial (Serial Communication)	II-14
2.5	Teknologi GSM.....	II-15
2.5.1.	Pengenalan GSM.....	II-15
2.5.2.	Jaringan GSM.....	II-15
2.5.3.	Spesifikasi GSM.....	II-16
2.6	Layanan SMS	II-17
2.6.1	Pengenalan SMS.....	II-17
2.6.2	Short Message Service (SMS).....	II-17
2.6.3	Manfaat SMS.....	II-21
2.6.4	Peralatan	II-24
2.6.5	Aplikasi SMS	II-26
2.7	Dasar-Dasar Pemrograman Bahasa C	II-27
2.7.1.	Komentar	II-29
2.7.2.	Tipe data di bahasa C	II-29
2.7.3.	Variabel	II-30
2.7.4.	Konstanta.....	II-33
2.7.5.	Operator, Operasi Dan Ungkapan	II-36
2.7.6.	Konversi Tipe Data	II-40
2.7.7.	Operator Bersyarat	II-41
2.7.8.	Program Loop.....	II-44
2.7.9.	Menulis Kode Dalam Bahasa Assembly	II-46
2.7.10.	Array.....	II-48
2.7.11.	Fungsi	II-50

BAB II METODOLOGI PENELITIAN

2.1	Jenis penelitian	III-1
2.2	Tahapan Penelitian	III-2
2.2.1	Studi Literatur.....	III-3
2.2.2	Perancangan Sistem.....	III-3
2.2.3	Pengujian Sistem	III-11

2.2.4	Pembuatan Laporan	III-12
2.3	Jenis dan Metode Pengumpulan Data	III-12
2.4	Analisa Data	III-12

BAB IV HASIL DAN ANALISA

4.1	Hasil <i>Prototype</i> Sistem.....	IV-1
4.2	Analisa Cara Kerja Mikrokontroler	IV-3
4.3	Analisa Pengujian Sensor Ketinggian air.....	IV-5
4.4	Analisa Pengujian Sensor Kecepatan air.....	IV-10
4.5	Pengujian Program Peringatan Bencana Banjir	IV-13
4.6	Pengujian Program SMS Melalui Modem GSM	IV-17
4.7	Pengujian Buzzer	IV-22

BAB V KESIMPULAN DAN SARAN

5.1	Kesimpulan	V-1
5.2	Saran.....	V-1

DAFTAR PUSTAKA

LAMPIRAN

DAFTAR RIWAYAT HIDUP

BAB I

PENDAHULUAN

1.5 Latar Belakang

Musibah banjir merupakan masalah yang umum terjadi disemua daerah di Indonesia baik di Provinsi Riau maupun di beberapa daerah di Indonesia. Lambatnya pemberitahuan bencana yang terjadi selama ini menyebabkan terjadinya keterlambatan penanggulangan bencana yang berujung pada kerugian materi maupun korban jiwa yang cukup memprihatinkan. Hal ini diperburuk oleh ketiadaan data lokasi rawan banjir terbaru dalam data pemerintah provinsi. Beberapa kejadian banjir terakhir yang terjadi pada akhir tahun 2011 dapat dilihat dalam uraian berikut

1. Banjir yang terjadi di Kecamatan Kampar Kiri Kabupaten Kampar pada 25 November 2011 pukul 05:00 WIB menyebabkan 2 orang tewas, 307 unit rumah terendam, 21 unit rumah hanyut, merendam 37 Ha kebun karet, menghanyutkan 55 ton karet, merusak 55 Ha perkebunan kelapa sawit, merusak 4 unit jembatan dan menghanyutkan 53 unit perahu nelayan. (TribunNews Edisi 28 November 2011).
2. Banjir yang terjadi di Kecamatan Gunung Sahilan Kabupaten kampar sejak 18 Desember 2011 merendam 189 unit rumah, sekitar 625 Ha kebun kelapa sawit, serta kebun karet seluas 154 hektar dengan ketinggian air sekitar 30 cm. Banjir merendam pemukiman warga hingga 30 centimeter di lima desa yakni Desa Gunung Sahilan, Sahilan Darussalam, Subarak, Sungai Lipai dan Kebun Durian. (Seruu.com edisi Senin, 19 Desember 2011 pukul 17:48)
3. Banjir yang terjadi di Kecamatan kampar kiri Kabupaten Kampar sejak 26 Desember 2011 pukul 01:00 merendam 1.151 unit rumah milik 1.283 KK yang dihuni 5.017 jiwa dengan ketinggian air mencapai 1,5 m. Banjir juga merendam 184 Ha lahan pertanian, merendam 245 Ha lahan peternakan, merusak fasilitas umum dan memutuskan jalur transportasi darat. (TvOneNews, 27 Desember 2011).

Melihat uraian diatas sudah sewajarnya jika kemudian masalah banjir ini menjadi perhatian kita semua. Pemberitahuan bencana dengan sistem manual yang selama ini diterapkan di daerah rawan banjir seringkali tidak berjalan efektif dan membutuhkan waktu dan biaya ekstra dalam penyampaian informasi mengenai datangnya musibah. Hal ini menyebabkan penanggulangan banjir berjalan lambat sehingga tidak efisien dalam pelaksanaan sistemnya.

Salah satu solusi yang mampu membantu masyarakat dalam hal tersebut adalah sistem peringatan dini bencana banjir yang bertujuan memberikan informasi dan peringatan dini mengenai datangnya bencana saat ketinggian air di luar batas normal. Sistem peringatan bencana banjir pada saat ini masih banyak menggunakan sistem manual melalui surat edaran ataupun pengumuman secara langsung. Sistem ini dinilai kurang efisien karena pemberitahuan yang lambat dan juga terkadang tidak tepat sasaran. Untuk itu penulis memiliki ide untuk merancang sebuah sistem yang mampu memberikan peringatan bencana secara cepat dan efisien berikut prediksi kapan bencana tersebut akan terjadi.

1.6 Rumusan Masalah

Berdasarkan latar belakang diatas maka penulis mengambil rumusan masalah ***“Bagaimana merancang dan mengimplementasikan sistem peringatan dini bencana banjir berbasis mikrokontroler PIC.”***

1.7 Tujuan

Adapun tujuan dari tugas akhir ini adalah merancang prototype sistem peringatan dini bencana banjir yang dapat memberitahukan secara dini datangnya musibah banjir berikut prediksi kapan musibah tersebut akan datang.

1.8 Batasan Masalah

Dalam tugas akhir ini penulis mengambil batasan masalah sebagai berikut :

1. Sistem menggunakan sensor ultrasonik sebagai sensor ketinggian air dan menggunakan sensor kecepatan untuk mengukur kecepatan aliran air.
2. Perancangan perangkat keras menggunakan mikrokontroler PIC16F877A dan modem GSM.
3. Perancangan perangkat lunak menggunakan bahasa pemrograman C.
4. Output berupa pesan singkat yang dikirimkan melalui pemancar berupa modem GSM kepada masyarakat di daerah rawan banjir.
5. Pesan singkat yang dikirimkan berisi peringatan mengenai datangnya musibah banjir berikut prediksi kapan musibah itu akan terjadi.
6. Sistem dilengkapi dengan *buzzer* sebagai antisipasi jika media SMS tidak terkirim.
7. Sistem diimplementasikan dalam bentuk *prototype*.

BAB II

TINJAUAN PUSTAKA

2.8 Penelitian Terkait

Penelitian mengenai sistem peringatan dini bencana banjir sendiri telah dilakukan oleh beberapa peneliti di Indonesia. Beberapa penelitian tersebut antara lain :

1. Penelitian oleh Venoruton Budi Irmawan (2007) dalam tugas akhir yang berjudul *“Deteksi Ketinggian Air Dengan Media Gelombang Radio Berbasis Mikrokontroler AT89S51”* dengan hasil akhir berupa *prototype* sistem yang terdiri dari sensor ketinggian air, IC DTMF8888, Mikrokontroler AT89S51 dan komputer *desktop* dengan sistem kerja sebagai berikut : Sensor permukaan air akan selalu membaca permukaan air. Baik pada kondisi normal maupun kondisi bahaya, sensor akan selalu mengirimkan data melalui gelombang radio yang terpasang pada unit sensor tersebut. Mikrokontroler akan merubah data yang masuk menjadi nada-nada DTMF dengan bantuan IC DTMF MT8888. Kemudian mengirimkannya melalui pemancar ke udara. Kemudian diterima oleh rangkaian penerima dan diterjemah menjadi nada-nada DTMF oleh rangkain DTMF RX. Dari rangkain DTMF RX kemudian dirubah menjadi data digital yang kemudian diumpankan ke mikrokontroler baru dikirim ke PC melalui *serial port*.
2. Penelitian oleh Joni Gunawan, Kartika Firdausy dan Wahyu Sapto Aji (2008) dengan judul *“Otomatisasi Pendeteksi Dini Bahaya Banjir menggunakan SMS Berbasis Mikrokontroler AT89S52”* yang menghasilkan *prototype* sistem yang disusun oleh sistem minimum mikrokontroler AT89S52 sebagai pengendali, rangkaian *interface* RS232 untuk komunikasi serial, sensor ultrasonik sebagai pengukur *level* ketinggian air, modul LCD dot matriks 2x16 karakter sebagai penampil hasil pengukuran dan bagian output yang berupa alarm dan LED sebagai indikator dengan cara kerja sebagai berikut : Sistem pendeteksi dini bahaya banjir akan menyalakan alarm dan LED sebagai indikator peringatan dini bahwa telah terjadi atau kemungkinan akan terjadi banjir. Sistem juga akan mengirim pesan SMS ke dua nomor HP yang berbeda untuk memberitahukan level, kecepatan, dan kondisi perubahan ketinggian air.
3. Penelitian oleh Dr. Ir. Harry Soekotjo Dachlan, M.Sc Dkk (2010) dengan judul *“Sistem Penunjang Keputusan Deteksi Dini Bahaya Banjir Secara Real Time*

Menggunakan Data Logger Berbasis GSM” menggunakan beberapa perangkat yaitu :

- a. Modul sensor ultrasonic
- b. Modul sensor pH
- c. Modul sensor suhu & kelembapan
- d. *Datalogger* berbasis GSM

Perancangan sistem dengan menggunakan perangkat sebagaimana disebutkan diatas menghasilkan prototype dengan cara kerja : Modul sensor akan memantau parameter yang ditentukan. Kemudian data dikumpulkan oleh datalogger lalu ditransmisikan ke stasiun master melalui SMS *Gateway* (*Dial-up* antara HP GSM yang terhubung pada datalogger dan HP GSM di stasiun master).

4. Penelitian oleh Aditya Pradhana (2011) dengan judul “***Peringatan Dini Bahaya Banjir Di Tambang Pasir Berbasis Sms Gateway***” dengan komponen penyusun sistem berupa Sensor elektroda, *Optocoupler*, Mikrokontroler ATmega8535, HP *transmitter*, *receiver* dan LCD ISD25120 dengan cara kerja sistem sebagai berikut : Pembacaan ketinggian level air dan kecepatan aliran air dengan memanfaatkan sensor elektroda, *optocoupler*, dan mikrokontroler ATmega8535. Data ketinggian level air dan kecepatan aliran air dikirim secara otomatis melalui *handphone* dari *transmitter* ke *receiver*. Data yang diterima *receiver* diolah oleh mikrokontroler yang akan ditampilkan pada LCD dan dalam bentuk suara (ISD25120) melalui pengeras suara.

Setelah mempelajari penelitian diatas penulis dapat mengemukakan beberapa kelebihan dalam penelitian kali ini yaitu :

1. Penelitian ini menggunakan layanan pesan singkat sebagai *output* sistem dimana pesan singkat dikirimkan berisi informasi status ketinggian permukaan air berikut prediksi kapan musibah banjir akan terjadi
2. Layanan pesan singkat informasi kondisi ketinggian permukaan air berikut prediksi sebagaimana dijelaskan pada poin 1 dikirimkan langsung kepada masyarakat di daerah rawan bencana banjir.

2.9 Sensor Ultrasonik

Gelombang ultrasonik adalah gelombang dengan besar frekuensi di atas frekuensi gelombang suara yaitu lebih dari 20 kHz. Sensor ultrasonik terdiri atas rangkaian pemancar

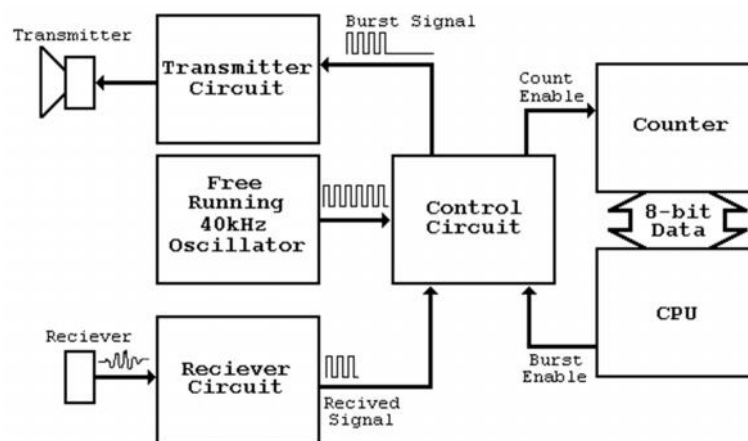
ultrasonik yang disebut *transmitter* dan rangkaian penerima ultrasonik yang disebut *receiver*. Sinyal ultrasonik yang dibangkitkan akan dipancarkan dari *transmitter* ultrasonik. Ketika sinyal mengenai benda penghalang, maka sinyal ini dipantulkan, dan diterima oleh *receiver* ultrasonik. Sinyal yang diterima oleh rangkaian *receiver* dikirimkan ke rangkaian mikrokontroler untuk selanjutnya diolah untuk menghitung jarak terhadap benda di depannya (bidang pantul).

Prinsip kerja dari sensor ultrasonik adalah sebagai berikut :

1. Sinyal dipancarkan oleh pemancar ultrasonik. Sinyal tersebut berfrekuensi diatas 20 kHz, biasanya yang digunakan untuk mengukur jarak benda adalah 40 kHz. Sinyal tersebut di bangkitkan oleh rangkaian pemancar ultrasonik.
2. Sinyal yang dipancarkan tersebut kemudian akan merambat sebagai sinyal /gelombang bunyi dengan kecepatan bunyi yang berkisar 340 m/s. Sinyal tersebut kemudian akan dipantulkan dan akan diterima kembali oleh bagian penerima Ultrasonik.
3. Setelah sinyal tersebut sampai di penerima ultrasonik, kemudian sinyal tersebut akan diproses untuk menghitung jaraknya. Jarak dihitung berdasarkan rumus :

$$S = \frac{340 \cdot t}{2} \dots (2.1)$$

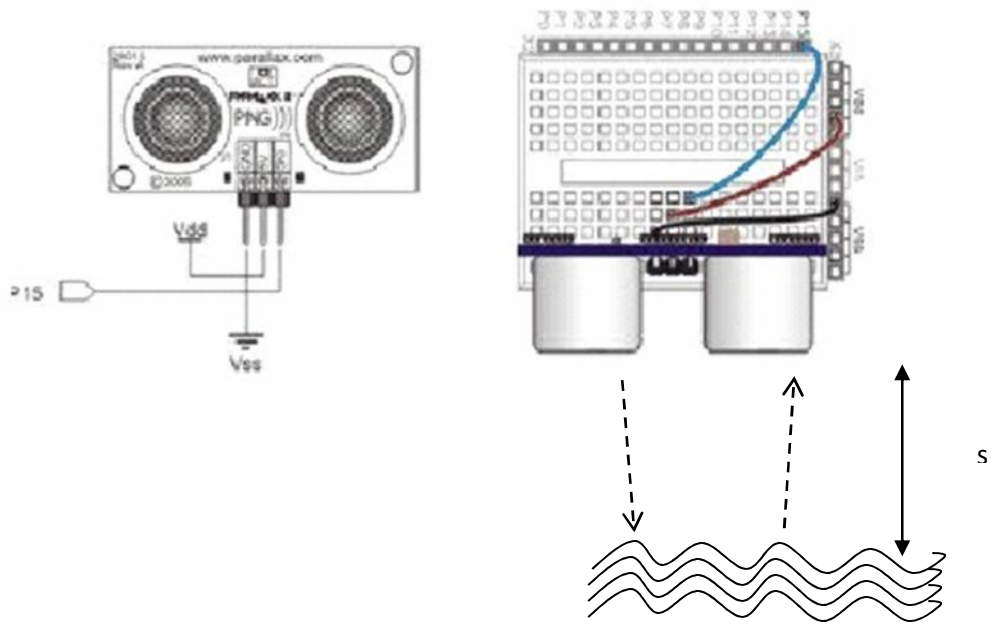
dimana S adalah jarak antara sensor ultrasonik dengan bidang pantul, dan t adalah selisih waktu antara pemancaran gelombang ultrasonik sampai diterima kembali oleh bagian penerima ultrasonik. (atmelmikrokontroler.wordpress.com edisi 24 Juni 2009).



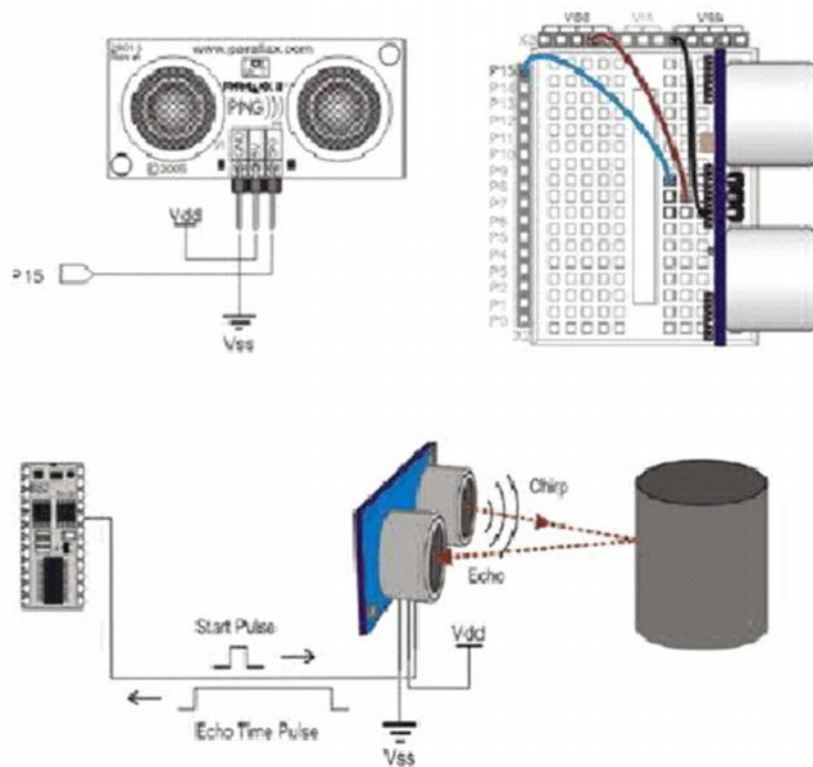
Gambar 2.1 Diagram blok sensor ultrasonik

(sumber : anonim, Micro Mouse Project,

<http://www.dominikdeak.com/index.php?page=micro-mouse-project>, 2006)



Gambar 2.2 Ilustrasi pemantulan gelombang ultrasonik



Gambar 2.3 Ilustrasi cara kerja sensor Ultrasonik

(sumber : Muhammad Yusuf Z., Alat Pengaman Jarak Antar Kendaraan Guna Mengurangi Angka Kecelakaan Di Jalan Raya,

<http://dc235.4shared.com/img/auWHXStg/preview.html>, 2009)

Untuk pengaktifan sensor ultrasonik, hubungkan Pin V_{ss} ke *Ground*, kemudian pin V_{dd} ke catu daya yang keluarannya sudah diset 5V, setelah batere dihubungkan dengan IC Regulator 7805, tinggal Pin SIG dihubungkan ke pin di Mikrokontroler, buat sensor ke port P1.7, sedangkan indikator output P3.7.

2.10 Flow Meter

Sensor kecepatan air sungai yang digunakan dalam penelitian ini adalah flow meter. Flow meter adalah alat yang digunakan untuk mengukur laju aliran linear, nonlinear, massa atau volume dari liquid, gas, ataupun solid. (wmablog.com, 27 Juni 2011).

Flow meter menggunakan sensor *hall effect* dengan keluaran berupa pulsa. Sensor *hall effect* berupa sebuah magnet yang diletakkan tegak lurus terhadap sepasang keping konduktor. Posisi magnet yang tegak lurus dengan konduktor menyebabkan munculnya tegangan pada sisi yang berlawanan dengan konduktor. Tegangan yang muncul ini disebut tegangan *Hall* yang dihitung berdasarkan rumus :

$$V_H = \frac{BI}{n_1 e d} \dots (2.2)$$

Dimana :

V_H = Tegangan Hall (Volt)

B = Medan magnet (Tesla)

I = Arus yang mengalir (Ampere)

d = Ketebalan bahan konduktor (meter)

n_1 = Rapat muatan

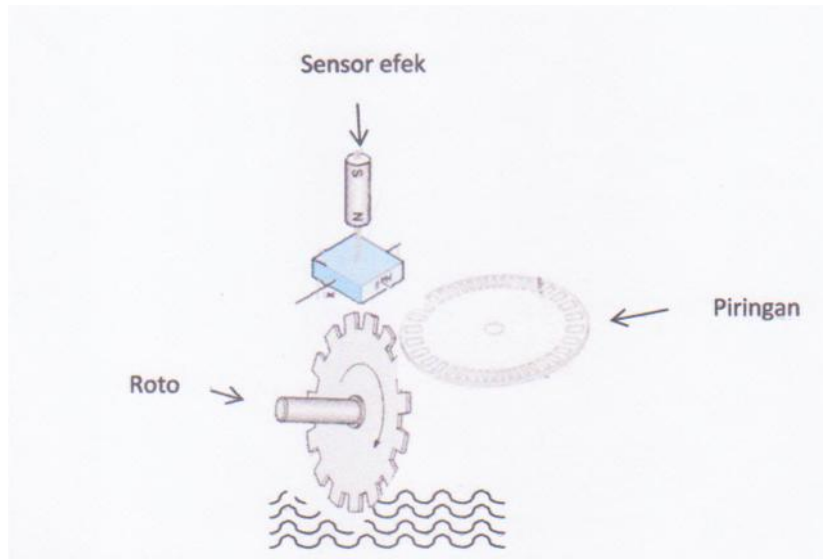
e = Muatan elektron (1.602×10^{-19} C) (sumber : scribd.com)



Gambar 2.4. Prinsip kerja sensor efek hall

(sumber : Chris Warner, Hall-effect Sensors Measure Up,

<http://www.ecnmag.com/articles/2009/03/sensor-zone-april-2009>)



Gambar 2.5. Hubungan antara sensor efek hall, rotor air dan piringan sensor optik

Water flow sensor ini terdiri atas katup plastik, rotor air, sebuah sensor hall-effect dan piringan sensor optik. Ketika air mengalir maka akan mengenai rotor, dan membuatnya berputar. Kecepatan putar rotor akan berubah ketika kecepatan aliran air berubah pula. Putaran rotor air akan ikut menggerakkan piringan sensor optik dengan kecepatan putaran sebesar yang dihitung berdasarkan rumus :

$$\dot{S} = \frac{\ell}{t \cdot L} \dots (2.3)$$

Dimana :

= kecepatan putar piringan sensor optik (rad/s)

ℓ = lebar celah piringan optik (meter)

t = waktu saat cahaya melewati celah piringan optik (sekon)

L = jarak celah dengan titik pusat piringan piringan optik (meter) (sumber : portal.fi.itb.ac.id)

Setelah nilai diketahui maka kita dapat menghitung pulsa frekuensi piringan sensor optik dengan rumus :

$$f = \frac{\dot{S} \cdot n_2}{60} \dots (2.4)$$

Dimana :

f = pulsa frekuensi piringan sensor optik (Hz)

= kecepatan putar piringan sensor optik (rad/s)

n_2 = jumlah celah piringan sensor optik (sumber : konversi.wordpress.com edisi 29 Juni 2012)

Sedangkan jarak antar celah piringan sensor optik dihitung dengan rumus :

$$K = \frac{2fr}{n} \dots (2.5)$$

Dimana :

K = konstanta atau jarak antar celah piringan sensor optik (meter)

r = jari – jari piringan sensor optik (meter)

n = jumlah celah piringan sensor optik (sumber : <http://elib.pdii.lipi.go.id>)

Setelah kecepatan frekuensi piringan sensor optik maka kecepatan aliran air dapat dihitung dengan rumus :

$$v = K \times f \dots (2.6)$$

Dimana :

v = debit air (liter/menit)

K = konstanta atau jarak antar celah piringan sensor optik (meter)

f = pulsa frekuensi piringan sensor optik (Hz)

Hasil perhitungan kecepatan inilah yang kemudian dibaca dan dikonversi oleh blok *Analog to Digital Converter* (ADC) pada mikrokontroler sebagai dasar perhitungan prediksi bencana banjir.

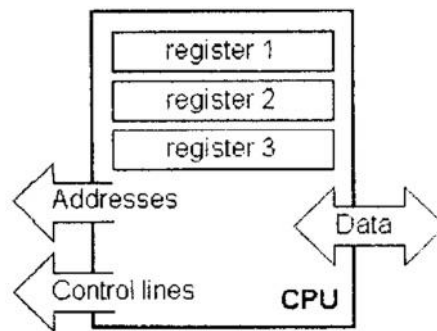
2.11 Mikrokontroler

Mikrokontroler adalah IC yang dapat diprogram berulang kali, baik ditulis atau dihapus (Agus Bejo, 2007). Biasanya digunakan untuk pengontrolan otomatis dan manual pada perangkat elektronika. Beberapa tahun terakhir, mikrokontroler sangat banyak digunakan terutama dalam pengontrolan robot. Seiring perkembangan elektronika, mikrokontroler dibuat semakin kompak dengan bahasa pemrograman yang juga ikut berubah. Dalam suatu mikrokontroler terdiri dari beberapa bagian sehingga dapat bekerja secara utuh seperti komputer, yaitu CPU, Memori. Bus, I/O. Unit Waktu (Timer), Watchdog Timer, ADC, dan Komunikasi serial.

2.4.9. Central Processing Unit (CPU)

CPU merupakan otak dari seluruh sistem. Di dalam CPU terdapat ALU (*Arithmetic Logic Unit*) yang berfungsi untuk melakukan operasi aritmetika dan logika. Fungsi dari CPU ini sama dengan fungsi mikroprosesor komputer, yaitu untuk mengeksekusi program, baik melakukan perhitungan aritmetika, logika, maupun menyalin data. Di dalam CPU

terdapat register yang digunakan untuk menyimpan data sementara, baik itu merupakan data hasil pengolahan maupun data yang akan diolah.



Gambar 2.6. Bagian – bagian CPU

(sumber : Husanto & Thomas, Kupas Tuntas Mikrokontroler PIC16F84A, Penerbit Andi, 2008)

Berdasarkan Gambar 2.6 dapat dilihat bahwa dalam suatu CPU terdapat register-register yang berfungsi untuk menyimpan data sementara yang akan diolah seperti yang telah dijelaskan diatas. Selain itu, dalam CPU memiliki 3 buah jalur yang digunakan untuk memproses data. Jalur tersebut adalah jalur alamat (*Address Bus*), jalur data (*Data Bus*), dan jalur kontrol (*Control Bus*).

1. Jalur Alamat (*Address Bus*)

Jalur alamat berfungsi untuk menentukan alamat instruksi ataupun data yang akan diproses. Sebagai gambaran proses ini misalkan kita ingin mencari rumah teman, maka kita membutuhkan alamat dari rumah teman tersebut. Sama halnya CPU juga membutuhkan alamat data yang akan diproses.

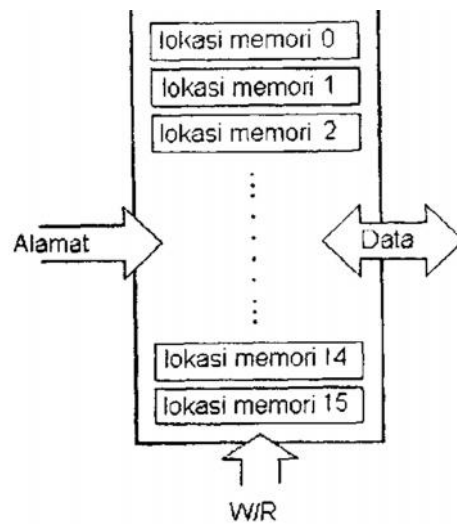
2. Jalur Control (*Control Bus*)

Jalur ini berfungsi untuk menentukan kegiatan apa yang akan dilakukan terhadap alamat data yang telah ditemukan, apakah membaca atau menulis data dari/ke alamat tersebut. Jadi, jika kita telah menemukan alamat rumah teman, apa yang kita lakukan, mengajaknya keluar atau menitip barang di rumahnya.

3. Jalur Data (*Data Bus*)

Jalur data berfungsi sebagai tempat lewatnya data yang akan dikirim atau diterima. Gambarannya seperti jalan raya sebagai tempat lewatnya teman yang Anda ajak.

2.4.10. Memori



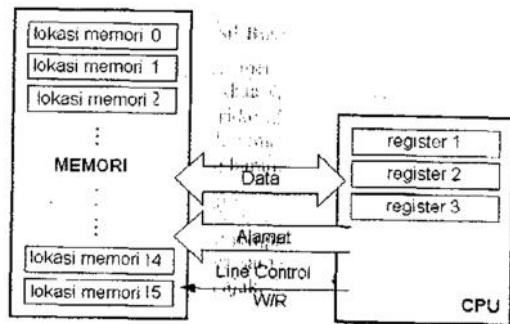
Gambar 2.7. Memori

(sumber : Husanto & Thomas, Kupas Tuntas Mikrokontroler PIC16F84A, Penerbit Andi, 2008)

Memori adalah bagian dari mikrokontroler yang berfungsi untuk menyimpan data. Jadi, memori dapat digunakan untuk menyimpan program yang akan dieksekusi oleh CPU ataupun menyimpan data-data. Sebagai gambaran umum, memori ini dapat diistilahkan sebagai suatu kompleks rumah teman Anda. Jadi, dalam suatu kompleks terdapat banyak rumah. Ini sama halnya dengan memori mikrokontroler yang memiliki beberapa lokasi di mana setiap lokasi memiliki alamat sendiri-sendiri.

Agar dapat mengetahui rumah teman Anda, Anda membutuhkan alamat rumahnya, sama halnya dengan alamat memori, yaitu untuk mempermudah dalam mengaksesnya memori. Jadi, untuk mengakses memori kita perlu memberitahukan alamatnya lewat bus *addresses* sehingga isi memori pada alamat tersebut dapat diakses lewat bus *data*. Untuk menentukan apakah kita akan menyimpan (*Write Data*) atau membaca data (*Read Data*), maka pada memori perlu adanya bus yang memberi informasi, bus ini dinamakan bus W/R (*Control Bur*). Jadi, kerjanya adalah seperti berikut bila r/w (*Read/Write*) = 1, maka akan dilakukan pembacaan data dan bila r/w sebaliknya akan dilakukan penyimpanan data ke memori. Lalu antar CPU dan Memori perlu berkomunikasi agar data dari memori dapat dipindah ke CPU atau sebaliknya. Untuk itu, diperlukan sebuah 'jalan' yang dinamakan *bus*.

2.4.11. Sistem Bus



Gambar 1.4 Sistem Bus

Gambar 2.8. Sistem Bus

(sumber : Husanto & Thomas, Kupas Tuntas Mikrokontroler PIC16F84A, Penerbit Andi, 2008)

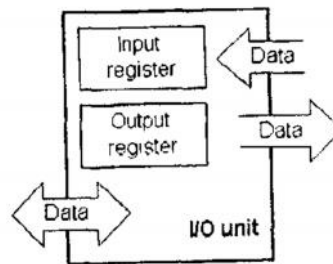
Sistem bus ini telah dijelaskan sebelumnya pada bagian CPU, namun pada bagian ini penulis akan memperjelas lagi kegunaannya. Bus berfungsi seperti `jalan' sehingga data dari memori dapat diambil oleh CPU atau sebaliknya. Jadi, antar bagian dalam mikrokontroler dihubungkan dengan bus-bus agar dapat saling berkomunikasi untuk menjalankan fungsinya secara sempurna. Berikut ini adalah gambar komunikasi antara CPU dan Memori melalui Bus.

Gambar 2.8 memberikan ilustrasi bus yang menghubungkan CPU dan memori, yaitu bus data, bus alamat (*address bus*), dan bus kontrol (*control bus*). Jadi, CPU dapat mengakses memori dengan cara memberitahu alamat mana yang akan diakses melalui bus alamat dan mode akses lewat bus control, lalu data yang ada pada alamat tersebut akan dipindahkan ke CPU lewat bus data.

Namun, mengingat CPU – memori selain digunakan untuk memproses dan menyimpan data secara intern, tentu juga harus dapat berkomunikasi dengan dunia luar atau dengan kita! Untuk itu, dapat dilakukan penambahan IO (*input-output*) pada mikrokontroler yang dihubungkan ke bagian lain dengan sistem bus, sehingga bagian dalam mikrokontroler seperti memori dan CPU dapat berkomunikasi dengan dunia luarnya.

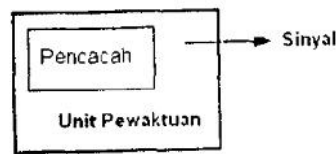
2.4.12. I/O (Input-Output)

Unit I/O merupakan sebuah blok dalam mikrokontroler yang akan menghubungkan internal mikrokontroler dengan dunia luar. Jadi, blok ini akan terhubung baik ke internal mikrokontroler melalui bus maupun ke luar yang berbentuk pin-pin IC, atau banyak orang menyebutnya port-port.



Gambar 2.9. Unit I/O

(sumber : Husanto & Thomas, Kupas Tuntas Mikrokontroler PIC16F84A, Penerbit Andi, 2008)



Gambar 2.10. Unit Pewaktuan (Timer)

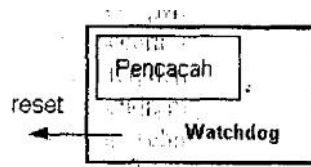
(sumber : Husanto & Thomas, Kupas Tuntas Mikrokontroler PIC16F84A, Penerbit Andi, 2008)

Ada beberapa jenis port, yaitu port keluaran, masukan. ataupun keduanya sekaligus. Jadi, kita dapat memasukkan data ataupun membaca data pada mikrokontroler melalui port-portnya.

2.4.13. Unit Pewaktuan (Timer)

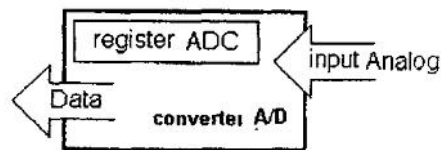
Dalam mikrokontroler juga diperlukan unit pewaktuan. Unit pewaktuan ini berfungsi untuk membuat kerja bagian dalam mikrokontroler menjadi sinkron. Jadi, untuk mengetahui saat kapan harus memindah data, menyimpan, ataupun mengeksekusi secara tepat diperlukan pewaktuan. Pewaktuan di dalam mikrokontroler berupa register pencacah (*counter*) yang nilainya terus bertambah satu untuk interval waktu tertentu secara tetap. Dengan adanya ini kita bisa mendapat informasi mengenai waktu, durasi, dan sebagainya.

2.4.14. WDT (Watchdog Timer)



Gambar 2.11. Watchdog Timer

(sumber : Husanto & Thomas, Kupas Tuntas Mikrokontroler PIC16F84A, Penerbit Andi, 2008)



Gambar 2.12. ADC

(sumber : Husanto & Thomas, Kupas Tuntas Mikrokontroler PIC16F84A, Penerbit Andi, 2008)

Saat kita menggunakan komputer, bila terjadi error atau hang, kita hanya menekan tombol reset untuk *me-restart* komputer. Tetapi pada mikrokontroler yang hanya berupa sebuah IC tidak memiliki tombol reset. diperlukan satu blok lagi agar jika terjadi '*hang*' pada mikrokontroler, secara otomatis mikrokontroler tersebut akan *di-reset*. Mikrokontroler telah memiliki pewaktuan yang berfungsi untuk mengatasi masalah yang mungkin timbul seperti yang telah dijelaskan sebelumnya. Timer ini dinamakan *Watchdog Timer*. *Watchdog timer* bekerja dengan cara menghitung waktu pengekseskuan CPU. Bila waktu pengekseskuan CPU melebihi batas waktu yang wajar (*hang*), maka *watchdog timer* akan mengirimkan logika 1 pada line reset agar CPU melakukan *reset*. Hal ini dapat terjadi pada waktu program mengalami proses pengulangan (*looping*) yang tidak berakhir. Dalam eksekusi normal, *watchdog* akan selalu mengirim logika 0 sehingga CPU tidak *di-reset*.

2.4.15. ADC (Analog to Digital Converter)

Beberapa mikrokontroler telah dilengkapi dengan blok tambahan, yaitu blok ADC. ADC berfungsi untuk menerima input dalam bentuk analog dari port input dan mengubahnya ke bentuk digital sehingga dapat diproses oleh bagian lain dalam mikrokontroler. Pada beberapa mikrokontroler, modul ini telah ada atau *build-in*. Jadi,

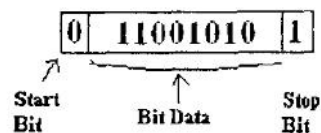
apabila modul ini diperlukan, Anda tidak perlu lagi menambah dari eksternal.

2.4.16. Komunikasi Serial (Serial Communication)

Dalam transmisi data jarak jauh, kurang efektif bila kita mentransmisinya secara paralel. Pengiriman data yang banyak akan membutuhkan jalur paralel yang banyak juga sehingga hal ini akan meningkatkan biaya yang diperlukan. Untuk itu, dalam beberapa kasus, transmisi serial lebih efektif karena hanya satu data yang ditransfer tiap satu waktu sehingga hanya dibutuhkan satu line untuk transmisinya. Dalam mikrokontroler hanya dibutuhkan 3 *line* untuk melakukan komunikasi secara serial. yaitu *line transmitter*, *line receiver*, dan *line ground*.

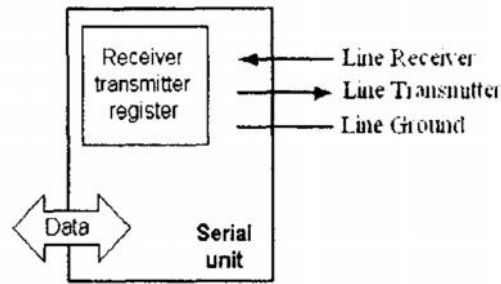
Proses transmisi data serial, umumnya diawali dengan memberikan logika 0 sebagai *Start Bit* dari transmisi bare, kemudian dilanjutkan dengan mengirimkan 8 bit data dan diakhiri dengan *Stop Bit* berupa logika 1. Sebagai contoh, bila ingin mengirim data 11001010 secara serial, maka format pengiriman satu blok data dapat dilihat pada Gambar 2.13. Jadi. pada saat bagian penerima mendeteksi logika 1 untuk pertama kali. maka logika tersebut akan dianggap sebagai awal dari penerimaan data.

Komunikasi secara serial menggunakan 2 buah jalur yang berbeda untuk transmisi dan penerimaan datanya sehingga dapat dilakukan pengiriman dan penerimaan data secara bersamaan. Sistem komunikasi ini biasanya dinamakan dengan *full duplex*. Pada komunikasi secara paralel, untuk melakukan ini dibutuhkan jalur yang banyak. Sebagai contoh, bila kita ingin mengirimkan data 8 bit secara paralel, maka dibutuhkan 8 jalur untuk mentransmisinya. Jadi, untuk komunikasi jika kita ingin membuatnya *full duplex*. maka dibutuhkan 16 buah jalur. Hal inilah yang sebelumnya telah dijelaskan bahwa komunikasi secara paralel akan membutuhkan jalur yang sangat banyak.



Gambar 2.13. Format serial bit

(sumber : Husanto & Thomas, Kupas Tuntas Mikrokontroler PIC16F84A, Penerbit Andi, 2008)



Gambar 2.14. Unit Serial

(sumber : Husanto & Thomas, Kupas Tuntas Mikrokontroler PIC16F84A, Penerbit Andi, 2008)

2.12 Teknologi GSM

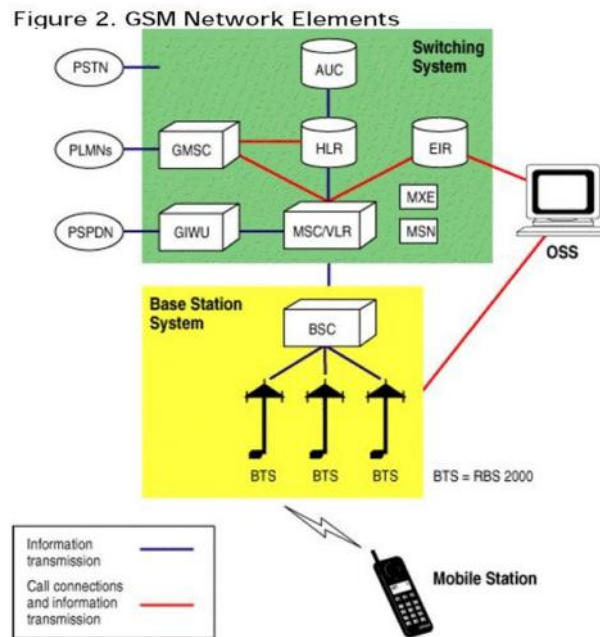
2.5.4. Pengenalan GSM

Global system for mobile communication (GSM) merupakan standar yang diterima secara global untuk komunikasi selular digital. GSM adalah nama group standarisasi yang di mapankan pada tahun 1982 untuk menghasilkan standar telepon bergerak di eropa, digunakan sebagai formula spesifikasi untuk pan-eropa sistem selular radio bergerak yang bekerja pada frekuensi 900 Mhz. Dan diperkirakan banyak negara lainnya diluar eropa akan turut menggunakan teknologi GSM.

Secara keseluruhan evolusi dari telekomunikasi selular, sistem yang beraneka telah dikembangkan tanpa menguntungkan dari spesifikasi yang standar. Ini secara langsung menghadirkan banyak masalah kompatibilitas, khususnya perkembangan teknologi radio digital. Standar GSM memfokuskan ke arah tersebut. Dari tahun 1982 sampai 1985 diskusi diselenggarakan untuk memutuskan antara membangun sistem analog atau digital. Setelah melalui berkali-kali pengujian, sistem digital di adopsi dari GSM.

2.5.5. Jaringan GSM

GSM memberikan suatu rekomendasi bukan suatu persyaratan. GSM menspesifikasikan fungsi-fungsi dan antarmuka yang diperlukan secara detail bukan mengarah ke perangkat keras yang digunakan. Alasan tersebut didasari untuk membatasi para desainer sekecil mungkin namun tetap saja memungkinkan para operator untuk membeli perangkat dari penyedia yang berbeda. Jaringan GSM dibagi menjadi tiga sistem utama yakni *switching system* (SS), *base station sistem* (BSS), dan *operation and support system* (OSS). Elemen dasar jaringan GSM di tunjukkan pada gambar 2.22.



Gambar 2.15. Elemen Jaringan GSM

(sumber : Parallila Keimena, Global System for Mobile Communication (GSM),
http://edu.eap.gr/pli/pli23/documents/Parallila_Keimena/GSM.pdf)

2.5.6. Spesifikasi GSM

Spesifikasi untuk layanan sistem *personal communication services* (PCS) yang berlainan akan merubah jaringan PCS tersebut. Daftar dibawah mendeskripsikan spesifikasi dan karakteristik GSM.

1. *Frequency Band* (rentang frekuensi)

Rentang frekuensi yang dispesikasikan untuk GSM adalah 1,850 to 1,990 Mhz (*mobile station ke base station*).

2. *Duplex Distance*

Duplex distance ialah jarak antara frekuensi *uplink* dan *downlink*. *Duplex distance* jaringan GSM adalah 80 Mhz. Satu kanal memiliki dua frekuensi, terpisah 80 Mhz.

3. *Channel Separation* (Pemisahan Kanal)

Pemisahan antara frekuensi pembawa terdekat. Di GSM, ini adalah 200 kHz.

4. *Modulation* (modulasi)

Modulasi adalah proses mengirim sinyal dengan merubah karakteristik dari frekuensi pembawa. Hal ini dapat dilakukan di GSM melalui *Gaussian Minimum Shift Keying* (GMSK).

5. *Transmission Rate* (kecepatan transmisi)

GSM adalah sistem digital dengan laju over-the-air 270 kbps.

6. *Access Method* (metode akses)

GSM memanfaatkan konsep *Time Division Multiple Access* (TDMA). TDMA adalah teknik dimana beberapa panggilan berbeda memungkinkan berbagagi pembawa yang sama. Tiap panggilan di tkitai slot waktu yang akurat.

7. *Speech Coder*

GSM menggunakan *Linear Predictive Coding* (LPC). Maksud dari LPC adalah untuk mengurangi laju bit. LPC memberikan parameter untuk filter yang menirukan vokal. Sinyal lewat melalui filter ini, meninggalkan dibelakang sinya sisa. Percakapan di encode pada 13 kbps.

2.13 Layanan SMS

2.13.1 Pengenalan SMS

Layanan SMS hanyalah sebuah fasilitas jaringan digital yang memungkinkan pengguna telepon digital untuk menerima pesan teks pada ponsel digital mereka. Setiap pesan mungkin maksimal 160 karakter. Dalam dokumen ini, kami memberikan pengenalan konsep dasar SMS, jaringan dan spesifikasi, dan alat-alat dan layanan SMS.

2.13.2 Short Message Service (SMS)

Short Message Service (SMS) adalah transmisi pesan teks singkat ke dan dari ponsel, mesin faks, dan / atau alamat IP. Pesan harus tidak lebih dari 160 karakter alfanumerik dan tidak mengandung gambar atau grafis. SMS adalah sistem pesan yang relatif sederhana yang disediakan oleh jaringan telepon seluler. Pesan SMS yang didukung oleh GSM, TDMA dan jaringan telepon seluler berbasis CDMA yang sedang digunakan. Meskipun layanan berbasis SMS telah layak selama bertahun-tahun, penetrasi ponsel terbaru dan adopsi skala besar dari layanan yang ada oleh pengguna, telah membuat layanan berbasis SMS bahkan lebih menarik bagi penyedia layanan.

Setelah pesan dikirim, itu diterima oleh *Short Message Service Center* (SMSC), yang kemudian harus langsung ke perangkat mobile yang sesuai. Untuk melakukan hal ini, SMSC mengirimkan SMS. Permintaan untuk *Home Location Register* (HLR) untuk menemukan permintaan pelanggan *roaming*. HLR, ia akan merespon SMSC dengan status pelanggan yang: 1) tidak aktif atau aktif 2) di mana pelanggan adalah *roaming*. Jika respon

adalah 'tidak aktif', maka SMSC akan menyimpan *message* selama periode waktu. Saat pelanggan mengakses *device*-nya, HLR mengirimkan notifikasi SMS ke SMSC dan SMSC akan berusaha pengiriman.

SMSC mentransfer pesan dalam Pesan Pendek Pengiriman Format *Point-to-Point* ke sistem melayani. Halaman-halaman sistem perangkat, dan jika merespon, pesan akan dikirimkan. SMSC akan menerima verifikasi bahwa pesan tersebut telah diterima oleh pengguna akhir, kemudian mengkategorikan pesan sebagai 'dikirim' dan tidak akan mencoba untuk mengirim lagi.

Meskipun layanan diaktifkan oleh WAP (*Wireless Application Protocol*) dan UMTS (*Universal Mobile Telecommunications System*) akan paling mungkin menggantikan pesan SMS sebagai media paling populer untuk aplikasi nirkabel, masih akan ada basis pengguna yang sangat besar untuk waktu yang lama. Besar bunga pasar yang berkaitan dengan WAP dan disebut *mCommerce* (*mobile commerce*) telah membuat juga SMS menarik sebagai jalur distribusi. Operator dan penyedia layanan menciptakan banyak layanan baru. Penyediaan Layanan *Wireless Application* (WASP) adalah, menarik arsitektur layanan terbaru untuk menyediakan layanan berbasis SMS.

Prinsip dasarnya adalah bahwa hanya ada satu SMSC (*SMS Center*) yang mengkodekan pesan yang akan disampaikan melalui jaringan GSM. Dasar kesulitan dalam mengembangkan layanan berbasis SMS adalah berbagai protokol yang digunakan dalam *SMS Center*. *The European Telecommunication Standards Institute* (ETSI) telah menyetujui empat protokol SMSC: SMPP (oleh Logica), CIMD (by Nokia), UCP / EMI (dengan CMG) dan SMS2000 (oleh SEMA). Semua protokol ini memiliki fungsi yang sedikit berbeda dan konversi karakter yang sebagian besar berbeda. Mendukung semua protokol ini adalah tugas yang menuntut untuk penyedia layanan. Ada beberapa gateway SMS dapat berinteraksi dengan beberapa atau semua protokol SMS. Namun, tidak ada cara standar untuk penyedia layanan untuk berinteraksi dengan SMS *gateway*. Juga, hanya beberapa gateway SMS mendukung semua protokol SMSC. Draft ini mengusulkan sebuah solusi dengan memperkenalkan sebuah antarmuka mudah adoptable ke Pusat SMS atau SMS gateway bagi penyedia jasa. Sebagian besar negara menggunakan standar GSM, Amerika Serikat adalah salah satu dari beberapa negara untuk mendukung penggunaan CDMA dan standar TDMA melalui GSM (meskipun ada jaringan GSM di seluruh Amerika Serikat). CDMA dan TDMA memungkinkan sangat terbatas kemampuan SMS.

Pesan singkat dapat dikirim dan diterima secara bersamaan dengan suara GSM, data dan panggilan faks. Hal ini dimungkinkan karena sementara suara, data dan panggilan faks mengambil alih saluran radio khusus untuk durasi panggilan, pesan singkat perjalanan atas dan di atas saluran radio menggunakan jalur sinyal. Dengan demikian, pengguna SMS jarang, jika pernah, mendapatkan sinyal sibuk atau terlibat seperti yang mereka lakukan pada saat penggunaan jaringan puncaknya.

Cara mengirimkan beberapa pesan singkat yang tersedia. SMS Rangkaian (merangkai beberapa pesan singkat bersama-sama) dan kompresi SMS (mendapatkan lebih dari 160 karakter informasi dalam pesan singkat tunggal) telah didefinisikan dan dimasukkan dalam standar GSM SMS.

Untuk menggunakan Short Message Service, pengguna perlu langganan dan perangkat keras yang relevan, khususnya:

1. Berlangganan ke jaringan telepon seluler yang mendukung SMS
2. Sebuah ponsel yang mendukung SMS.
3. Penggunaan SMS harus diaktifkan bagi pengguna. (Akses otomatis ke SMS diberikan oleh beberapa operator jaringan selular, yang lain biaya berlangganan bulanan dan memerlukan spesifik opt-in untuk menggunakan layanan ini).
4. Pengetahuan tentang bagaimana mengirim atau membaca pesan singkat dengan menggunakan model spesifik ponsel.
5. Sebuah tujuan untuk mengirimkan pesan singkat kepada, atau menerima pesan dari.

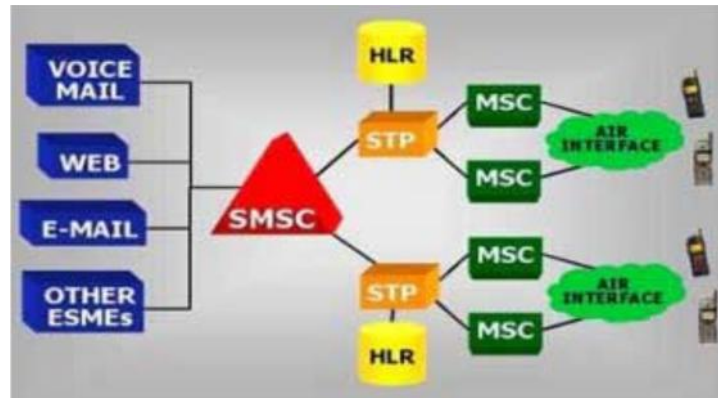
Hal ini biasanya ponsel lain tapi mungkin mesin fax, PC atau alamat internet.

Pesan SMS yang ditransfer antara ponsel melalui SMSC. SMSC adalah perangkat lunak yang berada di jaringan operator dan mengelola proses termasuk antrian pesan, penagihan pengirim dan kembali penerimaan jika diperlukan. Banyak operator sekarang menawarkan antarmuka berbasis web untuk SMSC mereka sehingga kita dapat mengirim pesan singkat ke telepon seluler dari web. Beberapa situs sekarang menawarkan SMS gratis.

Di Amerika Utara, SMS dibuat tersedia awalnya pada jaringan nirkabel digital dibangun oleh perintis awal seperti BellSouth Mobility, PrimeCo, dan Nextel, antara lain. Ini jaringan nirkabel digital didasarkan pada GSM, code division multiple access (CDMA), dan pembagian waktu multiple access (TDMA) standar.

Jaringan konsolidasi dari merger dan akuisisi telah menghasilkan jaringan nirkabel besar yang memiliki cakupan nasional atau internasional dan terkadang mendukung lebih

dari satu teknologi nirkabel. Ini kelas baru penyedia layanan menuntut produk jaringan kelas yang andal dan mudah dapat memberikan solusi yang seragam, memungkinkan kemudahan operasi dan administrasi, dan mengakomodasi kapasitas yang ada pelanggan, throughput yang pesan, pertumbuhan di masa depan, dan jasa. Pendek messaging service center (SMSC) solusi berdasarkan jaringan cerdas (IN) pendekatan sangat cocok untuk memenuhi persyaratan ini, sambil menambahkan semua manfaat dari implementasi IN.



Gambar 2.16. Arsitektur dasar jaringan IS-41

(Sumber : <http://educyclopedia.karadimov.info/library/SMS.pdf>)

Gambar 2.16 merupakan arsitektur jaringan dasar untuk IS-41 SMSC penyebaran menangani beberapa sumber masukan, termasuk sistem voice-mail (VMS), messaging berbasis Web, integrasi e-mail, dan entitas pesan singkat eksternal lainnya (ESMEs). Komunikasi dengan elemen-elemen jaringan nirkabel seperti home location register (HLR) dan mobile switching centers (MSC) dicapai melalui titik transfer sinyal (STP). SMS menyediakan mekanisme untuk mengirimkan pesan singkat ke dan dari perangkat nirkabel. Layanan ini memanfaatkan sebuah SMSC, yang bertindak sebagai sistem *token-and-forward* untuk pesan singkat. Jaringan nirkabel menyediakan mekanisme yang diperlukan untuk menemukan stasiun tujuan dan mengangkut pesan singkat antara SMSC dan stasiun nirkabel. Berbeda dengan layanan pesan teks transmisi lainnya yang ada seperti *paging* alfanumerik, unsur jasa yang dirancang untuk memberikan jaminan pengiriman pesan teks ke tujuan. Selain itu, SMS mendukung beberapa mekanisme masukan yang memungkinkan interkoneksi dengan sumber pesan yang berbeda dan tujuan.

Karakteristik yang membedakan dari layanan adalah bahwa *handset mobile* aktif dapat menerima atau mengirimkan pesan singkat setiap saat, terlepas dari apakah suara atau panggilan data berlangsung (dalam beberapa implementasi, hal ini mungkin tergantung pada MSC atau kemampuan SMSC). SMS juga menjamin pengiriman pesan

singkat oleh jaringan. Kegagalan sementara karena stasiun penerima tidak tersedia diidentifikasi, dan pesan singkat disimpan pada SMSC sampai perangkat tujuan menjadi tersedia.

SMS ditandai dengan pengiriman paket *out-of-band* dan *bandwidth* rendah transfer pesan, yang menghasilkan sarana yang sangat efisien untuk transmisi semburan data pendek. Aplikasi awal SMS difokuskan pada menghilangkan pager alfanumerik dengan mengizinkan dua arah layanan pemberitahuan tujuan umum pesan dan, terutama untuk surat suara. Sebagai teknologi dan jaringan berkembang, berbagai layanan telah diperkenalkan, termasuk e-mail, faks, dan integrasi *paging*, perbankan interaktif, layanan informasi seperti harga saham, dan integrasi dengan aplikasi berbasis internet. Aplikasi data nirkabel termasuk *download subscriber identity module* (SIM) card untuk aktivasi, debit, tujuan profil-editing, poin penjualan nirkabel (POSS), dan aplikasi bidang-layanan lain seperti membaca meter otomatis, penginderaan jauh, dan berbasis lokasi jasa. Selain itu, integrasi dengan internet memacu pengembangan *Web-messaging* dan aplikasi berbasis interaktif lainnya seperti *instant messaging*, *game*, dan *chatting*.

2.13.3 Manfaat SMS

Dalam dunia yang kompetitif saat ini, diferensiasi adalah faktor yang signifikan dalam keberhasilan penyedia layanan. Setelah layanan dasar, seperti telepon suara, dikerahkan, SMS menyediakan kendaraan yang kuat untuk diferensiasi layanan. Jika pasar memungkinkan untuk itu, SMS juga dapat merupakan sumber tambahan pendapatan bagi penyedia layanan.

Manfaat SMS ke pelanggan sekitar pusat kenyamanan, fleksibilitas, dan integrasi layanan pesan dan akses data. Dari perspektif ini, manfaat utama adalah kemampuan untuk menggunakan handset sebagai perpanjangan komputer. SMS juga menghilangkan kebutuhan untuk perangkat terpisah untuk pesan karena layanan dapat diintegrasikan ke dalam perangkat-terminal tunggal mobile wireless. Manfaat ini biasanya tergantung pada aplikasi yang ditawarkan penyedia layanan.

Minimal, manfaat SMS adalah sebagai berikut:

1. Pengiriman pemberitahuan dan peringatan
2. Pengiriman pesan Dijamin
3. *Reliable*, mekanisme komunikasi murah untuk informasi ringkas
4. Kemampuan untuk pesan layar dan panggilan kembali secara selektif

5. Peningkatan produktivitas pelanggan

Fungsionalitas yang lebih canggih memberikan manfaat pelanggan ditingkatkan sebagai berikut:

1. Pengiriman pesan ke beberapa pelanggan pada satu waktu.
2. Kemampuan untuk menerima informasi yang beragam
3. Generasi E-mail
4. Pembentukan kelompok pengguna
5. Integrasi dengan data lain dan aplikasi berbasis Internet

Manfaat SMS ke *Service Provider* adalah sebagai berikut:

1. Kemampuan untuk kenaikan rata-rata pendapatan per pengguna (karena meningkatnya jumlah panggilan pada jaringan nirkabel dan kabel dengan memanfaatkan kemampuan pemberitahuan SMS)
2. Sebuah alternatif untuk layanan paging alfanumerik, yang dapat menggantikan atau melengkapi tawaran paging.
3. Kemampuan untuk mengaktifkan akses data nirkabel untuk pengguna korporat
4. Arus pendapatan baru yang dihasilkan dari penambahan nilai tambah layanan seperti pesan e-mail, suara, faks, dan integrasi aplikasi berbasis Web, layanan pengingat, saham dan kutipan mata uang, dan jadwal penerbangan
5. Penyediaan jasa administrasi utama seperti Advice of charge, over-the-air download, dan penyediaan layanan over-the-air
6. Perlindungan sumber daya penting jaringan (seperti saluran suara), karena SMS 'penggunaan hemat dari kontrol dan saluran lalu lintas
7. Mekanisme Pemberitahuan untuk layanan baru seperti yang memanfaatkan protokol aplikasi nirkabel (WAP)

Semua manfaat ini dicapai dengan cepat, dengan biaya tambahan sederhana dan periode pengembalian singkat, yang membuat SMS investasi yang menarik bagi penyedia layanan.

Dukungan SMS pada Jaringan berbeda Sejak dimasukkan dalam standar GSM, SMS juga telah dimasukkan ke dalam banyak standar jaringan telepon selular lainnya, termasuk Nordic Mobile Telephone (NMT), Code Division Multiple Access (CDMA) dan Personal Digital Selular (PDC) di Jepang . Masing-masing standar ini mengimplementasikan SMS dengan cara yang sedikit berbeda dan panjang pesan yang berbeda-beda. Ketersediaan dan karakteristik SMS pada standar jaringan selular yang berbeda adalah:

Tabel 2.1. Tabel dukungan SMS pada sstandar jaringan *mobile*

Standar <i>Mobile Network</i>	Type	SMS Availability	Panjang Pesan	Pengembangan
GSM 900	Digital	Yes	160	Sangat luas
GSM 1800	Digital	Yes	160	Sangat luas
GSM 1900	Digital	Yes	160	Amerika Utara
TACS / ETACS	Analog	No	N/A	N/A
NMT	Analog	Yes	N/A	Eropa Timur
TDMA / D-AMPS	Digital	Yes	N/A	Amerika Utara
NAMPS	Analog	Yes	14	Amerika Utara
			Alphanumeric32 numeric	
CDMA	Digital	Yes (MT	256	Amerika Utara
PHS	Digital	Only)	N/A	Jepang
PDC	Digital	Yes	N/A	Jepang
IDEN / NEXTEL	Digital	Yes	140	Amerika Utara
		Yes		dan Selatan
TETRA / Dolphin	Digital	Yes	256	Sebagian Eropa
Globalstar	Satellite	Yes	160	Global

2.13.4 Peralatan

1. ESMS-C:

ESMS-C adalah solusi SMS Ericsson yang menawarkan fungsi pesan singkat, pemberitahuan pesan, e-mail. Selain itu, ESMS-C menawarkan fitur yang lebih canggih seperti:

- a. Layanan Informasi - Layanan memberikan informasi mengenai, misalnya berita utama, skor olahraga dan kutipan harga saham.
- b. Layanan Over-The-Air seperti aktivasi dan pemrograman.
- c. Aplikasi Telemetry seperti penjual alarm mesin dan pembacaan meter jarak jauh.
- d. Ponsel e-Commerce seperti mobile banking, pembelian tiket film.

ESMS-C mendukung GSM, TDMA, CDMA, GPRS dan jaringan. Ini telah berhasil interoperabilitas diuji baik Ericsson dan lingkungan Nokia. Kemampuan ESMS-C cocok dalam jaringan nirkabel karena telepon selular yang mampu SMS yang tersedia.

Dengan menggunakan salah satu interface yang didukung Ericsson, ESMS-C terhubung langsung dengan Mobile-layanan Switching Center (MSC). Rute ESMS-C langsung ke set tangan pengguna akhir mobile di mana pun itu, apakah jelajah atau tidak dengan berkomunikasi langsung dengan Home Location Register. Interface yang didukung termasuk:

- a. SMPP - Short Message Protocol Peer to Peer adalah standar de facto untuk pesan singkat. SMPP menyediakan sebuah antarmuka untuk berbagai aplikasi termasuk "push" dan tarik jenis layanan serta Gateway WAP dan EMS.
- b. CAPII - Akses Komputer Protokol II adalah protokol pesan singkat Ericsson yang menawarkan beberapa keuntungan dalam menciptakan aplikasi berbasis web.
- c. SMTP - Simple Mail Transfer Protocol adalah protokol e-mail yang memungkinkan ESMS-C untuk mengkonversi SMS ke e-mail dan sebaliknya.

2. SMS-JDK:

www.noctor.com telah mengembangkan JDK untuk SMS. SMS-JDK menyediakan antarmuka yang sederhana dan efektif untuk dunia pesan singkat nirkabel, menyembunyikan semua kompleksitas protokol dari pengembang aplikasi sehingga memungkinkan pengembang untuk fokus pada layanan aplikasi. Hal ini secara signifikan mengurangi pengembangan kerangka waktu, overhead pemeliharaan dan biaya. SMS-JDK adalah Java 100%.

3. Nokia PC Connectivity SDK:

Nokia PC Connectivity SDK adalah canggih dan mudah-ke-menggunakan antarmuka pemrograman untuk Nokia GSM dan telepon TDMA. SDK ini memungkinkan komunikasi dengan Nokia GSM dan telepon TDMA dan pengembangan aplikasi PC yang memanfaatkan fitur yang didukung oleh ponsel.

Nokia PC Connectivity SDK perpustakaan terdiri dari beberapa perpustakaan yang terpisah, masing-masing melakukan satu set khusus tugas yang berhubungan dengan GSM atau fungsi telepon TDMA. Perpustakaan termasuk adalah:

- a. Pengaturan Umum Perpustakaan
- b. Perpustakaan SMS
- c. Memory Phonebook Perpustakaan
- d. WAP Perpustakaan

4. SMS Gateway (www.winsms.com):

SMS Gateway adalah 32 Bit utilitas Windows yang memungkinkan Anda untuk mengirim dan menerima pesan teks dan biner "Pesan Singkat" melalui jaringan telepon GSM selular digital. Paket terdiri dari kedua aplikasi interaktif messaging (dengan kode sumber penuh), dan gateway pesan yang berdiri sendiri untuk aplikasi Windows lainnya melalui penggunaan DDE, OLE, dan Command Line Interface. SMS Gateway juga mendukung POP3 untuk transmisi pesan, dan SMTP untuk penerimaan pesan, sehingga dapat digunakan di hampir setiap lingkungan tanpa kebutuhan untuk pengembangan kustom.

2.13.5 Aplikasi SMS

SMS pada awalnya dirancang untuk mendukung pesan terbatas ukuran, kebanyakan pemberitahuan dan halaman numerik atau alfanumerik. Sementara aplikasi ini dan akan terus digunakan secara luas, ada ceruk yang lebih baru yang masih dapat memanfaatkan SMS.

Semburan data pendek berada di jantung dari banyak aplikasi yang terbatas pada dunia jaringan data dengan terminal tetap terpasang ke jaringan area lokal (LAN) atau wide area network (WAN). Namun, banyak dari aplikasi ini lebih baik dilayani jika kemampuan komunikasi data dapat ditambahkan ke mobilitas stasiun. Dengan demikian, seorang pelayan yang dapat mengisi kanan kartu kredit pelanggan di meja, setiap saat, bukannya pergi ke terminal POS tetap terletak register akan dapat membantu pelanggan dalam lebih cepat, cara yang lebih nyaman.

Juga, kemampuan untuk melacak lokasi aset bergerak seperti truk atau beban sangat berharga bagi kedua penyedia dan klien. Aplikasi ini, sekali lagi, hanya perlu untuk pertukaran sejumlah kecil informasi, seperti bujur dan lintang pada waktu saat hari, dan parameter mungkin lain seperti suhu atau kelembaban.

Aplikasi ini tidak selalu memerlukan entitas dipantau berada dalam gerakan. Persyaratan dasarnya pendek, bursty data dan lokasi yang memiliki jangkauan jaringan digital. Sebagai contoh, di lingkungan, akan lebih cepat, lebih mudah, dan lebih murah untuk mengemudikan truk dari perusahaan listrik setempat, yang menanyakan meter cerdas untuk mendapatkan bacaan mereka saat ini dan kemudian mengirimkannya melalui pesan singkat ke pusat pengolahan data untuk menghasilkan penagihan. Demikian pula, truk pengiriman bisa diperingatkan persediaan pelanggan menipis, ketika truk dekat dengan fasilitas pelanggan. Sopir truk dapat menempatkan panggilan telepon cepat kepada pelanggan untuk menawarkan pengisian waktu singkat dengan biaya rendah bagi distributor.

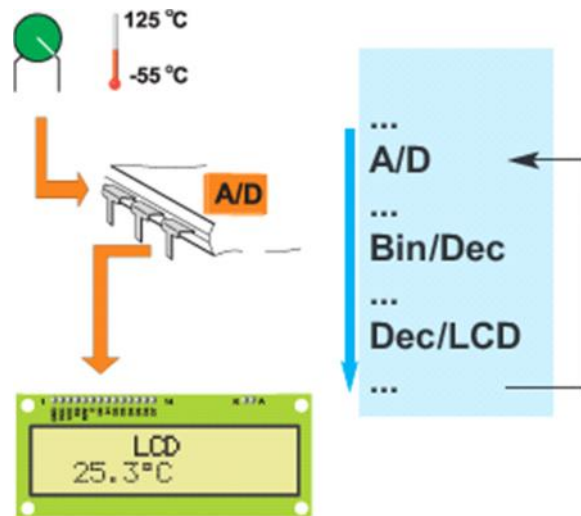
Keluarga lain aplikasi yang dapat menggunakan SMS sebagai mekanisme transportasi data perbankan. Sudah bukan rahasia lagi bahwa automated teller machine (ATM) dan transaksi internet lebih murah daripada transaksi selesai pada cabang. Transaksi Internet bahkan lebih murah dari transaksi ATM. Oleh karena itu, memungkinkan pelanggan nirkabel untuk mengecek saldo mereka, mentransfer dana antar rekening, membayar tagihan dan kartu kredit yang berharga, tidak hanya bagi pelanggan tetapi juga untuk lembaga keuangan. Aplikasi hiburan juga driver baik penggunaan SMS. Contoh dari ini adalah pertukaran

pesan singkat sederhana antara dua pihak ("SMS") atau antara beberapa peserta ("chatting"). Juga, penyampaian informasi bahwa pelanggan dapat menyesuaikan dengan gaya hidup nya merupakan proposisi menarik bagi pengguna nirkabel.

Web browsing nirkabel memungkinkan pengguna untuk mencari informasi tanpa pembatasan fisik PC. Mahasiswa tentu menghargai tidak harus pergi ke lab komputer atau asrama mereka untuk memeriksa e-mail atau mencari tahu apa buku yang diperlukan untuk semester yang akan dimulai.

E-mail terus menjadi jauh aplikasi data nirkabel yang paling banyak digunakan. Namun, handset yang berkembang cepat dan termasuk fungsionalitas lebih dan lebih yang mendukung aplikasi baru pada saat yang sama yang ramah pengguna meningkat. Mungkin keberhasilan besar berikutnya di luar Web nirkabel akan belanja internet dan aplikasi e-commerce lainnya seperti kupon elektronik, iklan, dll. Potensi aplikasi sangat besar, dan kebutuhan-kebutuhan baru muncul untuk muncul terus-menerus, menuntut solusi yang mungkin melalui SMS.

2.14 Dasar-Dasar Pemrograman Bahasa C



Gambar 2.17. Ide pokok penulisan program dalam bahasa C

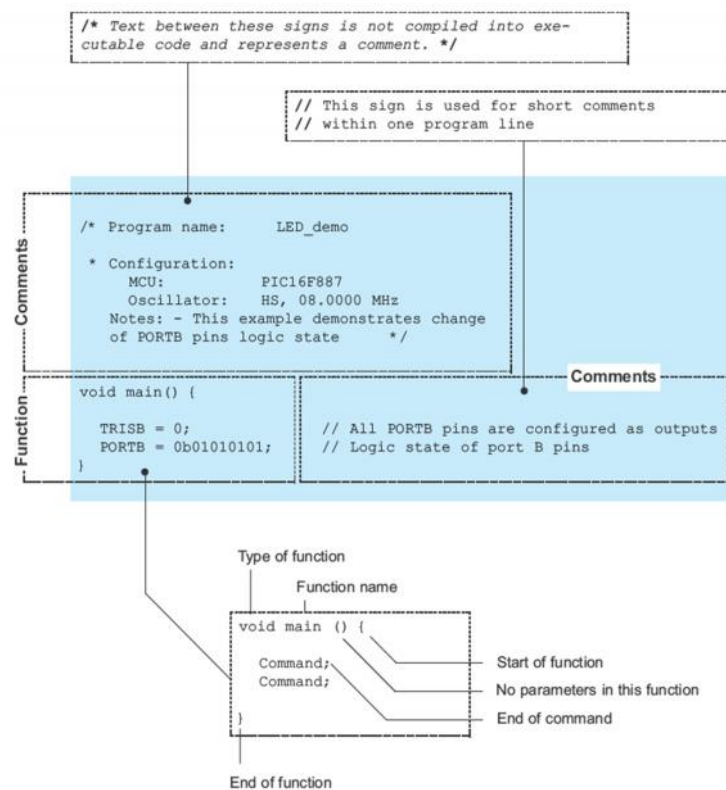
(Sumber : <http://www.mikroe.com/chapters/view/15/chapter-2-programming-microcontrollers>)

Ide utama dari menulis program dalam bahasa C adalah untuk memecahkan masalah yang lebih besar menjadi bagian-bagian yang lebih kecil. Misalkan perlu untuk menulis sebuah program untuk mikrokontroler yang akan mengukur hasil suhu dan menunjukkan pada layar LCD. Proses pengukuran dilakukan oleh sensor yang mengubah suhu menjadi

tegangan. Mikrokontroler menggunakan konverter A / D untuk mengubah ini (nilai analog) tegangan ke nomor (nilai digital) yang kemudian dikirim ke layar LCD melalui beberapa konduktor. Dengan demikian, program ini dibagi dalam empat bagian yang Anda harus melalui sesuai urutan sebagai berikut:

1. Aktifkan dan mengatur built-in A / D converter;
2. Mengukur nilai analog;
3. Menghitung suhu, dan
4. Mengirim data dalam bentuk yang tepat untuk layar LCD.

Seperti yang terlihat, bahasa pemrograman tinggi seperti C memungkinkan Anda untuk memecahkan masalah ini dengan mudah dengan menulis empat fungsi yang akan dieksekusi siklis dan berulang-ulang. Gambar di bawah ini menggambarkan struktur dari sebuah program sederhana, menunjukkan bagian-bagian program tersebut.



Gambar 2.18. Ilustrasi singkat struktur program

(Sumber : <http://www.mikroe.com/chapters/view/15/chapter-2-programming-microcontrollers>)

2.7.12. Komentar

Komentar adalah bagian dari program yang digunakan untuk menjelaskan pengoperasian program atau memberikan informasi lebih lanjut tentang hal itu. Komentar

diabaikan dan tidak dikompilasi ke dalam kode dieksekusi oleh compiler. Sederhananya, compiler dapat mengenali karakter khusus yang digunakan untuk menunjuk di mana memulai dan mengakhiri komentar dan benar-benar mengabaikan teks yang berada diantara penunjuk komentar selama kompilasi. Ada dua jenis karakter tersebut. Satu menunjuk komentar panjang memperpanjang beberapa baris program, sementara yang lain menunjuk komentar singkat mengambil satu baris. Meskipun komentar tidak dapat mempengaruhi pelaksanaan program, mereka sama pentingnya seperti halnya bagian lain dari program tersebut. Sebuah program yang ditulis selalu dapat ditingkatkan, dimodifikasi, upgrade, dan disederhanakan. Hal ini hampir selalu dilakukan. Tanpa komentar, bahkan mencoba memahami program sederhana adalah buang-buang waktu.

2.7.13. Tipe data di bahasa C

Ada beberapa jenis data yang dapat digunakan dalam bahasa pemrograman C. Sebuah tabel di bawah ini menunjukkan rentang nilai yang dapat memiliki data tersebut bila digunakan dalam bentuk dasar mereka.

Tabel 2.2. Tipe Data pada bahasa C

Tipe Data	Deskripsi	Ukuran (Jumlah bit)	Rentang Nilai
Char	Character	8	0 to 255
Int	Integer	16	-32768 to 32767
Float	Floating point	32	$\pm 1.17549435082 \cdot 10^{-38}$ to $\pm 6.80564774407 \cdot 10^{38}$
Double	Double precision floating point	32	from $\pm 1.17549435082 \cdot 10^{-38}$ to $\pm 6.80564774407 \cdot 10^{38}$

Dengan menambahkan awalan (qualificator) untuk semua jenis data, berbagai perubahan yang mungkin nilai-nilai serta jumlah byte memori yang diperlukan.

Tabel 2.3. Tipe data dengan prefix

Tipe Data	Tipe Data dengan Prefix	Ukuran (Jumlah bit)	Rentang
Char	signed char	8	-128 to 128
Int	unsigned int	16	0 to 65535

	short int	8	0 to 255
	signed short int	8	-128 to 127
	long int	32	0 to 4294967295
	signed long int	32	-2147483648 to 2147483647

2.7.14. Variabel

Setiap angka mengubah nilainya selama operasi program yang disebut variabel. Sederhananya, jika program menambahkan dua angka (angka1 dan number2), maka perlu memiliki nilai untuk mewakili apa yang kita miliki dalam kehidupan sehari-hari yang disebut jumlah. Dalam hal ini number1, number2 dan jumlah variabel.

1. Mendeklarasikan Variabel

- a. Nama variabel dapat mencakup salah satu karakter abjad AZ (az), 0-9 digit dan karakter garis bawah '_'. Compiler adalah *case sensitive* dan membedakan antara modal dan huruf kecil. Fungsi dan nama-nama variabel biasanya mengandung karakter huruf kecil, sedangkan nama konstanta berisi karakter huruf besar.
- b. Nama-nama variabel tidak boleh dimulai dengan digit.
- c. Beberapa nama tidak dapat digunakan sebagai nama variabel sebagaimana telah digunakan oleh compiler itu sendiri. Nama tersebut disebut kata kunci. Sebagai contoh, compiler MikroC mengakui secara total dari 33 kata. Perhatikan tabel 3.3 berikut.

Tabel 2.4. Daftar Variabel dalam program MikroC

mikroC – keywords				
Absolute	data	If	return	typedef
Asm	default	Inline	rx	typeid
At	delete	Int	sfr	typename
Auto	do	Io	short	union
Bit	double	Long	signed	unsigned
Bool	else	Mutable	sizeof	Using
Break	enum	Namespac e	static	virtual
Case	explicit	Operator	struct	Void
Catch	extern	Org	switch	volatile

Char	false	Pascal	template	while
Class	float	Private	this	
Code	for	Protected	throw	
Const	friend	Public	true	
Continue	goto	Register	try	

2. Pointer

Pointer adalah tipe khusus dari variabel memegang alamat variabel karakter. Dengan kata lain, pointer 'poin untuk' variabel lain. Hal ini dinyatakan sebagai berikut:

```
type_of_variable * pointer_name;
```

Untuk menetapkan alamat dari variabel ke pointer, perlu untuk menggunakan karakter '=' dan menulis nama variabel diawali dengan karakter '&'. Dalam contoh berikut, pointer 'multiplex' dideklarasikan dan ditugaskan pada alamat keluar pertama dari display delapan LED:

```
unsigned int *multiplex; // Deklarasi nama dan tipe pointer multiplex
multiplex = &display1; // Pointer multiplex diberikan alamat
                // variable display1
```

Untuk mengubah nilai dari variabel menunjuk, cukup dengan untuk menulis karakter '*' di depan pointer dan menetapkan nilai baru.

```
*multiplex = 6; // Variabel display1 diberikan nilai 6
```

Demikian pula, dalam rangka untuk membaca nilai dari variabel menunjuk, cukup dengan menulis:

```
temp = *multiplex; // Nilai variabel display1 di-copy ke temp
```

3. Mengubah bit individual

Ada beberapa cara untuk mengubah hanya satu bit dari variabel. Yang paling sederhana adalah untuk menentukan nama register, posisi bit atau nama dan keadaan yang diinginkan:

```
(PORTD.F3 = 0) ; // Clear bit RD3
```

...

```
(PORTC.RELAY = 1) ; // Set bit output PORTC (sebelumnya bernama RELAY)
```

```
// RELAY harus didefinisikan sebagai konstanta
```

4. Deklarasi

Setiap variabel harus dideklarasikan sebelum digunakan untuk pertama kalinya dalam program. Karena variabel yang disimpan dalam memori RAM, perlu untuk cadangan ruang bagi mereka (satu byte, dua byte atau lebih). Anda tahu apa jenis data yang Anda tulis atau harapkan sebagai hasil dari operasi, sementara compiler tidak tahu itu. Jangan lupa, penawaran program dengan variabel yang Anda tugaskan seperti nama gerbang, jumlah, minimum dll. Compiler mengakui mereka sebagai register memori RAM. Jenis variabel biasanya diberikan pada awal program.

```
unsigned int gate1; // Deklarasi nama dan tipe variabel gate1
```

Terlepas dari nama dan jenis, variabel ini biasanya diberikan nilai awal pada awal program juga. Ini bukan langkah 'harus-lakukan', tapi tentang kebiasaan yang baik. Dalam kasus ini, terlihat sebagai berikut:

```
unsigned int gate1; // Deklarasi nama dan tipe variabel
```

```
signed int start, sum; // Deklarasi tipe dan nama dua variabel lain
```

```
gate1 = 20; // Memberikan variabel gate1 sebuah nilai inisial
```

Proses untuk menempatkan nilai awal dan menyatakan jenis dapat dilakukan dalam satu langkah:

```
unsigned int Gate1 = 20; // Deklarasi jenis, nama dan nilai dari variabel
```

Jika ada beberapa variabel yang diberi nilai awal yang sama, proses dapat disederhanakan bahkan:

```
unsigned int gate1=gate2=gate3=20;
```

```
signed int start=sm=0;
```

Tipe variabel ini tidak disertai oleh '+' atau *sign* secara default ". Sebagai contoh, char dapat ditulis bukan *signed char* (variabel adalah byte ditandatangani). Dalam hal ini compiler menganggap nilai-nilai positif variabel. Jika Anda kebetulan lupa untuk mendeklarasikan tipe variabel, compiler secara otomatis akan menganggapnya sebagai *signed integer*. Ini berarti bahwa seperti variabel akan menempati dua byte memori dan memiliki nilai-nilai dalam kisaran -32.768-32.767.

2.7.15. Konstanta

Sebuah konstanta adalah nomor atau karakter yang memiliki nilai tetap yang tidak dapat diubah selama eksekusi program. Tidak seperti variabel, konstanta disimpan dalam memori program flash mikrokontroler untuk tujuan menghemat ruang berharga RAM. Compiler mengakui mereka dengan nama mereka dan awalan const.

1. Konstanta Integer

Konstanta integer dapat desimal, heksadesimal, oktal atau biner. Compiler mengakui format mereka ditambahkan atas dasar awalan. Jika nomor memiliki awalan tidak, itu dianggap desimal secara default. Jenis konstanta secara otomatis diakui oleh ukurannya. Dalam contoh berikut, konstan MINIMUM akan secara otomatis dianggap sebagai *signed integer* dan disimpan dalam dua byte memori Flash (16 bit):

Tabel 2.5. Konstanta Integer

Format	Prefix	Example
Decimal		const MAX = 100
Hexadecimal	0x or 0X	const MAX = 0xFF
Octal	0	const MAX = 016
Binary	0b or 0B	const MAX = 0b11011101

```
const MINIMUM = -100; // Deklarasi constant MINIMUM
```

2. Konstanta Floating Point

Konstanta floating point terdiri dari bagian bilangan bulat, titik, bagian fraksional dan e opsional atau E diikuti oleh eksponen *signed integer*.

```
const T_MAX = 32,60; T_MAX // Deklarasi temperatur  
const T_MAX = 3.260E1; // Deklarasi T_MAX konstan yang sama
```

Dalam kedua contoh diatas, sebuah T_MAX bernama konstan dinyatakan memiliki nilai pecahan 32,60. Hal ini memungkinkan program untuk membandingkan suhu diukur dengan konstanta bermakna bukan angka yang mewakilinya

3. Konstanta Karakter (Karakter ASCII)

Sebuah konstanta karakter adalah karakter tertutup dalam tanda kutip tunggal. Dalam contoh berikut, sebuah I_CLASS bernama konstan dinyatakan sebagai karakter A, sementara II_CLASS bernama konstan dinyatakan sebagai karakter B.

```
const I_CLASS = 'A'; // Deklarasi I_CLASS konstan  
const II_CLASS = 'B'; // Deklarasi II_CLASS konstan
```

Ketika dengan cara seperti ini, pelaksanaan perintah mengirimkan I_CLASS dan konstanta II_CLASS ke layar LCD, akan menyebabkan masing-masing karakter A dan B yang akan ditampilkan.

4. Konstanta String

Yang terdiri konstan dari urutan karakter disebut string. Konstanta string yang tertutup dalam tanda kutip ganda.

```
const Message_1 = "Tekan tombol START"; // Pesan 1 untuk LCD  
const Message_2 = "Tekan tombol KANAN"; // Pesan 2 untuk LCD  
const Message_3 = "Tekan tombol KIRI"; // 3 Pesan untuk LCD
```

Dalam contoh ini, mengirimkan Message_1 konstan untuk sebuah layar LCD akan menyebabkan 'tekan tombol START' pesan yang akan ditampilkan.

5. Konstanta Enumerasi

Konstanta enumerasi adalah jenis khusus dari konstanta integer yang membuat program yang lebih komprehensif dan lebih mudah untuk mengikuti dengan menetapkan elemen nomor urut. Dalam contoh berikut, elemen pertama dalam kurung keriting secara otomatis diberi nilai 0, yang kedua diberikan nilai 1, yang ketiga nilai 2 dll

```
enum {MOTOR ATAS, BAWAH, KIRI, KANAN}; // Deklarasi konstan MOTOR
```

Pada setiap terjadinya kata 'KIRI', 'KANAN', 'ATAS' dan 'BAWAH' dalam program, compiler akan menggantinya dengan nomor yang sesuai (0-3). Konkretnya, jika port B pin 0, 1, 2 dan 3 terhubung ke motor yang membuat sesuatu berjalan atas, bawah, kiri dan kanan, perintah untuk menjalankan 'KANAN' motor terhubung ke bit 3 dari port B terlihat sebagai berikut:

```
PORTB.RIGHT = 1; // mengatur bit PORTB 3 terhubung ke 'KANAN' motor
```

2.7.16. Operator, Operasi Dan Ungkapan

Operator adalah simbol yang menunjukkan aritmatika tertentu, logika atau beberapa operasi lain. Ada lebih dari 40 operasi yang tersedia dalam bahasa C, tapi umumnya hanya 10-15 dari mereka yang digunakan dalam praktek. Setiap operasi dilakukan pada satu operand atau lebih dapat menjadi variabel atau konstanta. Selain itu, setiap operasi fitur eksekusi prioritas dan juga diasosiasikan.

1. Operator Aritmatika

Operator aritmatika yang digunakan dalam operasi aritmatika dan selalu memberikan hasil positif. Tidak seperti operasi unary yang dilakukan pada satu operan, operasi biner dilakukan pada dua operand. Dengan kata lain, dua nomor yang diperlukan untuk melaksanakan operasi biner. Sebagai contoh: $a + b$ atau a/b .

Tabel 2.6. Operator Aritmatika

Operator	Operasi
+	Addition
-	Subtraction
*	Multiplication

/	Division
%	Reminder

2. Operator Tugas

Ada dua jenis tugas dalam bahasa C :

- Operator Sederhana menempatkan nilai ke variabel menggunakan umum '=' karakter. Misalnya: $a = 8$
- Tugas majemuk spesifik untuk bahasa C dan terdiri dari dua karakter seperti yang ditunjukkan dalam tabel. Ekspresi dapat ditulis dengan cara yang berbeda juga, tapi yang satu ini menyediakan kode mesin yang lebih efisien.

Tabel 2.7. operator Tugas

Operator	Example	
	Expression	Equivalent
+=	$a += 8$	$a = a + 8$
-=	$a -= 8$	$a = a - 8$
*=	$a *= 8$	$a = a * 8$
/=	$a /= 8$	$a = a / 8$
%=	$a \% = 8$	$a = a \% 8$

3. Operator Increment dan decrement

Operasi kenaikan dan penurunan sebesar 1 dilambangkan dengan '+' dan '-'. Karakter ini dapat mendahului atau mengikuti variabel. Dalam kasus pertama ($++x$), variabel x akan menjadi yang pertama bertambah dengan 1, kemudian digunakan dalam ekspresi. Jika tidak, variabel akan menjadi yang pertama digunakan dalam ekspresi, kemudian bertambah dengan 1. Hal yang sama berlaku untuk operasi pengurangan.

Tabel 2.8. Operator Increment dan Decrement

Operator	Example	Description
++	++a	Variable "a" is incremented by 1
	a++	
--	--b	Variable "b" is decremented by 1
	b--	

4. Operator Relasional

Operator relasional yang digunakan dalam perbandingan untuk tujuan membandingkan dua variabel yang dapat bilangan bulat (int) atau angka floating point (float). Jika ekspresi mengevaluasi ke true, 1 yang dikembalikan. Jika tidak, 0 yang dikembalikan. Ini digunakan dalam ekspresi seperti 'jika ungkapan benar maka ...'

Tabel 2.9. Operator Relasional

Operator	Meaning	Example	Truth condition
>	is greater than	$b > a$	if b is greater than a
>=	is greater than or equal to	$a >= 5$	If a is greater than or equal to 5
<	is less than	$a < b$	if a Is less than b
<=	is less than or equal to	$a <= b$	if a Is less than or equal to b
==	is equal to	$a == 6$	if a Is equal to 6
!=	is not equal to	$a != b$	if a Is not equal to b

5. Operator Logika

Ada tiga jenis operasi logika dalam bahasa C: logika AND, logika OR dan negasi (NOT). Demi kejelasan, logika negara dalam tabel di bawah ini direpresentasikan sebagai logika nol (0 = false) dan logika satu (1 = true). Operator logika kembali benar (logika 1) jika ekspresi bernilai non-nol, dan palsu (logika 0) jika ekspresi bernilai nol. Hal ini sangat penting karena operasi logika yang umum digunakan pada ekspresi, bukan pada variabel tunggal (nomor) dalam program. Oleh karena itu, operasi logika merujuk kepada kebenaran dari ekspresi keseluruhan.

Sebagai contoh: $1 \ \&\& \ 0$ adalah sama dengan (ekspresi benar) $\&\&$ (ekspresi palsu) Hasilnya adalah 0, yaitu - Salah dalam kedua kasus.

Tabel 2.10. Operator Logika

Operator	Logical AND		
&&	Operand1	Operand2	Result
	0	0	0
	0	1	0
	1	0	0
	1	1	1
Operator	Logical OR		
	Operand1	Operand2	Result

	0	0	0
	0	1	1
	1	0	1
	1	1	1
Operator	Logical NOT		
!	Operand1	Result	
	0	1	
	1	0	

6. Operator Bitwise

Tidak seperti operasi logika yang dilakukan pada variabel, operasi bitwise dilakukan pada bit tunggal dalam operan. Operator bitwise digunakan untuk memodifikasi bit dari variabel. Mereka terdaftar dalam tabel di bawah ini:

Tabel 2.11. Operator Bitwise

Operand	Meaning	Example	Result	
~	Bitwise complement	a = ~b	b = 5	a = -5
<<	Shift left	a = b << 2	b = 11110011	a = 11001100
>>	Shift right	a = b >> 2	b = 11110011	a = 00011110
&	Bitwise AND	c = a & b	a = 11100011 b = 11001100	c = 11000000
	Bitwise OR	c = a b	a = 11100011 b = 11001100	c = 11101111
^	Bitwise EXOR	c = a ^ b	a = 11100011 b = 11001100	c = 00101111

7. Cara Penggunaan Operator

- a. Kecuali untuk operator penugasan, dua operator tidak harus ditulis di samping satu sama lain.

x *% 12; ekspresi // tersebut akan menghasilkan kesalahan

- b. Operator dikelompokkan bersama-sama menggunakan tanda kurung mirip dengan ekspresi aritmatika. Ekspresi tertutup dalam kurung dihitung terlebih dahulu. Jika perlu, beberapa (nested) kurung dapat digunakan.
- c. Setiap operator memiliki prioritas dan asosiativitas seperti yang ditunjukkan dalam tabel.

Tabel 2.12. Prioritas dan asosiativitas

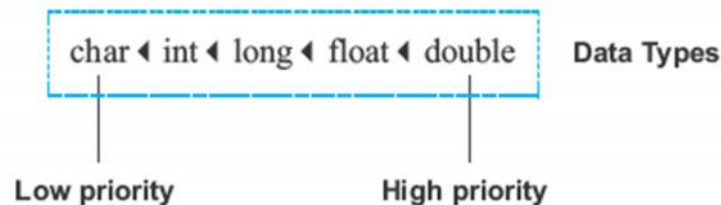
Priority	Operators	Associativity
High	() [] -> .	from left to right
	! ~ ++ -- +(unary) -(unary) *Pointer &Pointer	from right to left
	* / %	from left to right
	+ -	from left to right
	< >	from left to right
	< <= > >=	from left to right
	== !=	from left to right
	&	from left to right
	^	from left to right
		from left to right
	&&	from left to right
		from right to left
	?:	from right to left
Low	= += -= *= /= /= &= ^= = <= >=	from left to right

2.7.17. Konversi Tipe Data

Jenis data utama diletakkan dalam urutan hirarkis sebagai berikut: Jika dua operan dari tipe yang berbeda yang digunakan dalam operasi aritmatika, jenis prioritas operan yang lebih rendah secara otomatis dikonversi menjadi tipe prioritas yang lebih tinggi operan. Dalam ekspresi bebas dari tugas operasi, hasilnya diperoleh dengan cara berikut:

- a. Jika operan prioritas tertinggi adalah tipe ganda, maka jenis semua operand lainnya dalam ekspresi serta hasilnya secara otomatis dikonversi menjadi tipe double.

- b. Jika operan prioritas tertinggi adalah tipe lama, maka jenis semua operand lainnya dalam ekspresi serta hasilnya secara otomatis dikonversi menjadi tipe lama.
- c. Jika operan dari tipe lama atau char, maka jenis semua operand lainnya dalam ekspresi serta hasilnya secara otomatis dikonversi menjadi tipe int.



Gambar 2.19. Konversi tipe data

Konversi otomatis juga dilakukan dalam operasi penugasan. Hasil hak ekspresi dari operator penugasan selalu dikonversi menjadi jenis variabel kiri dari operator. Jika hasilnya lebih tinggi dari peringkat jenis, maka nilai akan dipotong atau dibulatkan dalam rangka untuk mencocokkan jenis variabel. Ketika mengkonversi data nyata ke integer, nomor-nomor berikut titik desimal selalu terpotong.

```
int x; x // Variabel dinyatakan sebagai int bilangan bulat
x = 3, x // Variabel diberi nilai 3
x + = 3.14; // Nomor PI (3.14) ditambahkan ke variabel x dengan melakukan
//Penugasan operasi
/* Hasil penambahan adalah 6 bukannya 6,14 diharapkan. Untuk memperoleh diharapkan
hasil tanpa truncating nomor-nomor berikut desimal titik, penambahan umum harus
dilakukan (x 3,14),. */
```

2.7.18. Operator Bersyarat

Sebuah kondisi adalah bahan umum dari program. Ketika bertemu, perlu untuk melakukan satu dari beberapa operasi. Dengan kata lain 'Jika kondisi ini terpenuhi (...), lakukan (...). Jika tidak, jika kondisi tidak terpenuhi, lakukan (...) '. Operan kondisional jika-lain dan beralih digunakan dalam operasi kondisional.

1. Operator Bersyarat If-Else

Operator kondisional dapat muncul dalam dua bentuk - seperti jika dan jika-lain operator. Berikut adalah contoh dari operator jika:

if (ekspresi) operasi;

Jika hasil ekspresi tertutup dalam kurung tidak 0 (benar), operasi dilakukan dan hasil program dengan eksekusi. Jika hasil dari ekspresi adalah 0 (false), operasi tidak dilakukan dan program segera melanjutkan dengan eksekusi. Seperti disebutkan, bentuk lainnya menggabungkan keduanya jika dan lain operator:

if (ekspresi) operation1 lain operation2;

Jika hasil dari ekspresi tidak 0 (benar), operation1 dilakukan, jika tidak operation2 dilakukan. Setelah melakukan operasi baik, program ini berlangsung dengan eksekusi. Sintaks dari pernyataan jika-lain adalah:

```
if (ekspresi)
operation1
else
operation2
```

Jika salah satu atau operation1 operation2 adalah senyawa, sekelompok operasi ini terdiri dari harus diapit dalam tanda kurung keriting. Sebagai contoh:

```
if (ekspresi) {
... //
... // Operation1
} ... //
Else
operation2
```

Operator jika-lain dapat ditulis menggunakan operator kondisional ':'? Seperti pada contoh di bawah ini:

(Expression1)? expression2: expression3

Jika `expression1` tidak 0 (benar), hasil dari ekspresi keseluruhan akan sama dengan hasil yang diperoleh dari `expression2`. Jika tidak, jika `expression1` adalah 0 (false), hasil dari ekspresi keseluruhan akan sama dengan hasil yang diperoleh dari `expression3`.

```
maksimum = (a > b) ? a : b maksimum // Variabel ini diberi nilai
// Lebih besar variabel (a atau b)
```

2. Operasi Switch

Berbeda dengan pernyataan jika-lain yang membuat pilihan antara dua opsi dalam program, operator switch memungkinkan Anda untuk memilih antara beberapa operasi. Sintaks dari pernyataan switch adalah:

```
switch (selector) // Selector adalah tipe char atau int
{
  case constant1:
    operation1 // Kelompok operator dijalankan jika
    ... // Pemilih dan constant1 adalah sama
    break;
  case constant2:
    operation2 // Kelompok operator dijalankan jika
    ... // Pemilih dan constant2 adalah sama
    break;
  ...
  default:
    expected_operation // Kelompok operator dijalankan jika tidak ada
    ... // Konstan sama dengan pemilih
    break;
}
```

Operasi switch dieksekusi dengan cara berikut: pemilih dijalankan pertama dan dibandingkan dengan `constant1`. Jika pertandingan ditemukan, pernyataan dalam blok kasus dijalankan sampai kata kunci istirahat atau akhir operasi saklar ditemui. Jika

pertandingan tidak ditemukan, pemilih lebih lanjut dibandingkan dengan constant2 dan jika cocok ditemukan, pernyataan dalam blok kasus dijalankan sampai kata kunci istirahat yang dihadapi dan sebagainya. Jika pemilih tidak dapat ditemukan pada konstan, operasi berikut operator default yang akan dieksekusi.

Hal ini juga memungkinkan untuk membandingkan ekspresi dengan sekelompok konstanta. Jika salah satu dari mereka cocok, operasi yang sesuai akan dieksekusi:

```
switch (number) // nomor mewakili satu hari dalam seminggu. Sekarang
    // Diperlukan untuk menentukan apakah itu adalah minggu-a
{ // Hari atau tidak.
    case1: Case2: case3: case4: case5: LCD_message = 'Weekday'; istirahat;
    case6: case7: LCD_message = 'Weekend'; break;
    default:
        LCD_message_1 = 'Pilih satu hari dalam seminggu'; istirahat;
}
```

2.7.19. Program Loop

Hal ini sering diperlukan untuk mengulang operasi tertentu selama beberapa kali dalam program. Satu set perintah yang berulang-ulang disebut loop program. Berapa kali itu akan dieksekusi, yaitu berapa lama program akan tinggal di loop, tergantung pada kondisi untuk meninggalkan loop.

1. While LOOP

While loop terlihat sebagai berikut:

```
while(expression){
    commands
    ...
}
```

Perintah-perintah yang dieksekusi berulang kali (program tetap dalam loop) sampai ekspresi menjadi palsu. Jika ekspresi adalah palsu pada masuk ke loop, maka loop tidak akan dieksekusi dan program akan melanjutkan dari akhir loop sementara. Jenis khusus dari loop program adalah lingkaran tak berujung. Hal ini terbentuk jika

kondisi tetap tidak berubah dalam loop. Eksekusi sederhana dalam kasus ini sebagai hasil dalam kurung adalah selalu benar ($1 = 1$), yang berarti bahwa remans program dalam loop yang sama:

```
while(1){  
    ... // Expressions enclosed within curly brackets will be  
    ... // endlessly executed (endless loop).  
}
```

2. For LOOP

For loop terlihat sebagai berikut:

```
for(initial_expression; condition_expression; change_expression) {  
    operations  
    ...  
}
```

Pelaksanaan urutan program tersebut adalah mirip dengan while loop, kecuali bahwa dalam kasus ini proses setting nilai awal (inisialisasi) dilakukan dalam deklarasi. Initial_expression set variabel awal loop, yang selanjutnya dibandingkan dengan condition_expression sebelum memasuki loop. Operasi dalam loop itu dilakukan berulang kali dan setelah setiap iterasi nilai ekspresi berubah. Iterasi berlanjut sampai condition_expression menjadi palsu.

```
for(k=1; k<5; k++) // Increase variable k 5 times (from 1 to 5) and  
    operation // repeat expression operation every time  
    ...
```

Operasi yang akan dilakukan lima kali. Setelah itu, akan divalidasi dengan memeriksa bahwa ekspresi $k < 5$ adalah palsu (setelah 5 iterasi $k = 5$) dan program akan keluar untuk loop.

Do

Operation

```
while (check_condition);
```

Dalam hal ini, operasi dijalankan minimal sekali terlepas dari apakah kondisi benar atau salah sebagai ekspresi `check_condition` dieksekusi pada akhir loop. Jika hasilnya tidak 0 (benar), prosedur diulang. Dalam contoh berikut, program ini masih dalam do-while loop sampai variabel mencapai 1E06 (satu juta iterasi).

```
a = 0; // Set initial value
```

```
do
```

```
a = a+1 // Operation in progress
```

```
while (a <= 1E06); // Check condition
```

2.7.20. Menulis Kode Dalam Bahasa Assembly

Kadang-kadang proses menulis program dalam bahasa C memerlukan bagian dari kode yang akan ditulis dalam bahasa assembly. Hal ini memungkinkan bagian-bagian rumit dari program yang akan dijalankan dengan cara yang tepat didefinisikan untuk periode yang tepat waktu. Misalnya, jika diperlukan untuk memiliki pulsa sangat pendek (beberapa mikrodetik) muncul secara periodik pada pin mikrokontroler. Dalam kasus tersebut dan yang sejenis, solusi yang paling sederhana adalah dengan menggunakan kode assembly untuk bagian dari program pengendalian durasi pulsa. Satu atau lebih instruksi perakitan dimasukkan dalam program yang ditulis dalam bahasa C menggunakan perintah `asm`:

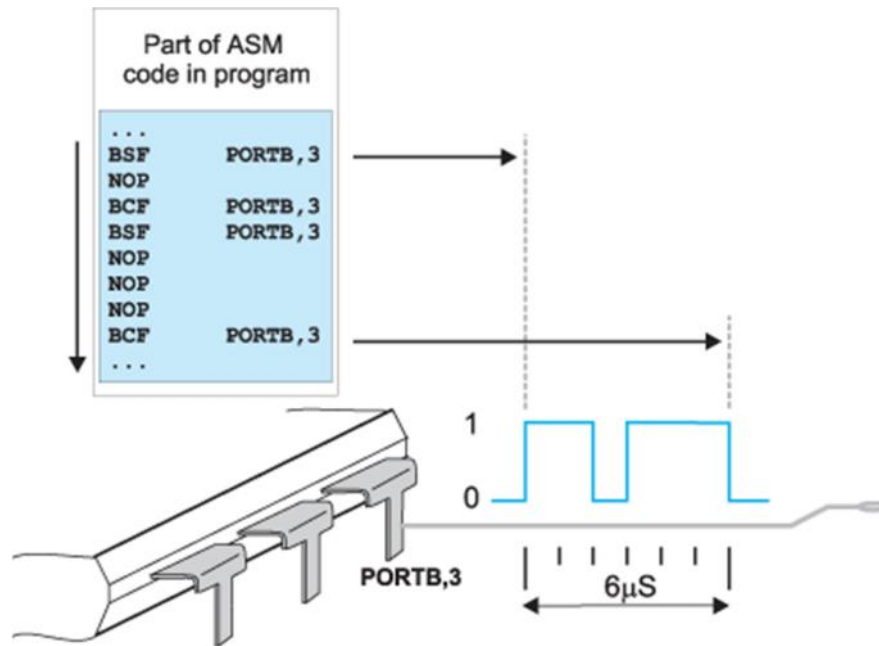
```
asm
```

```
{
```

```
Instruksi bahasa assembly
```

```
...
```

```
}
```



Gambar 2.20. Menulis Kode Dalam Bahasa Assembly

Kode ditulis dalam bahasa assembly dapat menggunakan konstanta dan variabel yang sebelumnya didefinisikan dalam bahasa C. Tentu saja, sebagai program secara keseluruhan ditulis dalam bahasa C, aturan daripadanya diterapkan ketika mendeklarasikan konstanta dan variabel tersebut.

unsigned char maximum = 100; // Deklarasi variabel: maksimum = 100

asm

{ // Mulai dari kode assembly

 MOVWF maximum, W // W = maksimum = 100

 ...

} // Akhir dari kode assembly

2.7.21. Array

Sekelompok variabel dari jenis yang sama disebut array. Elemen array disebut komponen, sedangkan tipe mereka disebut jenis utama. Array dideklarasikan dengan menetapkan nama, jenis dan jumlah elemen itu akan terdiri dari:

component_type array_name [number_of_components];

Seperti definisi rumit untuk sesuatu yang begitu sederhana, bukan? Array dapat dianggap sebagai daftar lebih pendek atau lebih variabel dari jenis yang sama di mana masing-masing diberikan sebuah nomor urut (penomoran selalu dimulai dari nol). Seperti array sering disebut vektor. Gambar di bawah menunjukkan sebuah rak array bernama yang terdiri dari 100 elemen.

Tabel 2.13. Array

Array "shelf"	Elements of array	Contents of element
7	shelf[0]	7
23	shelf[1]	23
34	shelf[2]	34
0	shelf[3]	0
0	shelf[4]	0
12	shelf[5]	12
9	shelf[6]	9
...
...
23	shelf [99]	23

Dalam hal ini, isi dari variabel (elemen dari array) mewakili sejumlah produk rak berisi. Elemen diakses oleh pengindeksan, yakni dengan menentukan nomor urut mereka (indeks):

```
shelf[4] = 12; // 12 item adalah 'ditempatkan' di rak [4]
temp = shelf [1]; // rak Variabel [1] adalah disalin ke
// variabel temp
```

Elemen dapat ditugaskan isi selama deklarasi array. Dalam contoh berikut, kalender array bernama dideklarasikan dan setiap elemen diberikan jumlah tertentu hari:

```
unsigned char calendar [12] = {31,28,31,30,31,30,31,31,30,31,30,31};
```

1. Array Dua Dimensi

Terlepas dari satu-dimensi array yang dapat dianggap sebagai daftar, ada juga array multidimensi dalam bahasa C. Dalam kalimat berikut beberapa kita akan menjelaskan hanya dua-dimensi array yang disebut matriks yang dapat dianggap sebagai tabel. Sebuah array dua dimensi dinyatakan oleh tipe data menentukan dari array, nama array dan ukuran masing-masing dimensi. Lihatlah contoh di bawah ini:

```
component_type array_name [number_of_rows] [number_of_columns];
```

number_of_rows and *number_of_columns* represent the number of rows and columns of a table, respectively.

```
int Table [3][4]; // Tabel didefinisikan memiliki 3 baris dan 4 kolom
```

Array ini dapat direpresentasikan dalam bentuk tabel.

```
table [0] [0] table [0] [1] table [0] [2] table [0] [3]
```

```
table [1] [0] table [1] [1] table [1] [2] table [1] [3]
```

```
tabel [2] [0] tabel [2] [1] tabel [2] [2] tabel [2] [3]
```

Mirip dengan vektor, elemen matriks bisa diberi nilai selama deklarasi array. Dalam contoh berikut, unsur-unsur dari Tabel array dua dimensi ditugaskan nilai. Seperti yang terlihat, array ini memiliki dua baris dan tiga kolom:

```
int Table [2] [3] = {{3,42,1}, {7,7,19}};
```

Matriks di atas juga dapat direpresentasikan dalam bentuk tabel unsur-unsur yang memiliki nilai berikut:

2.7.22. Fungsi

Setiap program yang ditulis dalam bahasa C terdiri dari sejumlah besar atau lebih kecil dari fungsi. Gagasan utama adalah untuk membagi program menjadi beberapa bagian menggunakan fungsi-fungsi dalam rangka memecahkan masalah yang sebenarnya mudah. Selain itu, fungsi memungkinkan kita untuk menggunakan keterampilan dan pengetahuan programmer lain. Misalnya, jika perlu untuk mengirim string ke layar LCD, jauh lebih mudah untuk menggunakan sebagian sudah ditulis dari program daripada memulai dari awal.

Fungsi terdiri dari perintah menentukan apa yang harus dilakukan pada variabel. Mereka dapat dibandingkan dengan subrutin. Sebagai aturan, itu jauh lebih baik untuk memiliki

program yang terdiri dari sejumlah besar fungsi sederhana dari sebuah fungsi besar beberapa. Sebuah fungsi tubuh biasanya terdiri dari beberapa perintah yang dieksekusi oleh urutan mereka ditentukan.

Setiap fungsi harus benar dinyatakan secara sehingga dapat ditafsirkan dengan benar selama proses kompilasi. Deklarasi berisi elemen-elemen berikut:

1. Nama Fungsi
2. Fungsi tubuh
3. Daftar parameter
4. Deklarasi parameter
5. Tipe hasil fungsi

Fungsi ini dapat terlihat seperti:

```
type_of_result function_name (tipe argumen1, tipe argumen2, ...)  
{  
    Command;  
    Command;  
    ...  
}
```

Contoh:

```
/* Fungsi menghitung hasil pembagian jumlah pembilang dengan penyebut  
DENOM. Fungsi mengembalikan struktur jenis div_t. */
```

```
div_t div (nomor int, int DENOM);
```

Perhatikan bahwa fungsi tidak perlu memiliki parameter, tetapi harus memiliki tanda kurung yang akan digunakan untuk memasukkan mereka. Jika tidak, compiler akan salah menafsirkan fungsi.

Jika fungsi tersebut, setelah dieksekusi, tidak memberikan hasil ke program utama atau fungsi yang disebut oleh, hasil Program dengan eksekusi setelah menghadapi penutupan braket keriting. Fungsi tersebut digunakan jika diperlukan untuk mengubah keadaan output pin mikrokontroler, selama transfer data melalui komunikasi serial, saat menulis data pada layar LCD dll compiler mengakui fungsi-fungsi oleh jenis hasil mereka ditentukan untuk menjadi **void**.

```

void function_name (type argument1, type argument2,...)
{
Commands;
}

```

Contoh:

```

void interrupt () {
    cnt ++; // Interrupt menyebabkan cnt akan bertambah 1
    PIR1.TMR1IF = 0; // Reset bit TMR1IF
}

```

Fungsi dapat diberi nama sewenang-wenang. Satu-satunya pengecualian adalah nama utama yang memiliki tujuan khusus. Yakni, program selalu dimulai eksekusi dengan fungsi ini. Ini berarti bahwa setiap program yang ditulis dalam bahasa C harus berisi satu fungsi bernama 'utama' yang tidak harus ditempatkan di awal program.

Jika perlu bahwa yang disebut fungsi mengembalikan hasil setelah dieksekusi, perintah kembali, yang dapat diikuti oleh ekspresi apapun, digunakan:

```

type_of_result function_name (type argument1, type argument2,...)
{
    Commands;
    ...
    return expression;
}

```

Jika fungsi berisi perintah return tanpa diikuti oleh ekspresi, fungsi berhenti eksekusi ketika bertemu perintah ini dan hasil Program dengan eksekusi dari perintah pertama setelah penutupan kurung kerawal.

1. Deklarasi Fungsi Baru

Terlepas dari fungsi bahwa bahasa C 'otomatis' mengakui, ada juga sepenuhnya fungsi baru yang sering digunakan dalam program. Setiap 'non-standar' fungsi harus

dideklarasikan di awal program. Fungsi deklarasi disebut prototipe dan terlihat sebagai berikut:

```
type_of_result function_name (formal parameters)
{
    description of formal parameters
    definition and declaration
    operators
    ...
}
```

Jenis fungsi yang tidak mengembalikan nilai batal. Jika jenis hasil tidak secara khusus dinyatakan dalam program ini, maka dianggap bertipe int (integer ditandatangani). Parameter ditulis dalam prototipe fungsi menentukan apa yang harus dilakukan dengan parameter nyata. Parameter fungsi prototipe disebut PARAMETER FORMAL. Contoh berikut menyatakan fungsi yang menghitung volume silinder.

Contoh:

```
const double PI = 3.14159; // Declare constant PI
```

```
float volume (float r, float h) // Declare type float for
{
    // formal parameters r and h
    float v; // Declare type of result v
    v = PI*r*r*h; // Declare function volume
    return v;
}
```

Jika perhitungan tersebut perlu dilakukan kemudian dalam program (dapat volume tangki dalam praktek), itu sudah cukup untuk menentukan PARAMETER REAL dan memanggil fungsi. Selama proses kompilasi, compiler adalah untuk menggantikan parameter formal dengan nyata seperti yang ditunjukkan di bawah:

```
float radius=5, height=10, tank; // declare type float for
... // real parameters radius,
... // height and tank
tank = volume (radius,height); // calculate the volume of tank
```

... *// by calling the volume function*

2. Perpustakaan Fungsi

Nama semua fungsi yang digunakan dalam bahasa C disimpan dalam file bernama sundulan. Fungsi-fungsi ini, tergantung pada tujuan mereka, diurutkan dalam file kecil yang disebut perpustakaan. Sebelum menggunakan salah satu dari mereka dalam program, maka perlu untuk menentukan file header yang sesuai dengan menggunakan # include perintah di awal program. Jika compiler menemukan sebuah fungsi yang tidak diketahui selama pelaksanaan program, pertama kali akan mencari deklarasi di perpustakaan tertentu.

3. Standar Ansi Perpustakaan C

Fungsi bahasa C yang tidak standar pada awal dan produsen perangkat lunak dimodifikasi mereka sesuai dengan kebutuhan mereka. Tetapi bahasa C menjadi sangat populer segera dan itu sulit untuk menjaga segala sesuatu di bawah kendali. Hal itu perlu untuk memperkenalkan semacam standar untuk meletakkan segala sesuatu dalam rangka. Standar yang ditetapkan disebut ANSI C dan berisi 24 perpustakaan dengan fungsi. Perpustakaan ini biasanya disediakan dengan setiap compiler C sebagai operasi yang paling sering dilakukan menggunakan fungsi tersebut. Adapun daftar pustaka Ansi C tersebut adalah :

<assert.h> <complex.h> <ctype.h>
<errno.h> <fcntl.h> <float.h>
<inttypes.h> <iso646.h> <limits.h>
<locale.h> <math.h> <setjmp.h>
<signal.h> <stdarg.h> <stdbool.h>
<stdint.h> <stddef.h> <stdio.h>
<stdlib.h> <string.h> <tgmath.h>
<time.h> <wchar.h> <wctype.h>

BAB III

METODOLOGI PENELITIAN

2.5 Jenis Penelitian

Dalam penelitian mengenai perancangan sistem peringatan dini bencana banjir ini penulis menggunakan jenis penelitian kuantitatif. **Penelitian kuantitatif** adalah penelitian ilmiah yang sistematis terhadap bagian-bagian dan fenomena serta hubungan-hubungannya. Tujuan penelitian kuantitatif adalah mengembangkan dan menggunakan model-model matematis, teori-teori dan/atau hipotesis yang berkaitan dengan fenomena alam. Proses pengukuran adalah bagian yang sentral dalam penelitian kuantitatif karena hal ini memberikan hubungan yang fundamental antara pengamatan empiris dan ekspresi matematis dari hubungan-hubungan kuantitatif.

Penelitian kuantitatif banyak dipergunakan baik dalam ilmu-ilmu alam maupun ilmu-ilmu sosial, dari fisika dan biologi hingga sosiologi dan jurnalisme. Pendekatan ini juga digunakan sebagai cara untuk meneliti berbagai aspek dari pendidikan. Istilah penelitian kuantitatif sering dipergunakan dalam ilmu-ilmu sosial untuk membedakannya dengan penelitian kualitatif.

Penelitian kuantitatif adalah definisi, pengukuran data kuantitatif dan statistik objektif melalui perhitungan ilmiah berasal dari sampel orang-orang atau penduduk yang diminta menjawab atas sejumlah pertanyaan tentang survei untuk menentukan frekuensi dan persentase tanggapan mereka. Sebagai contoh: 240 orang, 79% dari populasi sampel, mengatakan bahwa mereka lebih percaya pada diri mereka pribadi masa depan mereka dari setahun yang lalu hingga hari ini. Menurut ketentuan ukuran sampel statistik yang berlaku, maka 79% dari penemuan dapat diproyeksikan ke seluruh populasi dari sampel yang telah dipilih. pengambilan data ini adalah disebut sebagai survei kuantitatif atau penelitian kuantitatif.

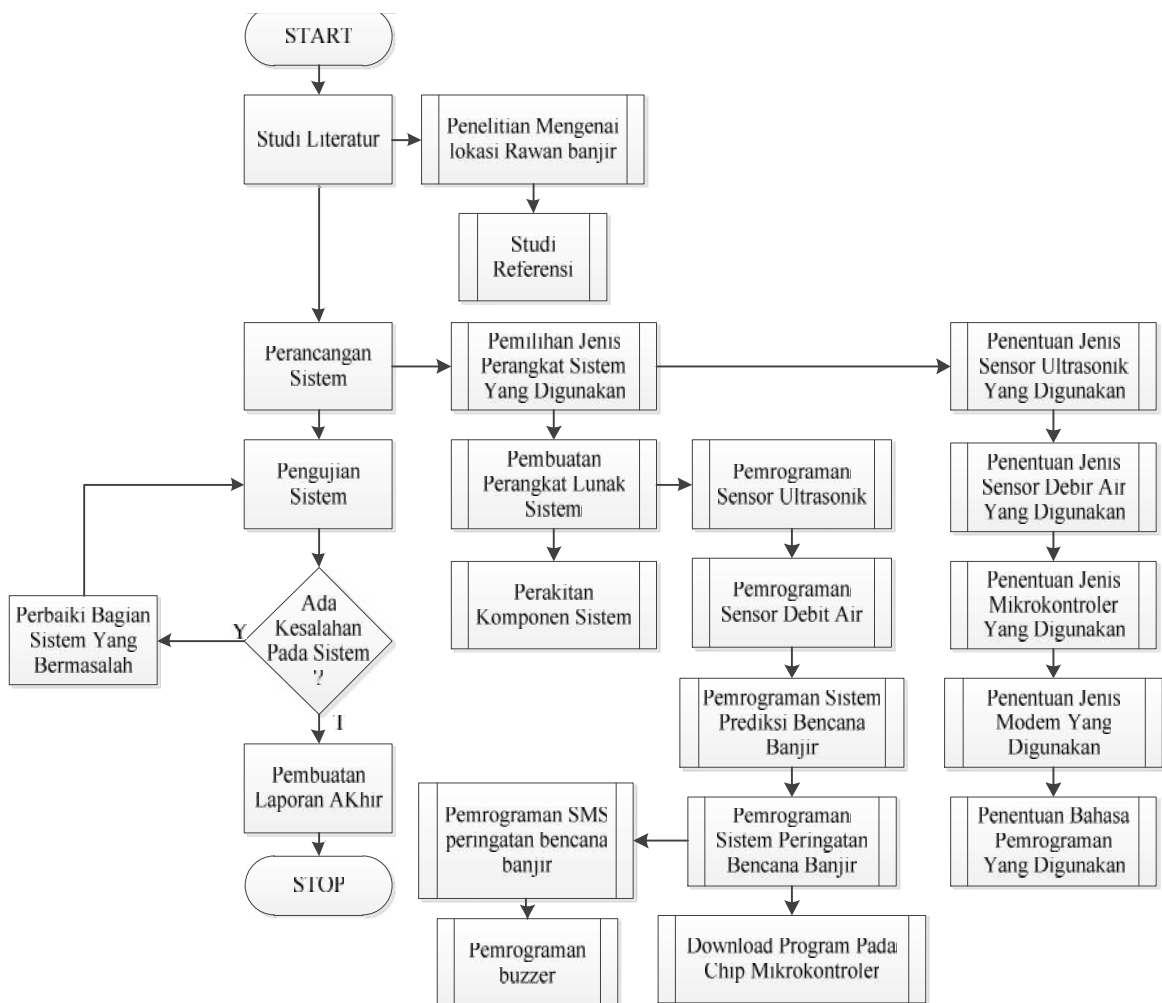
Ukuran sampel untuk survei oleh statistik dihitung dengan menggunakan rumusan untuk menentukan seberapa besar ukuran sampel yang diperlukan dari suatu populasi untuk mencapai hasil dengan tingkat akurasi yang dapat diterima. pada umumnya, para peneliti mencari ukuran sampel yang akan menghasilkan temuan dengan minimal 95% tingkat keyakinan (yang berarti bahwa jika Anda survei diulang 100 kali, 95 kali dari seratus, Anda akan mendapatkan respon yang sama) dan plus / minus 5 persentase poin

margin dari kesalahan. Banyak survei sampel dirancang untuk menghasilkan margin yang lebih kecil dari kesalahan.

Beberapa survei dengan melalui pertanyaan tertulis dan tes, kriteria yang sesuai untuk memilih metode dan teknologi untuk mengumpulkan informasi dari berbagai macam responden survei, survei dan administrasi statistik analisis dan pelaporan semua layanan yang diberikan oleh pengantar komunikasi. Namun, oleh karena sifat teknisnya metode pilihan pada survei atau penelitian oleh karena sifat teknis, maka topik yang lain tidak tercakup dalam cakupan ini.

2.6 Tahapan Penelitian

Dalam penelitian ini penulis melalui beberapa tahap dalam menyelesaikan penelitian. Adapun tahapan penelitian tersebut dapat dilihat pada *flowchart* sebagaimana terlihat pada gambar 3.1.



Gambar 3.1. *Flowchart* tahapan penelitian

2.6.1 Studi Literatur

Dalam studi literatur ada dua tahap yang akan dilakukan untuk menyelesaikan tugas akhir ini. Tahapan tersebut adalah :

1. Penelitian mengenai lokasi rawan banjir

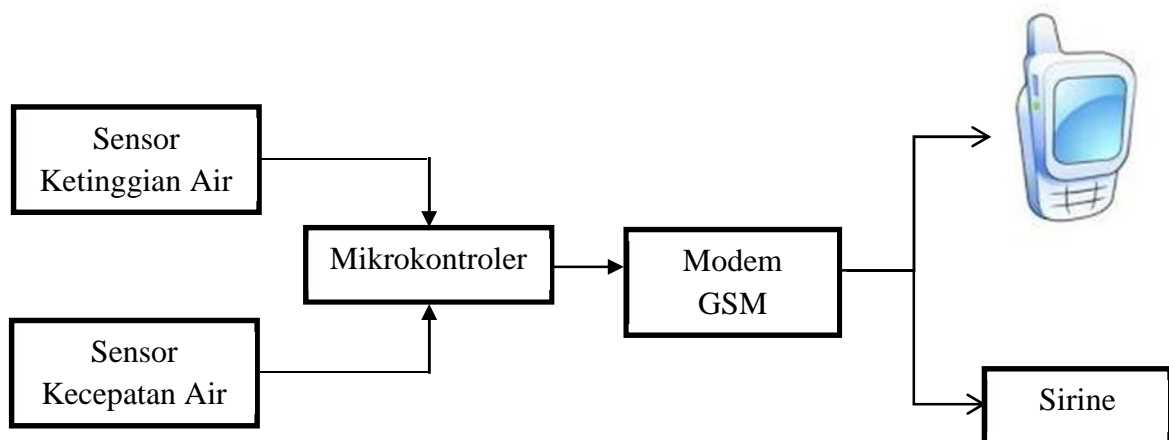
Studi ini dilakukan untuk mengetahui daerah mana saja di provinsi riau yang menjadi daerah rawan banjir. Penelitian ini juga meliputi pengumpulan data kejadian terdahulu sebagai data pendukung dalam perancangan sistem.

2. Studi referensi

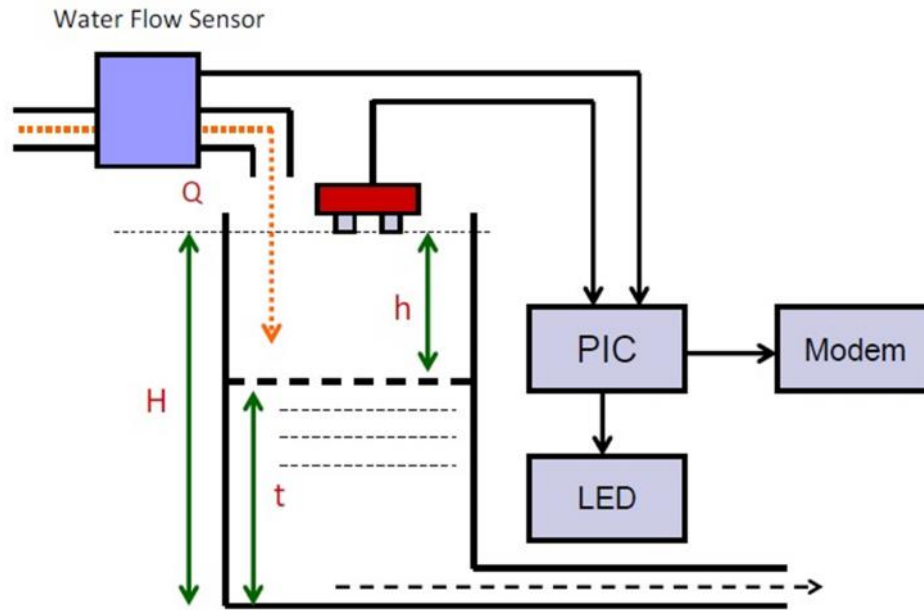
Metode ini digunakan untuk mengumpulkan dan menganalisa informasi dan konsep dasar mikrokontroler PIC16F877A, modem GSM, pengetahuan mengenai sistem output yang digunakan dalam penelitian ini.

2.6.2 Perancangan Sistem

Secara garis besar rancangan sistem dapat dilihat pada gambar 3.2 dan 3.3 dibawah ini



Gambar 3.2. Blok diagram rancangan sistem



Gambar 3.3. Rancangan *prototype* sistem

Berdasarkan gambar 3.3 dapat dilihat sistem peringatan dini musibah banjir ini terdiri atas perangkat keras dan perangkat lunak. Perangkat keras sistem ini terdiri dari :

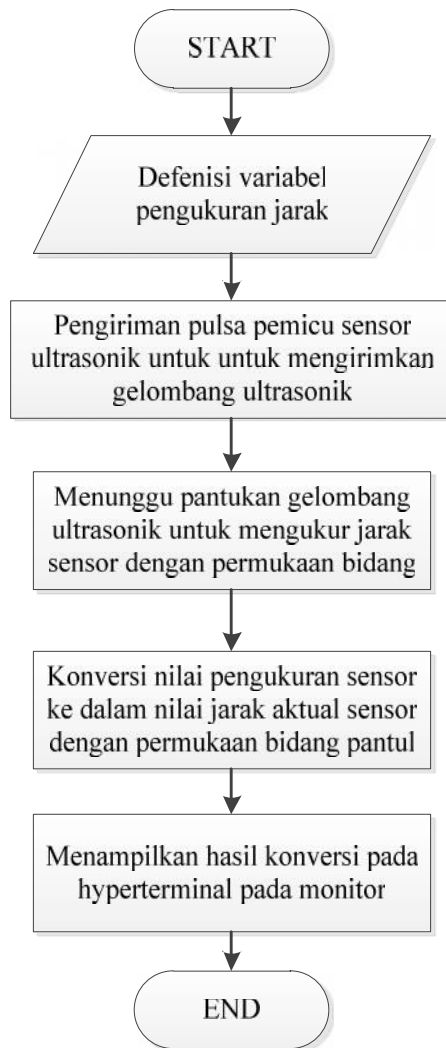
1. Sensor Ketinggian
2. Sensor kecepatan air.
3. Mikrokontroler.
4. Modem GSM
5. Sirine

Secara umum *prototype* sistem dapat digambarkan seperti gambar 3.2 dan 3.3. Sebagaimana terlihat pada gambar 3.2 dan 3.3 terdapat dua jenis data yang menjadi input mikrokontroler yaitu data pengukuran ketinggian dan data pengukuran debit air. Data pengukuran ketinggian air digunakan sebagai parameter apakah kondisi bendungan / sungai berada dalam kondisi membahayakan atau tidak. Setelah data ketinggian air dan data debit air diperoleh kemudian data tersebut dibaca oleh mikrokontroler melalui port I/O. Data hasil pembacaan port I/O untuk ketinggian air kemudian diklasifikasikan berdasarkan kondisi yang telah ditetapkan. Sedangkan data debit air diolah dalam kalkulasi matematis agar dapat menghasilkan waktu prediksi kapan banjir akan terjadi.

Setelah melalui proses klasifikasi tingkat ketinggian air dan kalkulasi waktu perkiraan terjadinya banjir kemudian mikrokontroler mengirimkan peringatan dini bencana kepada masyarakat yang berada di daerah yang terkena banjir melalui layanan pesan singkat dan bunyi sirine.

Dalam perancangan sistem ini ada beberapa tahapan yang dilakukan yaitu :

1. Menentukan jenis dan spesifikasi perangkat yang digunakan untuk membangun sistem. Proses ini meliputi :
 - a. Menentukan jenis dan spesifikasi sensor ultrasonik yang akan digunakan.
Pemilihan jenis dan spesifikasi sensor ultrasonik berdasarkan kemampuan pengukuran sensor terhadap bidang pantulan dengan tidak meninggalkan nilai ekonomis sensor.
 - b. Menentukan jenis dan spesifikasi sensor debit air yang akan digunakan.
Pemilihan dilakukan dengan menyesuaikan kemampuan sensor dalam mengukur debit air yang masuk saat pengujian sistem dengan kebutuhan sistem tersebut.
 - c. Menentukan jenis dan spesifikasi mikrokontroler yang digunakan.
Jenis dan spesifikasi mikrokontroler ditentukan berdasarkan kebutuhan sistem. Dalam perancangan sistem ini disamping mikrokontroler yang mampu mendukung pemrograman sensor dibutuhkan mikrokontroler dengan fasilitas komunikasi serial dan juga frekuensi yang besar untuk mendapatkan kecepatan eksekusi program dan akurasi sistem yang baik.
 - d. Menentukan jenis dan spesifikasi modem yang digunakan.
Pemilihan jenis dan spesifikasi modem yang digunakan dilakukandengan mempertibangkan kemampuan modem untuk pengiriman pesan singkat dengan sistem *SMS Broadcast* yang mampu mengirimkan pesan singkat ke banyak nomor tujuan dalam sekali pengiriman.
 - e. Menentukan bahasa pemrograman yang digunakan.
Bahasa program yang digunakan dalam pemrograman sistem adalah bahasa C. Bahasa C dipilih karena mudah dalam pemrograman, eksekusi dan lebih dekat kepada bahasa *assembly* sehingga mudah dalam *download*-kannya kedalam mikrokontroler.
2. Perancangan perangkat lunak sistem. Proses ini meliputi :
 - a. Pemrograman sensor ultrasonik.
Pembuatan program sensor ultrasonik ini didasarkan kepada cara kerja sensor ultrasonik. Adapun program sensor ultrasonik ini dibuat berdasarkan alur sebagaimana terlihat pada gambar 3.4.

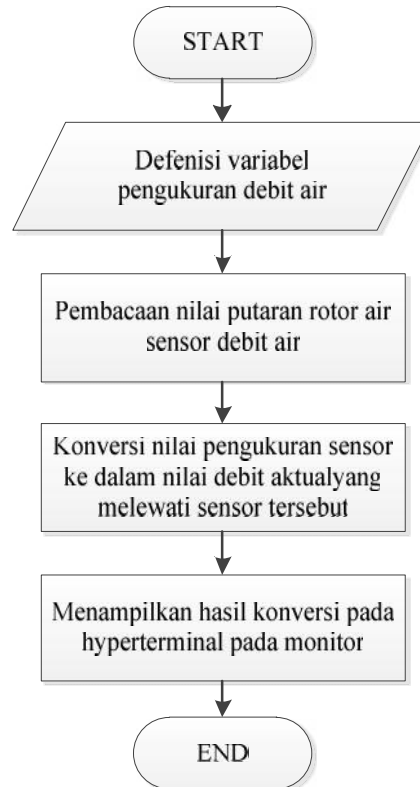


Gambar 3.4. *Flowchart* pemrograman sensor ultrasonik

Sebagaimana digambarkan dalam *flowchart* pada gambar 3.4 pemrograman sensor ultrasonik terdiri dari beberapa tahap. Pemrograman sensor diawali dengan deklarasi variabel pengukuran jarak. Setelah deklarasi variabel jarak kemudian aktifkan sinyal gelombang ultrasonik untuk memulai pengukuran dengan mengirimkan pulsa logika tinggi pada pin *signal* sensor selama 10 us. Setelah gelombang ultrasonik dikirim sensor akan menunggu hingga bagian penerima menerima pantulan sinyal. Waktu pemantulan sinyal akan disimpan sebagai nilai jarak sensor ultrasonik dalam nilai variabel yang telah dideklarasikan di awal program. Untuk mendapatkan nilai jarak yang sebenarnya nilai pengukuran kemudian dikonversi dengan nilai yang telah ditetapkan untuk mendapatkan nilai jarak yang sebenarnya. Hasil konversi inilah yang kemudian ditampilkan ke layar monitor melalui jendela *hyperterminal*.

b. Pemrograman sensor debit air.

Pemrograman sensor debit air yang digunakan berdasarkan spesifikasi dan faktor konversi nilai hasil pengukuran kepada nilai aktual debit yang masuk. Pemrograman sensor debit air dilakukan sesuai alur yang terlihat pada gambar 3.5.



Gambar 3.5. *Flowchart* pemrograman sensor debit air

Cara pemrograman sensor debit air relatif sama dengan cara pemrograman sensor ultrasonik. Deklarasi variabel pengukuran debit adalah langkah awal dalam pemrograman. Bedanya pada pemrograman sensor debit air program tidak perlu mengirimkan sinyal pemicu sensor melainkan langsung membaca nilai putaran rotor yang terdapat di dalam sensor. Nilai putaran rotor kemudian dikonversikan kepada nilai debit yang sebenarnya. Nilai debit inilah yang kemudian ditampilkan ke layar monitor melalui fasilitas *hyperterminal*.

c. Pemrograman sistem prediksi bencana banjir.

Pemrograman sistem prediksi bencana banjir didasarkan pada nilai waktu menuju banjir berdasarkan hasil perbandingan nilai volume bendungan dengan nilai debit air yang mengalir ke bendungan tersebut. Nilai waktu

hasil perbandingan dua variabel diatas adalah prediksi waktu menuju banjir.

d. Pemrograman sistem peringatan bencana banjir.

Dalam perancangan sistem peringatan dini bencana banjir ini penulis menggunakan dua media peringatan bencana banjir yaitu melalui pesan singkat (SMS) dan bunyi *buzzer*.

Sistem peringatan banjir berjalan berdasarkan nilai ketinggian air di bendungan. Maka langkah awal dalam sistem peringatan adalah pembacaan nilai ketinggian air hasil pengukuran sensor ultrasonik. Nilai ketinggian air hasil pengukuran sensor ultrasonik kemudian diklasifikasikan pada lima kondisi yaitu :

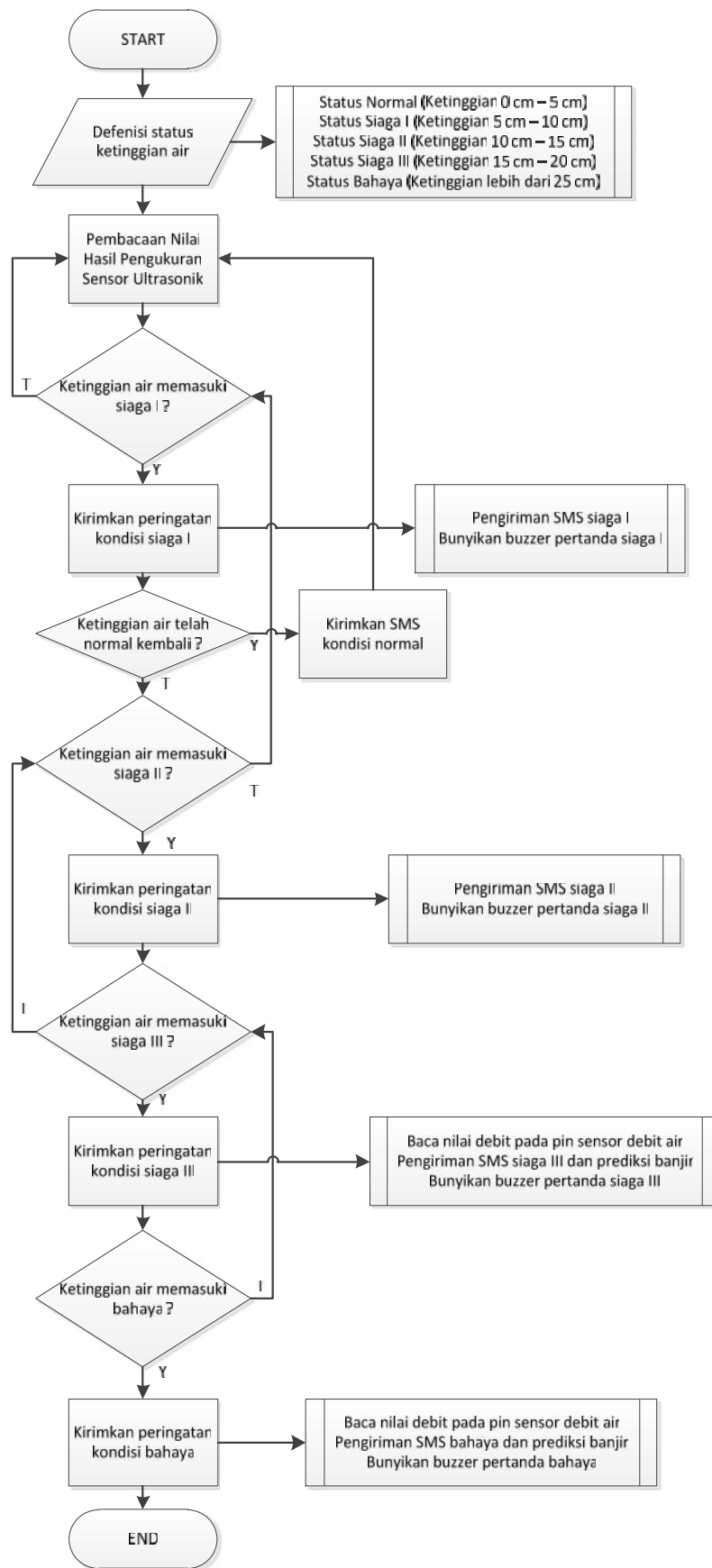
- 1) Kondisi normal
- 2) Kondisi siaga I
- 3) Kondisi siaga II
- 4) Kondisi siaga III
- 5) Kondisi bahaya

Pembacaan ketinggian air berlangsung secara kontinu selama sistem berjalan. Sistem peringatan berjalan berdasarkan beberapa parameter. Pada kondisi normal tingkat ketinggian air akan mempunyai dua kemungkinan, tetap pada kondisi normal atau memasuki kondisi siaga I. Jika ketinggian air memasuki kondisi siaga I maka sistem akan mengirimkan peringatan kondisi siaga I. Dalam kondisi siaga I ini akan ada tiga kemungkinan, tetap berada kondisi siaga I, masuk kondisi siaga II atau kembali ke kondisi normal. Jika ketinggian air kembali ke kondisi normal maka proses berjalan sebagaimana biasa. Namun jika ketinggian bergerak ke kondisi siaga II sistem akan masuk kepada opsi pada kondisi siaga II. Pada kondisi siaga I akan ada peringatan melalui media pesan singkat. Jika pesan singkat tidak tersampaikan *buzzer* akan menjadi media peringatan cadangan.

Pilihan pada kondisi siaga II adalah tetap pada kondisi siaga II, kembali ke kondisi siaga I atau malah bergerak pada kondisi siaga III. Kalaupun pada kondisi siaga II ketinggian air akan kembali ke kondisi normal tetap akan melewati dan berada pada kondisi siaga I untuk beberapa waktu sebelum

akhirnya kembali ke kondisi normal. Pada semua pilihan kondisi pengukuran ketinggian air akan tetap berlangsung sesuai proses yang berjalan. Pada kondisi siaga II sistem akan memberikan sinyal peringatan melalui pesan singkat sebagai media peringatan utama dan *buzzer* sebagai media peringatan cadangan. Intensitas bunyi *buzzer* akan berbeda pada setiap kondisi siaga untuk memudahkan pendengar mengidentifikasi kondisi yang terjadi pada saat kondisi siaga terjadi.

Jika kemudian ketinggian air bergerak ke kondisi siaga III opsi yang tersedia menjadi berbeda. Pilihan kondisi yang ada adalah tetap pada kondisi siaga III, turun ke kondisi siaga II atau bergerak ke kondisi tertinggi yaitu kondisi berbahaya. Penurunan ketinggian air akan tetap melalui tahapan kondisi siaga II dan siaga I sebelum menurun memasuki kondisi normal. Pada kondisi siaga III disamping adanya pemberitahuan mengenai status kondisi siaga III akan ada informasi perkiraan kapan kondisi banjir akan terjadi. Media peringatan banjir pada kondisi ini akan sama dengan kondisi siaga I dan siaga II yaitu melalui media pesan singkat dan *buzzer*. Jika terjadi penurunan dan penambahan ketinggian air pengukuran akan tetap berlangsung untuk mengetahui apakah ketinggian air berubah atau tidak. Jika ketinggian air masih berada pada salah satu kondisi siaga maka peringatan banjir akan muncul sesuai kondisi siaga yang terjadi pada saat itu. Sistem peringatan melalui pesan singkat dan *buzzer* ini juga akan berlaku pada kondisi berbahaya dengan pilihan kondisi turun ke kondisi siaga III atau tetap pada kondisi berbahaya sebagaimana tergambar pada gambar 3.6. dibawah ini.



Gambar 3.6. Flowchart pemrograman peringatan bencana banjir

e. *Download* program pada *chip* mikrokontroler.

Setelah perancangan program selesai secara keseluruhan kemudian program yang sudah jadi di-*download*-kan kedalam chip mikrokontroler. Setelah proses *download* program selesai selanjutnya sistem dirakit sesuai blok diagram yang telah dibuat.

3. Perakitan komponen sistem.

Seluruh komponen perangkat keras penyusun sistem dihubungkan sesuai perencanaan blok diagram yang telah dibuat. Setelah perakitan selesai sistem akan masuk pada tahap pengujian untuk mengetahui apakah sistem sudah berjalan sesuai rancangan atau tidak. Perbaikan akan dilakukan pada bagian sistem yang belum sesuai blok diagram.

2.6.3 Pengujian Sistem

Dalam tahapan ini Penulis melakukan simulasi terhadap sistem yang telah dibuat untuk mengetahui sejauh mana keberhasilan sistem deteksi dini musibah banjir berbasis mikrokontroler PIC16F877A dalam mendeteksi dan menginformasikan status ketinggian permukaan air pada sungai/bendungan. Adapun tahapan dalam pengujian sistem ini adalah :

1. Pengujian koneksi mikrokontroler dengan komputer

Pada tahapan ini penulis melakukan pengujian untuk mengetahui apakah koneksi antara mikrokontroler dan komputer berjalan dengan baik. Pengujian dilakukan dengan percobaan menampilkan hasil ketikan *keyboard* pada layar komputer melalui mikrokontroler.

2. Pengujian sensor ultrasonik dalam membaca ketinggian air.

Pengujian pengukuran sensor ultrasonik dilakukan dengan menghadapkan sensor ultrasonik dengan bidang pantul. Kemudian jarak diukur berdasarkan program yang telah dibuat dan hasilnya ditampilkan melalui *hyperterminal*. Hasil pengukuran dibandingkan dengan nilai sebenarnya untuk mengetahui tingkat akurasi sensor ultrasonik.

3. Pengujian sensor debit air dalam mengukur debit air.

Pengujian sensor debit air dilakukan sebagaimana tujuan dalam menguji sensor ultrasonik. Sensor debit air diberi masukan berupa aliran air yang kemudian akan diukur debitnya. Hasil pengukuran ditampilkan melalui *hyperterminal*

kemudian dibandingkan dengan volume air yang masuk untuk mengetahui tingkat akurasi sensor.

4. Pengujian sistem peringatan bencana banjir.

Pengujian peringatan bencana banjir dilakukan dengan mengukur ketinggian dan nilai debit air hingga ketinggian air mencapai kondisi siaga maupun berbahaya. Pengujian dilakukan dengan eksekusi program pada mikrokontroler kemudian menampilkan hasil peringatan banjir melalui media pesan singkat pada telepon seluler.

2.6.4 Pembuatan Laporan

Dalam tahap ini Penulis akan menyusun laporan dari kegiatan selama pengerjaan tugas akhir ini mulai dari pengumpulan referensi, perancangan sistem hingga hasil simulasi sistem deteksi dini bencana banjir berbasis mikrokontroler PIC16F877A.

2.7 Jenis dan Metode Pengumpulan Data

Dalam penelitian ini penulis mengumpulkan data hasil pengukuran sensor ultrasonik dan flow meter yang digunakan sebagai parameter peringatan dini bencana banjir. Adapun rincian data tersebut adalah :

1. Data pengukuran manual jarak objek dari sensor ultrasonik. Data tersebut diperoleh dengan cara mengukur jarak antara sensor ultrasonik dengan bidang pantul menggunakan mistar / penggaris.
2. Data hasil eksperimen program pengukuran jarak ultrasonik yang diperoleh saat mengeksekusi program sensor ultrasonik dimana hasil pengukuran sensor ultrasonik tersebut akan ditampilkan di layar monitor.
3. Data pengukuran manual debit air yang didapat dengan menghitung secara manual volume air yang masuk kedalam suatu bejana dalam rentang waktu tertentu.
4. Data hasil eksperimen program pembaca debit air yang didapat dari pembacaan mikrokontroler terhadap nilai debit air per menit.

2.8 Analisa Data

Setelah didapat data hasil eksperimen pengukuran sensor dan hasil pengukuran manual selanjutnya data dianalisa melalui perbandingan menggunakan persamaan

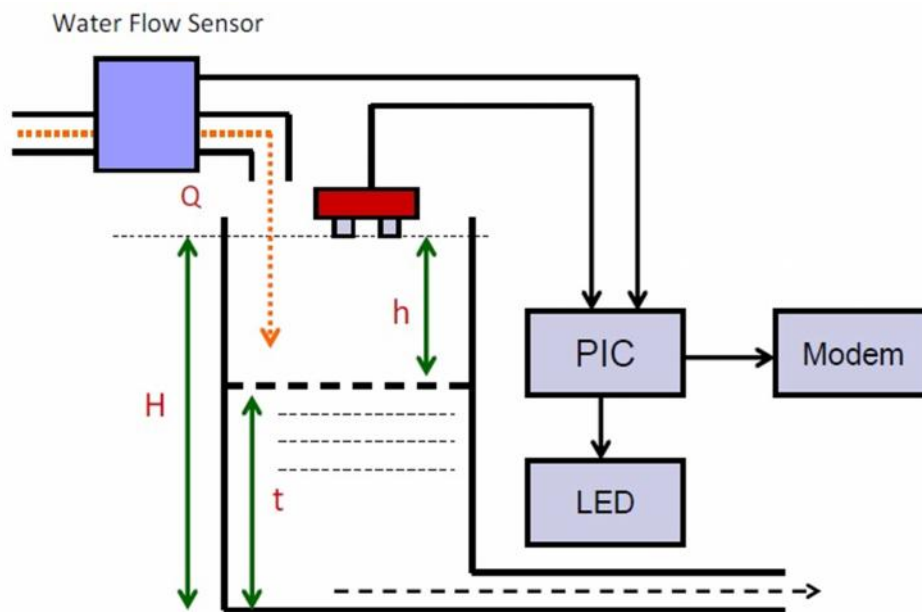
matematis untuk mendapatkan persamaan akhir. Persamaan tersebut akan disertakan di dalam program pengendali sensor agar hasil pengukuran sensor sesuai dengan hasil pengukuran manual untuk variabel ketinggian dan debit air.

BAB IV

HASIL DAN ANALISA

4.8 Hasil *Prototype* Sistem

Setelah melewati proses perancangan sistem baik dari segi perangkat keras maupun perangkat lunak sistem yang didasarkan kepada perancangan diagram blok sistem sebagaimana telah dijelaskan pada bab III yaitu bagian metodologi penelitian hasil rancangan *prototype* sistem dapat dilihat pada gambar 4.1 dan 4.2.



Gambar 4.1. Rancangan *prototype* sistem



(a)

(b)

Gambar 4.2. *Prototype* sistem (a) Tampak Depan (b) Tampak Belakang

Sebagaimana terlihat pada gambar 4.2. sistem telah dikemas sedemikian rupa sehingga siap untuk simulasi. Sistem peringatan dini bencana banjir ini terdiri dari

beberapa komponen yang berada di dalam dan luar kotak pembungkus rangkaian perangkat keras sistem. Perangkat yang berada didalam kotak tersebut adalah :

1. Rangkaian minimum sistem mikrokontroler.

Rangkaian minimum sistem adalah suatu kesatuan rangkaian elektronik yang berfungsi mengaktifkan mikrokontroler sekaligus media komunikasi antara mikrokontroler dengan perangkat eksternal. Rangkaian minimum sistem berupa papan sirkuit rangkaian mikrokontroler beserta perangkat pendukung yaitu :

- a. 1 buah chip mikrokontroler PIC16F877A lengkap dengan soket 40 pin sebagai media pemasangan chip mikrokontroler.
- b. Kristal yang berfungsi memberikan input clock sinyal bagi mikrokontroler.
- c. Port masukan tegangan 9 Volt arus searah (DC) sebagai catu daya mikrokontroler.
- d. Port tegangan keluaran 5 Volt arus searah sebagai sumber tegangan bagi perangkat eksternal yang akan dihubungkan dengan sistem / mikrokontroler.
- e. Port I/O bagi perangkat eksternal yang akan dihubungkan dengan sistem.
- f. Tombol reset minimum sistem yang difungsikan sebagai media reset mikrokontroler sekaligus mengaktifkan *bootstrap* yang berfungsi menerima dan menyimpan program yang *download*kan dari komputer / laptop ke mikrokontroler

2. Rangkaian konverter UART TTL to RS232

Minimum sistem mikrokontroler PIC16F877A memiliki *port* I/O yang berfungsi sebagai media komunikasi dengan perangkat luar. Untuk itu agar sistem dapat berkomunikasi dengan perangkat serial dibutuhkan rangkaian konverter berfungsi sebagai antarmuka minimum sistem mikrokontroler dengan perangkat serial.

Sedangkan perangkat yang berada diluar kotak berfungsi sebagai media input sistem untuk menentukan nilai dan parameter pengukuran yang dianalisa oleh sistem peringatan tersebut. Perangkat yang digunakan pada sisi luar kotak adalah :

1. 1 unit sensor ultrasonik untuk mengukur ketinggian permukaan air yang ditempelkan di bagian bawah kotak.
2. 1 unit flow meter sebagai alat pengukur debit air
3. 1 unit stop kran yang berfungsi mengendalikan air yang memasuki bendungan mini sistem.
4. Pipa yang berfungsi sebagai media input air kedalam bendungan mini.

5. 1 unit modem GSM berikut kabel penghubung antara modem dengan minimum sistem yang berfungsi sebagai media pengirim pesan singkat peringatan jika ketinggian air berada dalam status siaga atau berbahaya
6. 1 unit catu daya sistem mikrokontroler.
7. 1 unit catu daya modem GSM.

Setelah seluruh perangkat sistem dirangkai sesuai dengan rancangan *prototype* sistem diberi input air yang berasal dari kran kamar mandi kemudian dialirkan melalui selang menuju pipa air sehingga air akan memasuki bendungan mini. Debit air akan diukur melalui aliran arus air yang melewati flow meter sedangkan ketinggian permukaan air yang masuk ke bendungan mini akan diukur oleh sensor ultrasonik

4.9 Analisa Cara Kerja Mikrokontroler

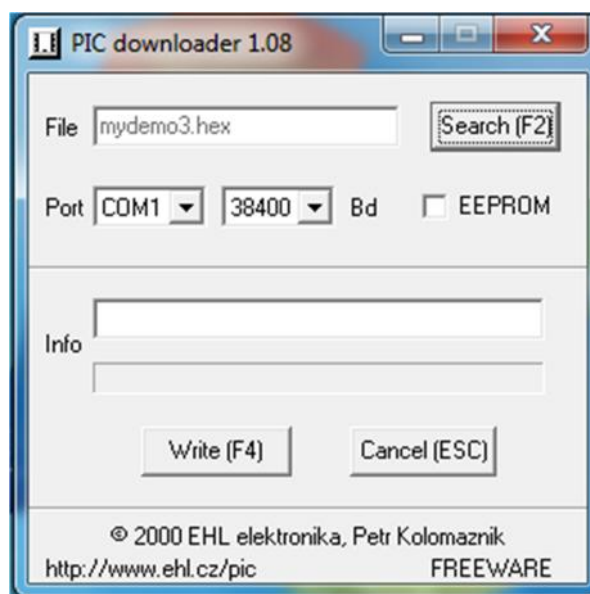
Dalam menjalankan program mikrokontroler memiliki prinsip kerja berbeda tergantung prinsip kerja program yang diunduh ke dalam *chip IC*-nya. Dengan mempelajari beberapa program mikrokontroler kita dapat menarik suatu kesimpulan mengenai prinsip kerja mikrokontroler secara umum. Berdasarkan ujicoba beberapa program yang diunduh ke dalam mikrokontroler maka didapat kesimpulan cara kerja mikrokontroler sebagai berikut :

1. Mikrokontroler membaca nilai yang berada pada register *Program Counter*, setelah membaca nilai pada *Program Counter*, mikrokontroler mengambil data pada ROM dengan *address* sebagaimana nilai yang tertera pada *Program Counter*. Selanjutnya akan terjadi peningkatan nilai pada *Program Counter (increment)* secara otomatis. Proses pembacaan *Program Counter* dan pengambilan data pada *address* dan register akan disesuaikan dengan urutan instruksi program pengendali mikrokontroler yang sebelumnya telah dibuat oleh pemakai.
2. Pembacaan data, operasi data dan eksekusi instruksi tersebut diolah dan dijalankan bergantung pada jenis instruksi: bisa membaca, mengubah nilai-nilai pada register, RAM, isi port atau melakukan pembacaan dan dilanjutkan dengan perubahan data. Proses pembacaan ROM dan pengolahan data pada RAM akan berlangsung secara berkelanjutan sesuai program yang diunduh ke dalam *chip* mikrokontroler tersebut sampai arus catu daya mikrokontroler dimatikan.

Untuk memprogram sebuah mikrokontroler kita membutuhkan sebuah rangkaian *downloader*. *Downloader* yang digunakan beragam tergantung jenis *chip* yang digunakan.

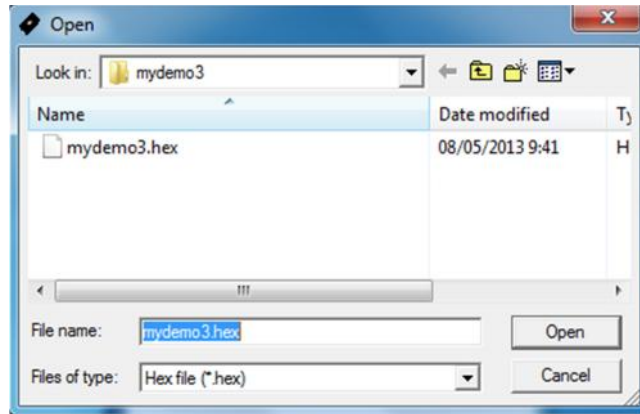
Untuk mikrokontroler *PIC* khususnya PIC16F877A proses *download* program bisa menggunakan *port Universal Serial Bus (USB)* yang tersedia pada komputer / *laptop*. salah satu cara yang paling mudah dalam memprogram mikrokontroler PIC16F877A adalah menggunakan *PIC Downloader*. *PIC Downloader* adalah program *downloader* yang bekerja dengan sebuah *firmware bootstrap* yang telah ditanamkan sebelumnya dalam *chip* mikrokontroler. Adapun cara pemrograman mikrokontroler PIC16F877A menggunakan *PIC Downloader* adalah sebagai berikut :

1. Jalankan program PIC Downloader sehingga muncul tampilan sebagaimana terlihat pada gambar 4.3.



Gambar 4.3. Tampilan program *PIC Downloader*

2. Jenis file yang akan didownloadkan oleh software ini ke dalam *chip* mikrokontroler adalah jenis *file* dengan ekstensi *hex* yang merupakan *file* pemrograman dengan bahasa tingkat rendah (*low level language*). Hal ini dikarenakan *chip* mikrokontroler tidak bisa mengenali dan mengeksekusi program bahasa C secara langsung melainkan hanya bisa mengenali dan mengeksekusi program dengan bahasa mesin sehingga harus di-*compile* untuk menghasilkan file *hex*. *File* dengan ekstensi *hex* adalah *file* hasil *compile software* pemrograman C untuk mikrokontroler. Untuk memprogram mikrokontroler PIC16F877A tersebut maka kita harus mencari dimana lokasi *file hex* yang akan di-*download*-kan ke dalam *chip* mikrokontroler dengan mengklik tombol *search* atau menekan tombol F2 pada *keyboard* sehingga akan muncul daftar direktori pada komputer sebagaimana terlihat pada gambar 4.4.



Gambar 4.4. Kotak dialog *Open file hex*

3. Setelah menentukan file yang akan di-*download*-kan ke dalam *chip* mikrokontroler kemudian klik ***Open***.
4. Setelah memilih file hex yang akan di-*download*-kan sesuaikan *port* pada *programer* dengan *port* mikrokontroler yang terhubung pada komputer. Setelah menentukan *port* yang akan digunakan tentukan *Baudrate* yang akan digunakan. Untuk mikrokontroler PIC16F877A *Baudrate* standarnya adalah 38400. *Baudrate* disesuaikan dengan jenis *chip* mikrokontroler yang digunakan.
5. Setelah memilih *file hex* yang akan di-*download*-kan, menentukan *port* dan *Baudrate* yang digunakan klik ***Write*** atau dengan menekan tombol F4 pada *keyboard* komputer. Sesaat setelah mengklik *Write* tekan tombol *reset* yang ada pada *board* sistem minimum mikrokontroler.

4.10 Analisa Pengujian Sensor Ketinggian Air

Sensor ketinggian air yang digunakan adalah sensor ultrasonik. Jenis sensor ultrasonik yang digunakan dalam perancangan sistem ini adalah Sensor Ultrasonik Seed. Sensor ultrasonik Seeed adalah modul pengukuran jarak non-kontak. Sensor ini dirancang untuk penggunaan proyek modular mudah beradaptasi dengan kinerja industri. Sensor ultrasonik seeed memiliki spesifikasi sebagai berikut :

- a. Supply tegangan 5 V
- b. Konsumsi arus keseluruhan 15 mA
- c. Frekuensi Ultrasonic 40k Hz
- d. Rentang maksimal 400 cm
- e. Rentang minimal 3 cm
- f. Resolusi 1 cm

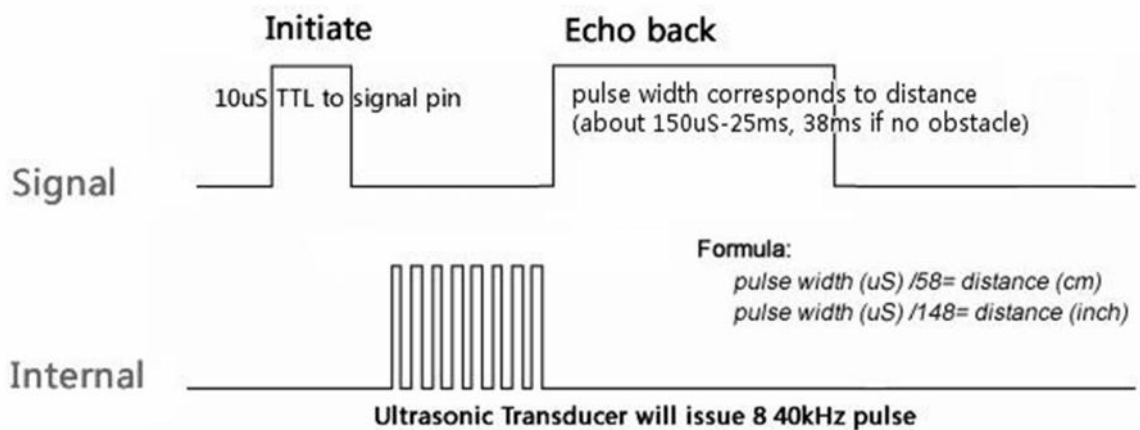
- g. Pemicu Pulse Width 10 μ s
- h. Dimensi luar 43x20x15 mm

Sebuah pulsa ultrasonik pendek ditransmisikan pada, waktu 0 tercermin oleh suatu benda. Sensor menerima sinyal ini dan mengubahnya menjadi sinyal listrik. Pulsa berikutnya dapat dipancarkan ketika gema tersebut memudar. Periode waktu ini disebut periode siklus. Periode siklus merekomendasikan seharusnya tidak kurang dari 50ms.



Gambar 4.5. Sensor ultrasonik Seed

(sumber : http://www.seeedstudio.com/wiki/Ultra_Sonic_range_measurement_module)



Gambar 4.6. Diagram sinyal sensor ultrasonik seed

(sumber : http://www.seeedstudio.com/wiki/Ultra_Sonic_range_measurement_module)

Jika pulsa pemicu selebar 10 μ s dikirim ke pin sinyal, modul ultrasonik akan menampilkan delapan sinyal ultrasonik 40kHz dan mendeteksi gema kembali. Jarak diukur sebanding dengan lebar pulsa gema dan dapat dihitung dengan rumus :

$$S = \frac{340 \cdot t}{2} \quad \dots (4.1)$$

Jika kendala tidak terdeteksi, output pin akan memberikan sinyal tingkat tinggi 38ms. Sebagaimana disebutkan diatas sensor ultrasonik digunakan untuk mengukur ketinggian air pada bendungan. Berdasarkan cara kerja sensor sebagaimana dijelaskan diatas maka kami

menyusun program untuk menjalankan fungsi sensor ultrasonik dalam bahasa C dengan rincian urutan kerja sebagai berikut :

1. Mendefinisikan variabel untuk menyimpan hasil pengukuran sensor ultrasonik. Sebagai contoh dalam program ini kami menggunakan variabel count dengan tipe data integer. Cara defenisi variabelnya adalah dengan mengetikkan :

```
unsigned int count;
```

2. Setelah mendefinisikan variabel count selanjutnya kita harus mengeset pin port I/O mana dari mikrokontroler yang akan digunakan untuk berkomunikasi dengan sensor ultrasonik. Dalam contoh ini kami menggunakan pin 1 port D mikrokontroler PIC16F877A. Defenisi pin port I/O sebagai output dilakukan melalui register TRIS. Mula-mula kita memberi logika 0 pada pin 1 port D agar pin 1 port D dapat berfungsi sebagai output. Hal ini bertujuan agar mikro dapat mengirimkan sinyal selama 10 us yang digunakan sebagai sinyal pemicu sensor untuk mulai mengukur jarak sensor dengan bidang pantul. untuk itu kami menggunakan perintah :

...

```
TRISD.F1 = 0; >> mengeset pin 1 port D sebagai output
```

```
PORTD.F1 = 0; >> memberikan nilai awal pin 1 port D dengan nilai 0
```

```
delay_us(5); >> memberikan delay selama 5 us sebelum mikro mengirimkan sinyal trigger
```

...

3. Setelah kita mengeset pin 1 port D sebagai output kemudian kita kirimkan logika 1 selama minimal 10 us sebagai sinyal pemicu sensor ultrasonik. Dalam contoh ini kami berikan sinyal pemicu selama 20 us dengan menggunakan perintah :

...

```
PORTD.F1 = 1;
```

```
delay_us(20);
```

...

4. Setelah sinyal pemicu diberikan dan sensor mulai mengukur kita memberikan logika 1 pada pin 1 port D untuk mengeset pin 1 port D sebagai input sehingga mikrokontroler dapat membaca hasil pengukuran sensor ultrasonik. Langkah ini dilakukan dengan perintah :

...

```
PORTD.F1 = 0; >> memberikan logika 0 pada pin 1 port D
```

```
TRISD.F1 = 1; >> mengeset pin 1 port D sebagai input
```

...

5. Setelah mengeset pin 1 pada port D sebagai input kita akan membaca hasil pengukuran sensor ultrasonik menggunakan perintah :

...

```
while( PORTD.F1 == 0 );  
count = 0;  
while( PORTD.F1 == 1 )  
count++;  
return count / 22;
```

...

Berdasarkan rincian urutan kerja dan perintah yang digunakan untuk proses yang dibutuhkan dalam pembacaan jarak oleh sensor ultrasonik maka didapat sebuah program lengkap yang disusun dalam sub program yang diberi nama us sebagaimana tertera pada lampiran 1.

Untuk memastikan program diatas bekerja dengan benar dalam mengukur jarak sensor dengan bidang pantul kami mencoba menampilkan hasil pengukurannya pada aplikasi hyperterminal. Untuk menampilkan hasil pengukuran pada terminal kita harus menggunakan rutin komunikasi serial (USART). Adapun tahap penampilannya adalah sebagai berikut :

1. Defenisi fungsi Usart_write_text

Defenisi fungsi Usart_write_text dilakukan agar kita dapat mengirimkan teks ke terminal melalui komunikasi serial. Defenisi fungsi dilakukan dengan perintah :

...

```
void Usart_Write_Text(char* msg) {  
    unsigned char i;  
    for ( i = 0; msg[i] != 0; i++ )  
        Usart_Write(msg[i]);  
}
```

...

2. Mendefenisikan variabel dist dan mentransfer nilai jarak dari sub program us ke variabel dist. Adapun perintah yang digunakan adalah :

...

```
unsigned int dist; >> defenisi variabel dist
```

```
dist = us(); >> memasukkan nilai hasil sub program us ke variabel dist
```

...

3. Inisialisasi komunikasi serial menggunakan perintah :

...

```
Usart_Init(9600);
```

...

4. Mentransfer nilai dist menjadi karakter agar nilai dist dapat ditampilkan pada terminal menggunakan perintah :

...

```
inttostr(dist,buff);
```

...

5. Menampilkan hasil pengukuran sensor ke terminal menggunakan perintah :

...

```
Usart_Write_Text("Jarak: ");
```

```
Usart_Write_Text(buff);
```

```
Usart_Write_Text(" cm\r\n");
```

...

Berdasarkan rincian diatas maka didapat program lengkap pengujian pengukuran jarak menggunakan sensor ultrasonik sebagaimana terdapat dalam lampiran 1. Tegangan kerja pada pin *signal* sensor ultrasonik dapat dilihat pada tabel 4.1.

Tabel 4.1. Tegangan kerja sensor ultrasonik pada saat pengukuran berlangsung.

Kondisi	Tegangan pin V _{cc}	Tegangan pin Signal
Normal	5 V	2,4 V
Siaga I	5 V	2,4 V
Siaga II	5 V	2,4 V
Siaga III	5 V	2,4 V
Bahaya	5 V	2,4 V

Adapun hasil pengujian program tersebut setelah dijalankan didapat hasil tampilan terminal seperti terlihat pada gambar 4.7.

Jarak:	194 cm
Jarak:	194 cm
Jarak:	193 cm
Jarak:	223 cm
Jarak:	13 cm
Jarak:	12 cm
Jarak:	11 cm
Jarak:	11 cm
Jarak:	22 cm
Jarak:	23 cm
Jarak:	23 cm
Jarak:	23 cm
Jarak:	192 cm
Jarak:	192 cm
Jarak:	193 cm
Jarak:	193 cm
Jarak:	193 cm
Jarak:	193 cm
Jarak:	193 cm
Jarak:	192 cm
Jarak:	193 cm
Jarak:	193 cm
Jarak:	192 cm

Gambar 4.7. Tampilan terminal hasil pengujian sensor ultrasonik

4.11 Analisa Pengujian Sensor Kecepatan Air



Gambar 4.8. Sensor aliran air

(sumber : <http://www.mikron123.com/index.php/Air/Sensor-Flow-Aliran-Air/Detailed-product-flyer.html>)

Sensor kecepatan air yang digunakan dalam perancangan sistem ini adalah flowmeter. Flowmeter terdiri dari tubuh katup plastik, rotor air, dan sensor hall-effect. Ketika air mengalir melalui, gulungan rotor rotor. Kecepatannya berubah dengan tingkat aliran yang berbeda. Output sinyal sensor hall-efek akan menghasilkan pulsa yang sesuai.

Adapun spesifikasi peralatan yang digunakan adalah sebagai berikut :

- a) Tegangan kerja 5V-24V
- b) Arus maksimum 15 mA (DC 5V)
- c) Berat 43 g
- d) Diameter eksternal 20mm
- e) Kisaran laju alir 1 ~ 30 L / menit
- f) Suhu pengoperasian 0 ~ 80
- g) Suhu cairan <120

- h) Operasi kelembaban 35% ~ 90% RH
- i) Tekanan operasi di bawah 1.2Mpa
- j) Suhu penyimpanan -25 ~ +80
- k) Kelembaban penyimpanan 25% ~ 90% RH

Dalam memprogram pembacaan input flowmeter ada beberapa tahap yang melibatkan interupsi. Hal ini disebabkan pembacaan flowmeter hanya berlaku pada kondisi tertentu dimana dalam perancangan ini kami hanya membaca hasil pengukuran flowmeter pada kondisi siaga 3 dan kondisi berbahaya. Dalam program pembacaan flowmeter ini kami menggunakan register INTCON yang berfungsi sebagai register yang mengatur terjadinya interupsi pada program. Untuk pembacaan pulsa flowmeter kita menggunakan perintah :

```
...
void interrupt() {
    if ( INTCON.INTF ) {
        flowmeter++;
        INTCON.INTF = 0;
    }
}
...
```

Selanjutnya kita akan mengeset bit GIE (Global Interrupt Enable bit) pada register INTCON agar mikrokontroler dapat mengizinkan interupsi dalam program. Pengesetan untuk mengaktifkan seluruh interupsi dilakukan dengan perintah :

```
...
INTCON.GIE = 1;
...
```

Disamping mengeset bit GIE pada INTCON kita juga harus mengeset isi register OPTION agar mikrokontroler dapat bekerja sesuai kebutuhan sistem. Dalam perancangan kali ini kami mengaktifkan bit pull-up dan pulsa timer pada sisi naik saat melakukan increment. Hal tersebut dilakukan dengan perintah :

```
...
OPTION_REG = 0x40;
...
```

Setelah mengeset register interupsi dan register option kita akan membaca hasil pengukuran flowmeter. Mula-mula kita berikan nilai 0 sebagai angka awal pulsa flowmeter dengan perintah :

...

```
flowmeter = 0;
```

...

Selanjutnya set bit INTE (INT External Interupt Enable bit) pada logika 1 agar interupsi eksternal pada pin INT (pin 33 pada mikrokontroler) dapat diizinkan. Interupsi eksternal akan digunakan untuk menunggu proses pembacaan pulsa flowmeter selama 1 detik. Setelah didapat jumlah pulsa flowmeter kemudian nilai pulsa dibagi konstanta yang ditetapkan masing-masing produsen flowmeter. Untuk flowmeter yang digunakan dalam rancangan ini konstanta pembagi ditetapkan senilai 7,5. Dengan demikian rumus perhitungan debit dalam program menjadi :

$$\text{debit} = \text{flowmeter} / 7.5 \quad \dots (4.2)$$

Nilai debit inilah yang nantinya diolah melalui perhitungan dalam sub program prediksi sehingga kita dapat menghitung waktu perkiraan kapan banjir akan terjadi. Untuk memastikan program flowmeter berjalan dengan baik kami juga melakukan ujicoba untuk menampilkan hasil pengukuran flowmeter ke terminal menggunakan rutin komunikasi serial yang sama seperti halnya rutin komunikasi serial yang digunakan sensor ultrasonik. Tegangan kerja pada pin *flowmeter* dapat dilihat pada tabel 4.2 dibawah ini.

Tabel 4.2. Tegangan kerja *flowmeter* pada saat pengukuran berlangsung.

Kondisi	Tegangan pin V _{cc}	Tegangan pin Signal
Normal	5 V	5 V
Siaga I	5 V	5 V
Siaga II	5 V	5 V
Siaga III	5 V	5 V
Bahaya	5 V	5 V

Saat melakukan pengujian hasil pengukuran flowmeter tampilan hasil pengukurannya dapat dilihat pada gambar 4.9.

```

Debit: 0 liter/menit
Debit: 0 liter/menit
Debit: 0 liter/menit
Debit: 0 liter/menit
Debit: 0 liter/menit
Debit: 0 liter/menit
Debit: 0 liter/menit
Debit: 0 liter/menit
Debit: 0 liter/menit
Debit: 0 liter/menit
Debit: 0 liter/menit
Debit: 0 liter/menit
Debit: 0 liter/menit
Debit: 0 liter/menit
Debit: 0 liter/menit
Debit: 4 liter/menit
Debit: 7 liter/menit
Debit: 8 liter/menit
Debit: 15 liter/menit
Debit: 6 liter/menit
Debit: 2 liter/menit
Debit: 19 liter/menit
Debit: 0 liter/menit
Debit: 0 liter/menit
Debit: 0 liter/menit
-

```

Gambar 4.9. Tampilan terminal hasil pengukuran flowmeter

4.12 Pengujian Program Peringatan Bencana Banjir

Dalam tahapan pengujian sistem kami menggunakan variabel volume air bendungan sebagai referensi untuk mempermudah prediksi banjir menggunakan debit air. Hal ini dikarenakan sistem prediksi melalui bendungan dinilai lebih mudah untuk dihitung prediksi volume dan debitnya dan bisa berlaku untuk beberapa daerah rawan banjir sebagaimana dijelaskan pada tabel 4.3.

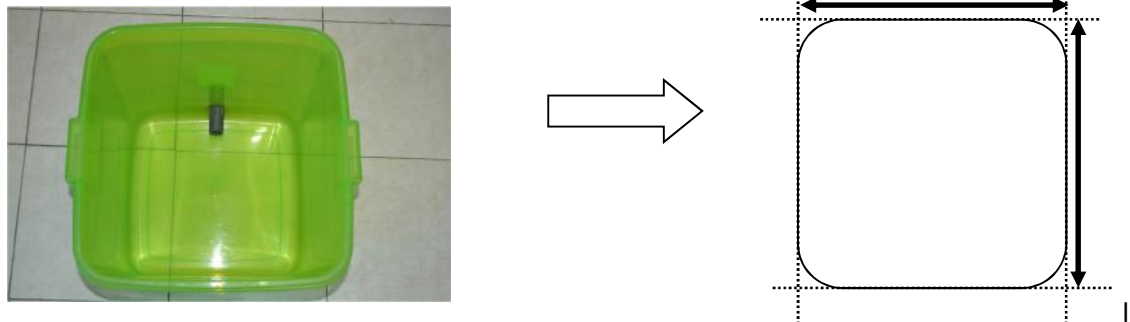
Tabel 4.3. Perbedaan teknik prediksi banjir sungai dan bendungan.

Sungai	Bendungan
- Perhitungan prediksi ketinggian untuk setiap daerah rawan berbeda.	- Perhitungan prediksi ketinggian bisa berlaku untuk beberapa daerah rawan berbeda sekaligus.
- Perhitungan debit air sungai menggunakan flowmeter sangat sulit karena harus mempertimbangkan kondisi air yang akan diukur kecepatannya.	- Perhitungan debit air sungai menggunakan flowmeter lebih mudah karena kondisi air yang akan diukur kecepatannya sudah diketahui.

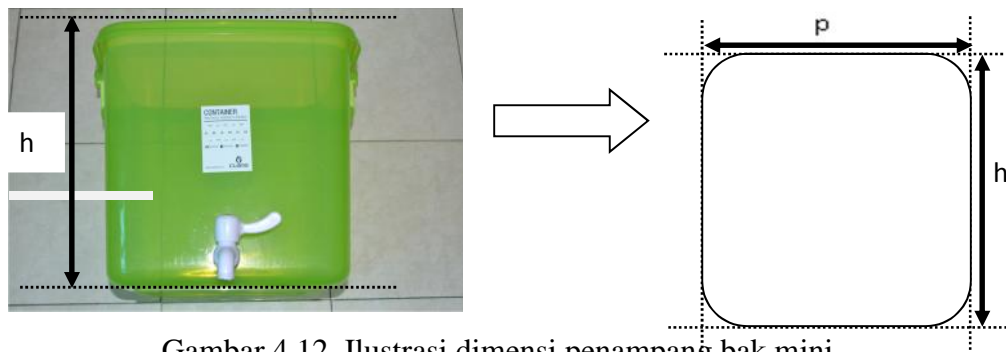
Dengan mempertimbangkan hal tersebut maka dalam pengujian prototype sistem ini kami menggunakan sebuah bak mini sebagai analogi dari sebuah bendungan seperti terlihat pada gambar 4.10. Perhitungan volume dari bak mini sebagaimana terlihat pada gambar 4.11 dapat dilakukan dengan memperhatikan gambar 4.12.



Gambar 4.10. Bak mini pengujian prediksi banjir (a) Tampak Atas (b) Tampak Samping



Gambar 4.11. Tampak atas ilustrasi penampang bak mini



Gambar 4.12. Ilustrasi dimensi penampang bak mini

Berdasarkan ilustrasi sebagaimana terlihat pada gambar 4.7 dan 4.8 maka didapat dimensi bak mini dengan data sebagai berikut :

- Panjang (p) = 35,5 cm
- Lebar keseluruhan (l_1) = 35,5 cm
- Tinggi keseluruhan bak (h) = 36 cm

Untuk mendapatkan nilai ketinggian permukaan air pada bak maka kita akan menghitung selisih jarak yang diukur sensor ultrasonik dengan tinggi keseluruhan bak dengan persamaan :

$$t_{ma} = h - \text{dist} \quad \dots (4.3)$$

Dimana :

t_{ma} = tinggi permukaan air (cm)

h = tinggi keseluruhan bak (cm)

$dist$ = jarak yang terukur oleh sensor ultrasonik (cm)

Berdasarkan data tinggi keseluruhan bak mini (h) 36 cm maka penulis mengklasifikasikan status siaga dengan rincian :

- Kondisi normal

Kondisi ini berlaku pada ketinggian permukaan air 0 cm – 5 cm. Dengan demikian pada saat pembacaan nilai tma 5 cm maka status ketinggian air berada pada kondisi normal. Jika status normal ini diterjemahkan ke dalam program C maka susunan programnya adalah :

...

```
if ( tma <= 5 ) {
```

```
if ( normal == 0 ) {
```

```
    PORTC.F1 = 1; // Indikator bahaya OFF
```

```
    PORTC.F3 = 0; // Indikator kondisi kembali normal dari siaga / bahaya
```

```
    PORTD.F2 = 0; // Buzzer OFF
```

```
    normal = 1;
```

...

List program ini memberikan pilihan pada kondisi ketinggian air maksimal 5 cm maka status berada pada kondisi normal tanpa bunyi buzzer atau sms peringatan banjir melainkan sms bahwa kondisi telah kembali normal

- Kondisi siaga 1

Kondisi ini berlaku pada ketinggian 5 cm – 10 cm. Kondisi ini diterjemahkan dengan baris program :

...

```
if ( tma > 5 && tma <= 10 )
```

```
    frek = 80; // durasi buzzer ~ 1 detik
```

...

Dengan memperhatikan baris program diatas jika ketinggian air berada di antara 5 – 10 cm maka sistem akan membunyikan buzzer dengan periode 1 detik untuk setiap bunyi buzzer.

- Kondisi siaga 2.

Kondisi ini berlaku pada ketinggian 10 cm – 15 cm. Kondisi ini diterjemahkan dengan baris program :

...

```
if ( tma > 10 && tma <= 15 )
```

```
    frek = 40;    // durasi buzzer ~ 500 ms
```

...

Dengan memperhatikan baris program diatas jika ketinggian air berada di antara 10 – 15 cm maka sistem akan membunyikan buzzer dengan periode 500 milidetik untuk setiap bunyi buzzer.

- Kondisi siaga 3.

Kondisi ini berlaku pada ketinggian 15 cm – 20 cm. Kondisi ini diterjemahkan dengan baris program :

...

```
if ( tma > 15 && tma <= 20 )
```

```
    frek = 20;    // durasi buzzer ~ 250 ms
```

...

Dengan memperhatikan baris program diatas jika ketinggian air berada di antara 15 – 20 cm maka sistem akan membunyikan buzzer dengan periode 250 milidetik untuk setiap bunyi buzzer.

- Kondisi berbahaya.

Kondisi ini berlaku pada ketinggian lebih dari 20 cm. Kondisi ini diterjemahkan dengan baris program :

...

```
else if ( tma > 25 )
```

```
    frek = 8;     // durasi buzzer ~ 100 ms
```

...

Dengan memperhatikan baris program diatas jika ketinggian air lebih dari 20 cm maka sistem akan membunyikan buzzer dengan periode 100 milidetik untuk setiap bunyi buzzer.

Selain defenisi status siaga yang didefenisikan sesuai uraian diatas kami juga mendefenisikan sistem prediksi berdasarkan debit air dan volume bendungan mini yang dibuat dimana untuk informasi debit didapat dari flowmeter. Sedangkan luas penampang dan volume bendungan mini dihitung berdasarkan data dimensi yang telah dijelaskan diatas. Melihat kesamaan antara panjang dan lebar bendungan mini maka dapat

disimpulkan bahwa penampang bendungan membentuk persegi sehingga untuk luas penampang bendungan mini dihitung dengan rumus :

$$\begin{aligned} A &= p \times l \quad \dots (4.4) \\ &= \text{sisi} \times \text{sisi} \\ &= (\text{sisi})^2 \\ &= (35,5)^2 \\ &= 1260,25 \text{ cm}^2 \end{aligned}$$

Sedangkan volume keseluruhan bendungan mini dihitung dengan rumus :

$$\begin{aligned} V &= A \times h \quad \dots (4.5) \\ &= 1260,25 \text{ cm}^2 \times 36 \text{ cm} \\ &= 45.369 \text{ cm}^3 \\ &= 45,37 \text{ liter} \end{aligned}$$

Berdasarkan perhitungan debit air dan volume bendungan mini sebagaimana dijelaskan diatas maka kondisi banjir dapat diprediksi dengan persamaan :

$$T = \frac{V}{\text{debit}} \quad \dots (4.6)$$

4.13 Pengujian Program SMS Melalui Modem GSM

Sebagaimana dijelaskan bab metodologi penelitian bagian perancangan sistem selain peringatan melalui buzzer juga ada peringatan melalui layanan *Short Message System* (SMS). Peringatan melalui layanan SMS bertujuan untuk menginformasikan status ketinggian air dan perkiraan waktu terjadinya banjir melalui modem GSM yang mendukung fitur SMS Broadcast sehingga dapat mengirimkan SMS kepada banyak nomor tujuan dalam sekali pengiriman.

Dalam pemrogramannya perintah pengiriman SMS disusun dalam suatu fungsi yang khusus menangani masalah pengiriman SMS. Meskipun dalam perancangannya SMS dikirimkan pada setiap perubahan kondisi siaga namun pada implementasinya SMS hanya dikirimkan pada kondisi siaga 3 dan berbahaya. Sedangkan untuk kondisi siaga 1 dan siaga 2 hanya melalui bunyi sirine yang berbeda frekuensinya pada setiap kondisi siaga. Dalam pemrograman prototype ini penulis mendefenisikan fungsi pengiriman SMS dalam fungsi `sendsms`. Pengiriman SMS pada mikrokontroler diberikan melalui perintah `Usart_write_text`. Perintah awal pengiriman sms ditulis dengan perintah :

...

```
if ( status == 0 ) {
```

```
strcpy_c(buffer,msg00);
Usart_Write_Text(buffer);
}
```

...

Pada saat ketinggian permukaan air mencapai kondisi siaga 1 yaitu pada ketinggian air 10 cm – 15 cm sistem akan memberikan perintah pengiriman ke modem GSM melalui perintah :

...

```
if ( tma > 10 && tma <= 15 ) // SMS : siaga 1
    sendsms(1);
```

...

Sedangkan pada kondisi siaga 2 perintah sms diberikan menggunakan perintah :

...

```
else if ( tma > 15 && tma <= 20 ) // SMS : siaga 2
    sendsms(2);
```

...

Untuk kondisi siaga 3 pengiriman sms dilakukan melalui perintah :

...

```
else if ( status == 3 ) {
    Usart_Write_Text("Siaga 3");
}
```

...

Sedangkan untuk status berbahaya pengiriman sms dilakukan dengan perintah :

...

```
else if ( status == 4 ) {
    Usart_Write_Text("Bahaya");
}
```

...

Namun jika pada status ketinggian air dari status siaga kembali ke status normal pada ketinggian permukaan air 0 cm – 5 cm sistem akan mengirimkan pesan bahwa ketinggian air telah normal kembali. Pengiriman pesan status normal didefinisikan melalui potongan program :

...

```

const char msg00[] = "Normal Kembali";
...
if ( status == 0 ) {
    strcpy_c(buffer,msg00);
    Usart_Write_Text(buffer);
...
    sendsms(0);    // SMS : Kembali normal
    PORTC.F3 = 1;
...

```

Sebagaimana dijelaskan diatas disaming informasi status ketinggian air pada kondisi siaga 3 dan kondisi bahaya juga diberikan prediksi waktu kejadian banjir. Pada penelitian ini penulis memperhitungkan waktu kejadian berdasarkan volume dan flowrate (debit air). Berdasarkan perhitungan volume dan perhitungan debit air sebagaimana dijelaskann pada sub bab sebelumnya perkiraan waktu kejadian banjir dihitung melalui persamaan :

$$\text{waktu} = \text{volume} / \text{flowrate} \quad \dots (4.7)$$

Kemudian hasil perhitungan waktu tersebut dimasukkan dalam program dengan format perintah :

```

...
if ( status == 3 || status == 4 ) {
    Usart_Write_Text(". Banjir dlm ");
    IntToStr(waktu,buff);
    Usart_Write_Text(buff);
    Usart_Write_Text(" min");
}
delay_ms(500);
Usart_Write(26);
...

```

Untuk pengiriman SMS kepada user melalui modem GSM menggunakan klasifikasi sebagaimana dijelaskan diatas penulis menggunakan perrintah *sendsms* untuk setiap SMS kondisi yang ada. Satu diantara potongan perintah *sendsms* yang dimaksud adalah :

...

```
if ( tma > 20 && tma <= 25 )
```

```
  sendsms(3);
```

```
else if ( tma > 25 )
```

```
  sendsms(4);
```

...

Dengan menjalankan potongan program ini jika ketinggian air mencapai ketinggian antara 20 cm sampai dengan 25 cm maka sistem akan memerintahkan modem mengirimkan SMS peringatan siaga 3 berikut prediksi kapan banjir akan terjadi sesuai hasil perhitungan waktu sesuai persamaan 4.9. Saat melakukan pengiriman SMS mikrokontroler akan berkomunikasi dengan rangkaian konverter RS232. Komunikasi tersebut dapat dilihat melalui tegangan terukur pada pin *Transmitter* (Tx) board rangkaian konverter RS232 sebesar 5 V dan pada pin *Receiver* (Rx) sebesar $\pm 4,4$ V. Kondisi tersebut memungkinkan terjadinya komunikasi serial antara modem GSM dengan mikrokontroler PIC16F877A dengan perantara rangkaian konverter RS232 yang berfungsi sebagai penghubung antara minimum sistem mikrokontroler dengan modem GSM. Pada saat pengujian blok program SMS maka SMS akan terkirim kepada nomor yang telah diprogram pada *compiler* sebagaimana terlihat pada gambar 4.13 berikut.



(a)

(b)

Gambar 4.13. Hasil simulasi sistem (a) Ketinggian air yang diukur (b) Pesan singkat yang dikirimkan modem GSM ke nomor penerima

Pada saat komunikasi serial tersebut berlangsung tegangan yang terukur pada rangkaian dapat dilihat pada tabel 4.4. berikut ini.

Tabel 4.4. Tegangan kerja pin komunikasi dengan modem GSM.

Kondisi	Tegangan pin Transmitter Rangkaian	Tegangan pin Receiver Rangkaian	Tegangan pin Transmitter Rangkaian	Tegangan pin Receiver Rangkaian

	Konverter	Konverter	Mikrokontroler (pin C6)	Konverter (pin C7)
Normal	5 V	4,4 V	5 V	4,4 V
Siaga I	5 V	4,4 V	5 V	4,4 V
Siaga II	5 V	4,4 V	5 V	4,4 V
Siaga III	5 V	4,4 V	5 V	4,4 V
Bahaya	5 V	4,4 V	5 V	4,4 V

Dengan pengaturan pengiriman pesan sedemikian rupa maka pada kondisi siaga dan bahaya maka sistem akan mengirimkan pesan peringatan sesuai status ketinggian permukaan air di bendungan. Namun demikian jika pada saat pengujian pesan tersebut tidak terkirim ada beberapa faktor penyebab kegagalan pengiriman pesan tersebut. Beberapa penyebab kegagalan pengiriman pesan adalah :

1. Kesalahan pada penyusunan program.

Ada beberapa jenis kesalahan pemrograman sehingga program tidak berjalan sebagaimana mestinya yaitu :

a. Kesalahan penyusunan logika program

Dalam kasus ini program yang dibuat berfungsi dengan baik tetapi adanya penyusunan logika pemrograman pengiriman pesan yang kurang tepat dapat menyebabkan kesalahan komunikasi pada waktu siaga atau bahaya yang menyebabkan pesan peringatan yang masuk tidak sesuai dengan kondisi ketinggian permukaan air yang terjadi pada saat pesan peringatan dikirimkan. Misalnya pada saat ketinggian air berada pada kondisi siaga 1 atau siaga 2 namun karena logika yang kurang tepat seperti telah dijelaskan diatas pesan yang masuk bukan pesan peringatan siaga 1 atau siaga 2 melainkan pesan kondisi normal atau kondisi bahaya. Untuk menghindari kondisi seperti ini penyusunan logika pemrograman perlu diperhatikan dengan seksama sehingga kesalahan seperti ini tidak terjadi karena kesalahan peringatan pada implementasinya dapat berakibat fatal bagi masyarakat sehingga sistem jadi terkesan tidak berguna.

b. Kesalahan pemicuan pengiriman pesan.

Kondisi ini terjadi saat definisi parameter pemicu pengiriman pesan tidak sesuai dengan kondisi ketinggian permukaan air. Definisi dan penggunaan parameter

yang sesuai akan membuat sistem bekerja dengan lebih akurat. Kesalahan pada poin ini menyebabkan pesan peringatan tidak terkirim sesuai kondisi yang terjadi.

c. Kesalahan saat proses *download* program.

Kesalahan ini terjadi disebabkan kegagalan proses *download* program atau proses *download* yang tidak berjalan dengan sempurna sehingga ada bagian program yang tidak masuk ke *chip* mikrokontroler saat program didownloadkan. Jika ada bagian program yang tidak masuk saat proses *download* maka sistem tidak akan berjalan dengan sempurna.

2. Rentang status ketinggian terlalu pendek.

Kondisi ini terjadi jika debit air yang masuk menyebabkan kenaikan ketinggian permukaan air yang cepat dalam waktu singkat. Rentang kondisi siaga terlalu pendek dan tidak seimbang dengan laju debit air yang masuk sehingga terjadi dua atau lebih komunikasi pada waktu yang hampir bersamaan. Kondisi ini dapat menyebabkan tumpang tindihnya pengiriman pesan peringatan kondisi siaga tersebut.

3. Adanya gangguan jaringan komunikasi pada saat pengiriman pesan sehingga pesan tidak sampai kepada penerima.

4.14 Pengujian Buzzer

Sebagaimana digambarkan pada gambar 4.1 peringatan bencana banjir diberikan dalam dua bentuk peringatan yaitu peringatan melalui SMS dan peringatan melalui sirine / *buzzer*. *Buzzer* akan berbunyi ketika ketinggian air mencapai kondisi siaga atau berbahaya. Untuk memudahkan pendengar dalam mengidentifikasi status bahaya yang diberikan frekuensi bunyi *buzzer* akan berbeda sesuai *level* siaga yang muncul. Untuk lengkapnya perhatikan potongan program berikut :

...

```
if ( tma > 10 && tma <= 15 )
    frek = 80; // durasi buzzer ~ 1 detik
else if ( tma > 15 && tma <= 20 )
    frek = 40; // durasi buzzer ~ 500 ms
else if ( tma > 20 && tma <= 25 )
    frek = 20; // durasi buzzer ~ 250 ms
else if ( tma > 25 )
    frek = 8; // durasi buzzer ~ 100 ms
```

...

Pada potongan program diatas *buzzer* akan berbunyi dengan frekuensi berbeda pada setiap kondisi siaga. Pada ketinggian air berada diantara 10 cm sampai dengan 15 cm atau kondisi siaga 1 maka *buzzer* akan berbunyi dengan durasi 1 sekon. Bunyi *buzzer* akan meningkat dengan durasi 500 ms atau dua bunyi per detik jika ketinggian air berada diantara 15 cm sampai dengan 20 cm atau kondisi siaga 2. Pada ketinggian air antara 20 cm sampai 25 cm atau kondisi siaga 3 frekuensi bunyi *buzzer* meningkat menjadi 250 ms untuk setiap bunyi *buzzer* atau sekitar empat bunyi dalam 1 detik. Sedangkan pada ketinggian air diatas 25 cm atau status berbahaya maka *buzzer* akan berbunyi dengan durasi 100 ms untuk setiap bunyi *buzzer* atau sekitar 10 bunyi *buzzer* per detik. Bunyi *buzzer* akan berbeda untuk setiap kondisi sesuai klasifikasi kelompok ketinggian air yang ditetapkan. Dengan adanya perbedaan durasi bunyi buzzer pada setiap kondisi siaga maka pendengar dapat membedakan kapan kondisi siaga 1, kondisi siaga 2, kondisi siaga 3 dan kondisi berbahaya berdasarkan durasi bunyi *buzzer* tersebut. Pada saat *buzzer* berbunyi tanda kondisi siaga tegangan yang terukur dapat dilihat pada tabel 4.5 berikut ini.

Tabel 4.5. Tegangan kerja pin *buzzer*.

Kondisi	Tegangan pada pin <i>buzzer</i>
Siaga I	2,4 V
Siaga II	2,4 V
Siaga III	2,4 V
Bahaya	2,4 V

BAB V

PENUTUP

5.3 Kesimpulan

Adapun tujuan dari tugas akhir ini adalah merancang prototype sistem peringatan dini bencana banjir yang dapat memberitahukan secara dini datangnya musibah banjir berikut prediksi kapan musibah tersebut akan datang. Hasil perancangan sistem yang diimplementasikan dalam bentuk contoh asli sistem peringatan dini telah bekerja dengan baik sesuai dengan perancangan yang direncanakan. Peringatan bencana diberikan dalam bentuk pesan singkat peringatan bencana banjir kepada masyarakat di daerah rawan banjir dan buzzer sebagai media peringatan lain yang berfungsi memberikan peringatan langsung jika pesan peringatan bencana tidak terkirim ke nomor tujuan.

5.4 Saran

1. Bagi pihak yang berminat melanjutkan penelitian ini dapat membuat sistem monitoring bendungan menggunakan *website* sebagai alternatif dalam prediksi bencana banjir.
2. Sistem dapat dikembangkan dengan menambahkan sistem pendaftaran nomor penerima pesan peringatan dan modifikasi sistem sehingga penerima pesan bisa disesuaikan dengan kondisi siaga yang ada.
3. Belum adanya perangkat atau sistem keamanan bagi sistem ini dapat menjadi ide dalam perancangan selanjutnya mengingat adanya kemungkinan perusakan sistem baik secara perangkat atau program oleh orang yang tidak bertanggung jawab.

DAFTAR PUSTAKA

- Aditya Pradhana. *Peringatan Dini Bahaya Banjir Di Tambang Pasir Berbasis SMS Gateway*. Semarang: Universitas Diponegoro. 2011.
- Anonim. (2011). *Flow meter*. [Online]. Available: <http://www.wmablog.com/2011/11/flow-meter.html>.
- Anonim. *SMS (Short Message Service) -Technical Overview*. <http://educyclopedia.karadimov.info/library/SMS.pdf>.
- Agus Bejo. (2007). *Landasan Teori*. [Online]. Available: <http://repository.usu.ac.id/bitstream/123456789/20194/4/Chapter%20II.pdf>.
- Chris Warner. (2009). *Hall-effect Sensors Measure Up*. [Online]. Available: <http://www.ecnmag.com/articles/2009/03/sensor-zone-april-2009>.
- D-D-S.nl. *M2M Modem Fastrack GSM/GPRS Modem M1306B*. Wavecom.
- Dr. Ir. Harry Soekotjo Dachlan. M.Sc Dkk. *Sistem Penunjang Keputusan Deteksi Dini Bahaya Banjir Secara Real Time Menggunakan Data Logger Berbasis GSM*. Malang: Universitas Brawijaya. 2010.
- Creswell. John W. *Research design. Qualitative. Quantitative. and Mixed Methods Approaches*. SAGE. 2003.
- Husanto & Thomas. *Kupas Tuntas Mikrokontroler PIC16F84A*. Yogyakarta. Penerbit Andi. 2008.
- Venoruton Budi Irmawan. *Deteksi Ketinggian Air Dengan Media Gelombang Radio Berbasis Mikrokontroler AT89S51*. Surakarta: Universitas Muhammadiyah. 2007.
- Joni Gunawan. Kartika Firdausy dan Wahyu Sapto Aji. *Otomatisasi Pendeteksi Dini Bahaya Banjir Menggunakan SMS Berbasis Mikrokontroler AT89S52*. Yogyakarta: Universitas Ahmad Dahlan.
- Microchip. *PIC16F87XA Datasheet*. Microchip Technology Inc. 2003.
- Mikroelektronika. *PIC Microcontrollers: Programming In C*. [Online]. Available: <http://www.mikroe.com/chapters/view/15/chapter-2-programming-microcontrollers>.
- Seed Studio Works. *Seed Ultrasonic Sensor*. 2010.
- Seed Studio Works. *Water Flow Sensor*. 2010.
- Seruu.com. (Desember 2011). *Ratusan Rumah di Kampar Terendam Banjir*. [Online]. Available: <http://www.seruu.com/utama/politik/artikel/ratusan-rumah-di-kampar-terendam-banjir>.

Tribunnews.com. (November 2011). *Korban Jiwa dan Kerusakan Akibat Banjir Bandang Riau*. <http://www.tribunnews.com/2011/11/28/korban-jiwa-dan-kerusakan-akibat-banjir-bandang-raiu>. diakses 12 Juni 2012.

tvOneNews. (Desember 2011). *Banjir Masih Rendam 1.151 Rumah Warga di Kampar Riau*. [Online]. Available: http://nusantara.tvonenews.tv/berita/view/52291/2011/12/27/banjir_masih_rendam_1_151_rumah_warga_di_kampar_riau.tvOne.