



**SISTEM PAKAR MENDETEKSI KERUSAKAN
MIKROKONTROLER ATMEGA328 MENGGUNAKAN
METODE CASE BASE REASONING (CBR)**

SKRIPSI

DISUSUN OLEH :

NAMA : M. WAHYUDI HUTABARAT
N.P.M : 1724370837
PROGRAM STUDI : SISTEM KOMPUTER

**PROGRAM STUDI SISTEM KOMPUTER
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS PEMBANGUNAN PANCA BUDI
2021**

PENGESAHAN SKRIPSI

JUDUL : SISTEM PAKAR MENDETEKSI KERUSAKAN MIKROKONTROLER
ATMEGA328 MENGGUNAKAN METODE CASE BASE REASONING (CBR)

NAMA : M. WAHYUDI HUTABARAT
N.P.M : 1724370837
FAKULTAS : SAINS & TEKNOLOGI
PROGRAM STUDI : Sistem Komputer
TANGGAL KELULUSAN : 09 April 2021

DIKETAHUI

DEKAN

KETUA PROGRAM STUDI



Hamdani, ST., MT.



Eko Hariyanto, S.Kom., M.Kom

DISETUJUI

KOMISI PEMBIMBING

PEMBIMBING I

PEMBIMBING II



Solly Aryza, ST.,M.Eng



Jodi Hendrawan, S.Kom., M.Kom



UNIVERSITAS PEMBANGUNAN PANCA BUDI

FAKULTAS SAINS & TEKNOLOGI

Jl. Jend. Gatot Subroto Km 4,5 Medan Fax. 061-8458077 PO.BOX : 1099 MEDAN

PROGRAM STUDI TEKNIK ELEKTRO
 PROGRAM STUDI ARSITEKTUR
 PROGRAM STUDI SISTEM KOMPUTER
 PROGRAM STUDI TEKNIK KOMPUTER
 PROGRAM STUDI AGROTEKNOLOGI
 PROGRAM STUDI PETERNAKAN

(TERAKREDITASI)
 (TERAKREDITASI)
 (TERAKREDITASI)
 (TERAKREDITASI)
 (TERAKREDITASI)
 (TERAKREDITASI)

PERMOHONAN JUDUL TESIS / SKRIPSI / TUGAS AKHIR*

Saya yang bertanda tangan di bawah ini :

Nama Lengkap

: M. WAHYUDI HUTABARAT

Tempat/Tgl. Lahir

: MEDAN / 18 Agustus 1997

Nomor Pokok Mahasiswa

: 1724370837

Program Studi

: Sistem Komputer

Konsentrasi

: Sistem Kendali Komputer

Jumlah Kredit yang telah dicapai

: 112 SKS, IPK 3.12

Nomor Hp

: 081397297531

Dengan ini mengajukan judul sesuai bidang ilmu sebagai berikut :

No.	Judul
1.	Sistem Pakar Mendeteksi Kerusakan Mikrokontroler ATmega328 menggunakan Metode Case Base Reasoning (CBR)

Catatan : Diisi Oleh Dosen Jika Ada Perubahan Judul

*Coret Yang Tidak Perlu



Rektor I,

(Cahyo Pramono, S.E., M.M.)

Medan, 12 Oktober 2020

Pemohon,

(M. Wahyudi Hutabarat)

Tanggal :

Disahkan oleh :
 Dekan

(Hamdani, ST., MT)

Tanggal :

Disetujui oleh :
 Dosen Pembimbing I :

(Solly Aryza, ST., M.Eng)

Tanggal :

Disetujui oleh:
 Ka. Prodi Sistem Komputer

(Eko Hariyanto, S.Kom., M.Kom)

Tanggal :

Disetujui oleh:
 Dosen Pembimbing II:

(Jodi Hendrawan, S.Kom., M.Kom)

UNIVERSITAS PEMBANGUNAN PANCA BUDI

JL. Jend. Gatot Subroto KM 4,5 PO. BOX 1099 Telp. 061-30106057 Fax. (061) 4514808
 MEDAN - INDONESIA
 Website : www.pancabudi.ac.id - Email : admin@pancabudi.ac.id

LEMBAR BUKTI BIMBINGAN SKRIPSI

Mahasiswa : M. WAHYUDI HUTABARAT
 : 1724370837
 Studi : Sistem Komputer
 : Strata Satu
 Pembimbing : Solly Aryza, ST.,M.Eng
 Skripsi : Sistem Pakar Mendeteksi Kerusakan Mikrokontroler ATmega328 menggunakan Metode Case Base Reasoning (CBR)

	Pembahasan Materi	Status	Keterangan
	bab 2 mana penelitian terdahulunya	Revisi	
	acc bab 2 lanjut bab 3	Revisi	
	acc bab3 lanjut bab 4	Revisi	
	acc seminar hasil	Disetujui	
	perbaiki semua dia seminar hasil	Revisi	
	apabila salah juga dan tidak diperbaiki waktu sseminar kemarin maka sidang dibatalkan	Disetujui	
	acc sidan	Disetujui	
	ACC jilid	Disetujui	

Medan, 27 Mei 2021
 Dosen Pembimbing,



Solly Aryza, ST.,M.Eng

UNIVERSITAS PEMBANGUNAN PANCA BUDI

JL. Jend. Gatot Subroto KM 4,5 PO. BOX 1099 Telp. 061-30106057 Fax. (061) 4514808

MEDAN - INDONESIA

Website : www.pancabudi.ac.id - Email : admin@pancabudi.ac.id**LEMBAR BUKTI BIMBINGAN SKRIPSI**

Mahasiswa : M. WAHYUDI HUTABARAT
 NPM : 1724379837
 Program Studi : Sistem Komputer
 Tingkat Pendidikan : Strata Satu
 Dosen Pembimbing : Jodi Hendrawan, S.Kom., M.Kom
 Judul Skripsi : Sistem Pakar Mendeteksi Kerusakan Mikrokontroler ATmega328 menggunakan Metode Case Base Reasoning (CBR)

Tanggal	Pembahasan Materi	Status	Keterangan
14 Desember 2020	Acc bab II dan III	Revisi	
17 Desember 2020	Acc seminar hasil	Disetujui	
1 Januari 2021	Penulisan abstack, kata pengantar, daftar isi, daftar tabel, daftar gambar, belum sesuai dengan ketentuan panduan seperti spasi yg digunakan	Revisi	
1 Januari 2021	Kemudian pada laporan masih mencantumkan "gejala" sedangkan judul skripsi tidak membahas penyakit. Silahkan disesuaikan antara laporan, system dan penelitiannya	Revisi	
1 Januari 2021	Acc sidang meja hijau	Disetujui	
1 April 2021	perbaiki sub bab laporan tidak sesuai dengan panduan	Revisi	
1 Mei 2021	Silahkan perbaiki seperti nama2 rektor dan sterusnya. Acc jilid	Disetujui	
1 Mei 2021	Acc jilid	Disetujui	

Medan, 27 Mei 2021
 Dosen Pembimbing,



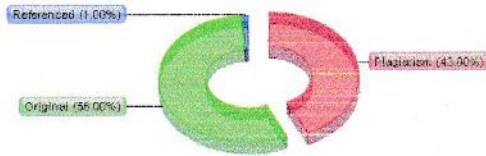
Jodi Hendrawan, S.Kom., M.Kom

Plagiarism Detector v. 1740 - Originality Report 1/25/2021 8:37:43 AM

Document: M. WAHYUDI HUTABARAT [1724370837](#) SISTEM KOMPUTER.docx Licensed to: Universitas Pembangunan Panca Budi_License04
Comparison Preset: Rewrite. Detected language: Indonesian



Relation chart:



Distribution graph:



**SURAT BEBAS PUSTAKA
NOMOR: 3211/PERP/BP/2020**

Perpustakaan Universitas Pembangunan Panca Budi menerangkan bahwa berdasarkan data pengguna perpustakaan saudara/i:

: M. WAHYUDI HUTABARAT

: 1724370837

Semester : Akhir

: SAINS & TEKNOLOGI

Studi : Sistem Komputer

Yang terhormat, sejak tanggal 26 Oktober 2020, dinyatakan tidak memiliki tanggungan dan atau pinjaman buku yang tidak lagi terdaftar sebagai anggota Perpustakaan Universitas Pembangunan Panca Budi Medan.

Medan, 26 Oktober 2020

Diketahui oleh,
Kepala Perpustakaan


Sugiarjo, S.Sos., S.Pd.I

Dokumen : FM-PERPUS-06-01

: 01

Tanggal : 04 Juni 2015

KARTU BEBAS PRAKTIKUM
Nomor. 1435/BL/LAKO/2020

anda tangan dibawah ini Ka. Laboratorium Komputer dengan ini menerangkan bahwa :

Semester : M. WAHYUDI HUTABARAT
: 1724370837
: Akhir
Prodi : SAINS & TEKNOLOGI
: Sistem Komputer

elah menyelesaikan urusan administrasi di Laboratorium Komputer Universitas Pembangunan Panca Budi Medan.

Medan, 28 Mei 2021
Ka. Laboratorium



Sari Panjaitan, S. Kom., M.Kom.

an : FM-LAKO-06-01

Revisi : 01

Tgl. Efektif : 04 Juni 2015

SURAT PERNYATAAN TIDAK PLAGIAT

Saya yang bertanda tangan dibawah ini :

NAMA : M. WAHYUDI HUTABARAT
NPM : 1724370837
PROGAM STUDI : SISTEM KOMPUTER
JUDUL SKRIPSI : SISTEM PAKAR MENDETEKSI KERUSAKAN
MIKROKONTROLER ATMEGA328MENGUNAKAN
METODE CASE BASE REASONING (CBR)

Dengan ini menyatakan bahwa :

1. Skripsi ini merupakan hasil karya tulis saya sendiri dan bukan merupakan hasil karya orang lain (plagiat).
2. Memberi izin hak bebas Royalti Non-Eksklusif kepada Universitas Pembangunan Pancabudi untuk menyimpan, mengalih-media/formatkan, mengola, mendistribusikan dan mempublikasikan karya skripsinya melalui internet atau media lain bagi kepentingan akademis.

Pernyataan ini saya buat dengan penuh tanggung jawab dan saya bersedia menerima konsekuensi apapun sesuai dengan peraturan yang berlaku apabila di kemudian hari di ketahui bahwa pernyataan ini tidak benar.

Medan, 1 juni 2021



M. Wahyudi Hutabrat
NPM. 1724370837

Permohonan Meja Hijau

Medan, 28 Mei 2021
 Kepada Yth : Bapak/Ibu Dekan
 Fakultas SAINS & TEKNOLOGI
 UNPAB Medan
 Di -
 Tempat

Yang hormat, saya yang bertanda tangan di bawah ini :

Nama : M. WAHYUDI HUTABARAT
 Tempat/Tgl. Lahir : Sunggal Kanan / 18 Agustus 1997
 Nama Orang Tua : rusman hutabarat
 NIM : 1724370837
 Fakultas : SAINS & TEKNOLOGI
 Program Studi : Sistem Komputer
 NPM : 081397297531
 Alamat : Dusun II jln setia makmur no.8

Yang bermohon kepada Bapak/Ibu untuk dapat diterima mengikuti Ujian Meja Hijau dengan judul Sistem Pakar Mendeteksi Kerusakan Kontroler ATmega328 menggunakan Metode Case Base Reasoning (CBR), Selanjutnya saya menyatakan :

1. Melampirkan KKM yang telah disahkan oleh Ka. Prodi dan Dekan
2. Tidak akan menuntut ujian perbaikan nilai mata kuliah untuk perbaikan indek prestasi (IP), dan mohon diterbitkan ijazahnya setelah lulus ujian meja hijau.
3. Telah tercap keterangan bebas pustaka
4. Terlampir surat keterangan bebas laboratorium
5. Terlampir pas photo untuk ijazah ukuran 4x6 = 5 lembar dan 3x4 = 5 lembar Hitam Putih
6. Terlampir foto copy STTB SLTA dilegalisir 1 (satu) lembar dan bagi mahasiswa yang lanjutan D3 ke S1 lampirkan ijazah dan transkripnya sebanyak 1 lembar.
7. Terlampir pelunasan kwintasi pembayaran uang kuliah berjalan dan wisuda sebanyak 1 lembar
8. Skripsi sudah dijilid lux 2 exemplar (1 untuk perpustakaan, 1 untuk mahasiswa) dan jilid kertas jeruk 5 exemplar untuk penguji (bentuk dan warna penjiilidan diserahkan berdasarkan ketentuan fakultas yang berlaku) dan lembar persetujuan sudah di tandatangani dosen pembimbing, prodi dan dekan
9. Soft Copy Skripsi disimpan di CD sebanyak 2 disc (Sesuai dengan Judul Skripsinya)
10. Terlampir surat keterangan BKKOL (pada saat pengambilan ijazah)
11. Setelah menyelesaikan persyaratan point-point diatas berkas di masukan kedalam MAP
12. Bersedia melunaskan biaya-biaya uang dibebankan untuk memproses pelaksanaan ujian dimaksud, dengan perincian sbb :

1. [102] Ujian Meja Hijau	: Rp.	
2. [170] Administrasi Wisuda	: Rp.	
3. [202] Bebas Pustaka	: Rp.	
4. [221] Bebas LAB	: Rp.	
Total Biaya	: Rp.	0

Ukuran Toga :

M

Diketahui/Disetujui oleh :

Hormat saya



M. Wahyudi Hutabarat,
 Fakultas SAINS & TEKNOLOGI



M. WAHYUDI HUTABARAT
 1724370837

Yang :

1. Surat permohonan ini sah dan berlaku bila ;
 - o a. Telah dicap Bukti Pelunasan dari UPT Perpustakaan UNPAB Medan.
 - o b. Melampirkan Bukti Pembayaran Uang Kuliah aktif semester berjalan
2. Dibuat Rangkap 3 (tiga), untuk - Fakultas - untuk BPAA (asli) - Mhs.ybs.

ABSTRAK

M. WAHYUDI HUTABARAT

**SISTEM PAKAR MENDETEKSI KERUSAKAN MIKROKONTROLER
ATMEGA328 MENGGUNAKAN METODE CASE BASE REASONING
(CBR)
2020**

ATMega328 adalah mikrokontroler keluaran dari atmel yang mempunyai arsitektur RISC (*Reduce Instruction Set Computer*) yang dimana setiap proses eksekusi data lebih cepat dari pada arsitektur CISC (*Completed Instruction Set Computer*). Dalam hal ini, pada mikrokontroler *ATMega328* sering sekali terjadi kerusakan yang sangat berpengaruh pada struktur dari sistem. Sering terjadi ketika sebuah alat sudah dirakit dan rekatkan oleh solder, *ATMega321* menjadi error sehingga struktur ataupun rangkaian yang sudah direkatkan harus dibongkar kembali. Berdasarkan dari masalah tersebut penulis ingin membuat sistem pakar yang dapat mendiagnosa kerusakan pada *ATMega328* menggunakan metode *case base reasoning* (CBR). Aplikasi ini berbasis pemrograman dekstop dengan memanfaatkan sistem pakar metode *Case-Based Reasoning*. Dengan metode *Case-Based Reasoning* yang diimplementasikan di dalam Aplikasi mendiagnosa kerusakan pada *ATMega328*. Hasil yang diharapkan dari penelitian ini yaitu aplikasi ini dapat mendiagnosa kerusakan pada *ATMega328* diharapkan dapat membantu para masyarakat untuk mendiagnosa kerusakan pada *ATMega328*.

Kata Kunci : Case-Based Reasoning, Diagnosa *ATMega328*, Sistem Pakar.

DAFTAR ISI

KATA PENGANTAR.....	iv
DAFTAR ISI.....	v
DAFTAR GAMBAR.....	vii
DAFTAR TABEL	viii
DAFTAR LAMPIRAN	ix
DAFTAR ISTILAH	x
BAB I PENDAHULUAN	
1.1. Latar Belakang	1
1.2. Rumusan Masalah	2
1.3. Batasan Masalah	3
1.4. Tujuan Penelitian	3
1.5. Manfaat Penelitian	4
BAB II LANDASAN TEORI	
2.1. Sistem Pakar.....	5
2.1.1. Manfaat Sistem Pakar	5
2.1.2. Kelebihan Sistem Pakar Dan Kelemahan Sistem Pakar	6
2.1.3. Ciri-Ciri Sitem Pakar	8
2.1.4. Konsep Dasar Sistem Pakar	8
2.1.5. Struktur Sistem Pakar.....	11
2.2. Teori <i>Case Base Reasoning</i>	14
2.3. Atmega328.....	17
2.4. Pemodelan.....	18
2.5. UML (<i>Unified Modeling Language</i>).....	19
2.5.1. <i>Use Case Diagram</i>	19
2.5.2. <i>Activity Diagram</i>	20
2.5.3. <i>Class Diagram</i>	22
2.6. <i>Flowchart</i>	23
2.6.1. <i>Flow Direction Symbols</i>	24
2.6.2. <i>Processing Symbols</i>	25
2.6.3. <i>Input-Output Symbols</i>	26
2.7. Aplikasi Pendukung	26
2.7.1. <i>Visual Basic NET</i>	26
2.7.2. <i>Crystal Report</i>	32
2.7.3. <i>Microsoft Acces 2010</i>	33
2.8. Penelitian Sebelumnya.....	35
BAB III METODE PENELITIAN	
3.1. Metode Penelitian	41
3.2. Analisa Permasalahan	42
3.3. Algoritma Sistem	42

3.3.1. Menentukan Sumber Pengetahuan.....	42
3.3.2. Nilai Densitas.....	43
3.4. <i>Flowchart</i> Program	46
3.5. Perancangan Sistem	47
3.5.1. <i>Use Case Diagram</i>	47
3.5.2. <i>Activity Diagram</i>	50
3.5.3. <i>Class Diagram</i>	51
3.6. Rancangan Basis Data.....	52
3.7. Perancangan Antarmuka (<i>Interface</i>)	54
 BAB IV IMPLEMENTASI DAN PENGUJIAN	
4.1. Kebutuhan Sistem	59
4.1.1. Kebutuhan Perangkat Keras.....	59
4.1.2. Kebutuhan Perangkat Lunak.....	59
4.2. Implementasi Sistem	59
4.2.1. <i>Login</i>	60
4.2.2. <i>Menu Utama</i>	60
4.2.3. <i>Form</i> Data Kerusakan	61
4.2.4. <i>Form</i> Data Kasus	62
4.2.5. <i>Form</i> Data Pemilik Atmega328.....	62
4.2.6. <i>Form</i> Diagnosa Kerusakan	63
4.2.7. <i>Form</i> Proses Diagnosa	64
4.2.8. <i>Form</i> Hasil Diagnosa	64
4.3. Kelebihan Dan Kelemahan Sistem	65
 BAB V KESIMPULAN DAN SARAN	
5.1. Kesimpulan	67
5.2. Saran.....	67
 DAFTAR PUSTAKA	
BIOGRAFI PENULIS	
LAMPIRAN-LAMPIRAN	

DAFTAR GAMBAR

Gambar 2.1 Komponen-Komponen Sistem Pakar	12
Gambar 2.2 Tampilan <i>Visual Basic</i>	28
Gambar 2.3 Mengaktifkan <i>IDE Visual Basic2008</i>	29
Gambar 2.4 Mengaktifkan <i>IDE Visual Basic 2008</i>	29
Gambar 2.5 <i>Dialog Box New Project</i> ditampilkan sesaat anda menjalankan IDE Jendela <i>Visual Basic 2008</i>	30
Gambar 2.6 Jendela <i>interface Visual Basic 2008</i>	31
Gambar 2.7 Tampilan <i>Menu Utama crystal Report</i>	33
Gambar 2.8 Tampilan <i>Access 2010</i>	35
Gambar 3.1 Tahapan Penelitian	41
Gambar 3.2 Pohon Keputusan Rusak Perbaikan (P1).....	45
Gambar 3.3 Pohon Keputusan Rusak Parah	45
Gambar 3.4 <i>Flowchart Program Menggunakan Dempster-Shafer</i>	46
Gambar 3.5 <i>Use Case Diagram</i>	47
Gambar 3.6 <i>Activity Diagram Admin</i>	51
Gambar 3.7 <i>Class Diagram</i> Resiko Menggunakan <i>ATMega328</i>	52
Gambar 3.8 <i>Form Menu Login</i>	54
Gambar 3.9 <i>Form Menu Utama</i>	55
Gambar 3.10 <i>Form Kerusakan</i>	55
Gambar 3.11 <i>Form Kerusakan</i>	56
Gambar 3.12 <i>Form Rule</i>	57
Gambar 3.13 <i>Form Diagnosa</i>	58
Gambar 3.14 <i>Form Hasil Diagnosa</i>	58
Gambar 4.1 <i>Form Login</i>	60
Gambar 4.2 <i>Form Menu Utama</i>	61
Gambar 4.3 <i>Form Data Kerusakan</i>	61
Gambar 4.4 <i>Form Data Kasus</i>	62
Gambar 4.5 <i>Form Data Pemilik ATMEGA328</i>	63
Gambar 4.6 <i>Form Diagnosa Kerusakan</i>	63
Gambar 4.7 <i>Form Proses Diagnosa Kerusakan Pada ATMEGA328</i>	64
Gambar 4.8 <i>Form Hasil Diagnosa Kerusakan Pada ATMEGA328</i>	65

DAFTAR TABEL

Tabel 2.1 Perbedaan antara Sistem Konvensional dengan Sistem Pakar.....	11
Tabel 2.2 Simbol-simbol <i>Use Case Diagram</i>	20
Tabel 2.3 Simbol-simbol <i>Activity Diagram</i>	21
Tabel 2.4 Simbol-simbol <i>Class Diagram</i>	22
Tabel 2.5 Simbol-simbol <i>Flow Direction Symbols</i>	24
Tabel 2.6 <i>Processing Symbols</i>	25
Tabel 2.7 <i>Input-Output Symbols</i>	26
Tabel 3.1 Jenis Resiko Penggunaan VGA	43
Tabel 3.2 Sumber Pengetahuan Kerusakan ATMega328	43
Tabel 3.3 Basis Aturan Resiko ATMega328	44
Tabel 3.4 Data Keputusan.....	44
Tabel 3.5 Sampel Kasus.....	46
Tabel 3.6 Tabel <i>Login</i>	53
Tabel 3.7 Tabel Kerusakan	53
Tabel 3.8 Tabel Kerusakan	53
Tabel 3.9 Tabel <i>Rule</i>	53
Tabel 3.10 Tabel Diagnosa	53
Tabel 3.11 Tabel Diagnosa (Lanjutan).....	54

DAFTAR LAMPIRAN

Lampiran 1. Lembar Pengesahan Tugas Akhir.....	L-1
Lampiran 2. Daftar Pustaka.....	L-2
Lampiran 3. Biografi Penulis	L-3
Lampiran 4. Lembar Bimbingan Doping 1	L-4
Lampiran 5. Lembar Bimbingan Doping 2.....	L-5
Lampiran 6. Surat Riset.....	L-6
Lampiran 7. Data Diri Pakar	L-7
Lampiran 8. <i>Listing Program</i>	L-8

DAFTAR ISTILAH

- Sistem Pakar** Sistem pakar adalah sistem yang menggunakan pengetahuan manusia dimana pengetahuan tersebut dimasukkan ke dalam sebuah komputer dan kemudian digunakan untuk menyelesaikan masalah yang membutuhkan kepakaran atau keahlian manusia.
- Casebase Reasoning** Case Base Reasoning adalah fakta-fakta berupa kasus-kasus sebelumnya yang pernah ada dan serangkaian alur untuk memeriksa, menghitung serta menyimpulkan suatu solusi dari permasalahan atau kasus yang baru tahapan pada Case Base Reasoning ada 4 yaitu: **Retrieve, Reuse, Revise,** dan **retain.**
- ATMega328** ATMega328 adalah mikrokontroler keluaran dari atmel yang mempunyai arsitektur RISC (*Reduce Instruction Set Computer*) yang dimana setiap proses eksekusi data lebih cepat dari pada arsitektur CISC (*Completed Instruction Set Computer*).

KATA PENGANTAR

Penulis mengucapkan puji dan syukur kepada Tuhan Yang Maha Esa atas segala rahmat dan karunia-Nya yang telah diberikan kepada Penulis sehingga dapat menyelesaikan Tugas akhir ini dengan judul “**Sistem Pakar Mendeteksi Kerusakan Mikro Kontroler Atmega328 Menggunakan Metode Case Base Reasoning (CBR)**”.

Penyusunan Tugas Akhir ini sebagai syarat untuk memperoleh Kelulusan pada Universitas Pembangunan Panca Budi Medan.

Tugas akhir ini dapat diselesaikan dengan baik dan tidak lepas dari bantuan dan bimbingan dari banyak pihak baik secara langsung maupun tidak langsung. Dalam kesempatan ini, penulis mengucapkan terima kasih kepada pihak yang telah membantu dalam pelaksanaan dan penyusunan Tugas Akhir ini, khususnya kepada :

1. Bapak dan Ibu sekeluarga yang selalu mendukung, mendoakan dan mendidik dengan penuh kasih.
2. Bapak Dr. H. Muhammad Isa Indrawan, S.E, M.M selaku Rektor I di Universitas Pembangunan Panca Budi
3. Bapak Hamdani S.T, M.T selaku Dekan Fakultas Sistem Komputer Universitas Pembangunan Panca Budi.
4. Bapak Eko Hariyanto, S.Kom., M.Kom selaku Ketua Program Studi Sistem Komputer Universitas Pembangunan Panca Budi.
5. Bapak Solly Ariza Lubis, S.T., M.Eng selaku Dosen pembimbing I yang telah memberikan pengalaman, arahan dan pengetahuan selama penyusunan Tugas akhir ini.
6. Bapak Jodi Hendrawan, S.kom., M.Kom selaku Dosen pembimbing II yang telah memberikan pengalaman, arahan dan pengetahuan selama penyusunan Tugas akhir ini.

Penulis juga menyadari bahwa penyusunan Tugas Akhir ini belum sempurna baik dalam penulisan maupun isi disebabkan keterbatasan kemampuan penulis. Oleh karena itu, penulis mengharapkan kritik dan saran yang sifatnya membangun dari pembaca untuk penyempurnaan isi Tugas Akhir ini.

Medan, 23 Mei 2021
Penulis

M. Wahyudi Hutabarat
1724370837

BAB I

PENDAHULUAN

1.1 Latar Belakang

Secara umum sistem pakar (*expert system*) adalah salah satu bidang ilmu komputer yang dapat membuat sistem berperilaku cerdas seperti manusia. Sistem ini berusaha mengadopsi pengetahuan manusia ke komputer, agar komputer dapat menyelesaikan masalah seperti yang biasa dilakukan oleh para ahli. Sistem pakar akan memberi daftar gejala-gejala sampai bisa mengidentifikasi suatu objek berdasarkan jawaban yang diterima. Dengan adanya sistem pakar ini diharapkan nantinya bisa membantu para masyarakat mendapatkan informasi seputar penyakit menular karena mata pada ATmega328 beserta diagnosanya.

ATmega328 adalah mikrokontroler keluaran dari atmel yang mempunyai arsitektur RISC (*Reduce Instruction Set Computer*) yang dimana setiap proses eksekusi data lebih cepat dari pada arsitektur CISC (*Completed Instruction Set Computer*). Dalam hal ini, pada mikrokontroler ATmega328 sering sekali terjadi kerusakan yang sangat berpengaruh pada struktur dari sistem. Sering terjadi ketika sebuah alat sudah dirakit dan rekatkan oleh solder, ATmega328 menjadi *error* sehingga struktur ataupun rangkaian yang sudah direkatkan harus dibongkar kembali. Berdasarkan dari masalah tersebut penulis ingin membuat sistem pakar yang dapat mendiagnosa kerusakan pada ATmega328 menggunakan metode *case base reasoning (CBR)*.

Metode *case base reasoning* adalah salah satu metode untuk membangun sistem pakar dengan pengambilan keputusan dari kasus yang baru dengan berdasarkan solusi dari kasus – kasus sebelumnya. Konsep dari metode *case based reasoning* ditemukan dari ide untuk menggunakan pengalaman – pengalaman yang terdokumentasi untuk menyelesaikan masalah yang baru. *Case-Based Reasoning* (CBR) merupakan sistem penalaran komputer yang menggunakan pengetahuan lama untuk mengatasi masalah baru. CBR memberikan solusi terhadap kasus baru dengan melihat kasus lama yang paling mendekati kasus baru. Hal ini akan sangat bermanfaat karena dapat menghilangkan kebutuhan untuk mengekstrak model seperti yang dibutuhkan oleh sistem berbasis aturan (Rismawan, 2012).

Berdasarkan latar belakang diatas, maka diperlukan adanya sistem pakar yang dapat mendiagnosa kerusakan pada ATmega328 menggunakan metode *case base reasoning* (CBR). Sistem pakar ini akan memberikan bantuan diagnosa awal dan saran pengobatan atas penyakit yang diderita ATmega328 berdasarkan gejala-gejala yang akan dimasukkan *user* ke dalam sistem.

Berdasarkan uraian tersebut, diangkatlah sebuah topik penelitian dengan judul **“Sistem Pakar Mendeteksi Kerusakan Mikrokontroler Atmega328 Menggunakan Metode *Case Base Reasoning* (CBR)”**.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan diatas, mendapatkan rumusan masalah dalam penelitian ini adalah :

1. Bagaimana mengenali kerusakan pada ATMega328, sehingga dapat dilakukan pencegahan lebih cepat?
2. Bagaimana menerapkan metode *Case Base Reasoning* untuk mendiagnosa kerusakan pada ATMega328?
3. Bagaimana membangun aplikasi sistem pakar dalam mendiagnosa kerusakan pada ATMega328?

1.3 Batasan Masalah

Agar penelitian ini sesuai dengan permasalahan yang di teliti, maka diperlukan batasan masalah yaitu :

1. Penerapan sistem pakar ini digunakan sebagai media konsultasi dan hanya membahas tentang mendiagnosa kerusakan pada ATMega328.
2. Penerapan sistem pakar dengan metode *Case Base Reasoning* ini hanya membahas kerusakan pada ATMega328.
3. Sistem yang dibuat menggunakan perangkat lunak berbasis Pemrograman Dekstop.

1.4 Tujuan Penelitian

Berdasarkan rumusan masalah yang dibuat, maka tujuan dari skripsi ini adalah :

1. Untuk mengenali kerusakan pada ATMega328, sehingga dapat dilakukan pencegahan kerusakan tersebut.
2. Untuk menerapkan metode CBR (*Case Base Reasoning*) sistem pakar mendiagnosa kerusakan pada ATMega328.

3. Untuk membangun aplikasi sistem pakar dalam mendiagnosa suatu kerusakan pada ATMEga328.

1.5 Manfaat Penelitian

Adapun manfaat yang di dapat dalam penelitian ini adalah sebagai berikut:

1. Dapat mengenali kerusakan pada ATMEga328, sehingga dapat dilakukan pencegahan kerusakan tersebut.
2. Dapat menerapkan metode CBR (*Case Base Reasoning*) sistem pakar mendiagnosa kerusakan pada ATMEga328.
3. Dapat membangun aplikasi sistem pakar dalam mendiagnosa suatu kerusakan pada ATMEga328.

BAB II

LANDASAN TEORI

2.1 Sistem Pakar

Menurut Turban, (2011) Sistem pakar adalah sistem yang menggunakan pengetahuan manusia dimana pengetahuan tersebut dimasukkan ke dalam sebuah komputer dan kemudian digunakan untuk menyelesaikan masalah yang membutuhkan kepakaran atau keahlian manusia.

Sistem Pakar adalah Prosedur umum ketika menerapkan CBR, diperkenalkan oleh Aamodt dan Plaza (1994). Basis pengetahuan pada *Case Base Reasoning* adalah fakta-fakta berupa kasus-kasus sebelumnya yang pernah ada dan serangkaian alur untuk memeriksa, menghitung serta menyimpulkan suatu solusi dari permasalahan atau kasus yang baru Tahapan pada *Case Base Reasoning* ada 4 yaitu: ***Retrieve, Reuse, Revise*** dan ***Retain***.

2.1.1 Manfaat Sistem Pakar

Sistem Pakar menjadi sangat populer karena sangat banyak kemampuan dan manfaat yang diberikannya, di antaranya adalah

1. Membuat seorang yang awam bekerja seperti layaknya seorang pakar.
2. Mampu menangkap pengetahuan dan kepakaran seseorang.
3. Dapat beroperasi di lingkungan yang berbahaya.
4. Memudahkan akses pengetahuan seorang pakar.

5. Meningkatkan kualitas, dengan memberi nasehat yang konsisten dan mengurangi kesalahan.
6. Meningkatkan produktivitas, karena Sistem Pakar dapat bekerja lebih cepat dari pada manusia.
7. Bisa digunakan sebagai media pelengkap dalam pelatihan. Pengguna pemula yang bekerja dengan Sistem Pakar akan menjadi lebih berpengalaman karena adanya fasilitas penjelas yang berfungsi sebagai guru.
8. Meningkatkan kapabilitas sistem komputer. Integrasi Sistem Pakar dengan sistem komputer lain membuat sistem efektif dan mencakup lebih banyak aplikasi.
9. Mampu bekerja dengan informasi yang tidak pasti. Berbeda dengan sistem komputer konvensional, Sistem pakar dapat bekerja dengan informasi yang tidak lengkap. Pengguna dapat merespon dengan: “tidak tahu” atau “tidak yakin” pada satu atau lebih pertanyaan selama konsultasi dan sistem pakar tetap akan memberikan jawabannya.
10. Meningkatkan kemampuan untuk menyelesaikan masalah karena Sistem Pakar.
11. Andal. Sistem Pakar tidak pernah menjadi bosan dan kelelahan atau sakit.

2.1.2 Kelebihan Sistem Pakar dan Kelemahan Sistem Pakar

1. Kelebihan Sistem Pakar
 - a. Menghimpun data jumlah yang besar.
 - b. Menyimpan pengetahuan dan keahlian para pakar.
 - c. Meningkatkan kualitas.

- d. Memiliki reliabilitas .
- e. Dapat beroperasi di lingkungan yang berbahaya.
- f. Mengerjakan perhitungan secara cepat dan tepat serta mencari kembali data yang tersimpan dengan kecepatan tinggi.
- g. Meningkatkan kapabilitas dalam penyelesaian masalah.
- h. Memudahkan akses pengetahuan seorang pakar.
- i. Andal.
- j. Bisa digunakan sebagai media pelengkap dalam pelatihan. Pengguna pemula yang bekerja dengan sistem pakar akan menjadi lebih berpengalaman karena adanya fasilitas penjelas yang berfungsi sebagai guru.
- k. Membuat seorang yang awam bekerja seperti layaknya seorang pakar.

2. Kelemahan Sistem Pakar

- a. Pengetahuan tidak selalu bisa didapat dengan mudah. Karena pendekatan yang dibuat oleh satu pakar dengan pakar lainnya berbeda.
- b. Untuk membuat suatu sistem yang berkualitas sangat sulit dan memerlukan biaya yang tinggi.
- c. Sistem pakar tidak 100% benar, perlu diuji ulang sebelum digunakan. Dalam hal ini peranan manusia merupakan faktor dominan.
- d. Pengembangan perangkat lunak sistem pakar lebih sulit dibandingkan perangkat lunak konvensional.
- e. Susah dikembangkan.

2.1.3 Ciri-Ciri Sistem Pakar

Ciri-ciri dari sistem pakar adalah sebagai berikut

1. Mudah dimodifikasi.
2. Output bersifat anjuran.
3. Terbatas pada domain keahlian tertentu.
4. Dapat memberikan penalaran untuk data-data yang tidak lengkap atau tidak pasti.
5. Dapat menjelaskan alasan-alasan dengan cara yang dapat dipahami.
6. Bekerja berdasarkan pada kaidah atau *rule* tertentu.
7. Basis pengetahuan dan mekanisme inferensi terpisah.
8. Sistem dapat mengaktifkan kaidah secara searah yang sesuai.

2.1.4 Konsep Dasar Sistem Pakar

Konsep dasar sistem pakar meliputi enam hal berikut ini

1. Kepakaran (*Expertise*)

Kepakaran merupakan suatu pengetahuan yang diperoleh dari pelatihan, membaca, dan pengalaman. Kepakaran inilah yang memungkinkan para ahli dapat mengambil keputusan lebih cepat dan lebih baik. Kepakaran itu sendiri meliputi pengetahuan tentang:

- a. Fakta-fakta tentang bidang permasalahan tertentu.
- b. Teori-teori tentang bidang permasalahan tertentu.
- c. Aturan-aturan dan prosedur-prosedur menurut bidang permasalahan umumnya.
- d. Aturan *heuristic* yang harus dikerjakan dalam suatu situasi tertentu.

- e. Strategi global untuk memecahkan permasalahan.
- f. Pengetahuan tentang pengetahuan (*meta knowledge*).

2. Pakar (*Expert*)

Pakar adalah seseorang yang mempunyai pengetahuan, pengalaman, dan metode khusus, serta mampu menerapkannya untuk memecahkan masalah atau memberi nasehat. Seorang pakar harus mampu menjelaskan dan mempelajari hal-hal baru yang berkaitan dengan topik permasalahan, jika perlu harus mampu menyusun kembali pengetahuan-pengetahuan yang didapatkan, dan dapat memecahkan aturan-aturan serta menentukan relevansi kepakarannya. Jadi, seseorang pakar harus mampu melakukan kegiatan-kegiatan berikut:

- a. Mengenali dan memformulasikan permasalahan.
- b. Memecahkan permasalahan secara cepat dan tepat.
- c. Menerangkan pemecahannya.
- d. Belajar dari pengalaman.
- e. Merestrukturisasi pengetahuan.
- f. Memecahkan aturan-aturan.
- g. Menentukan relevansi.

3. Pemindahan Kepakaran (*Transferring Expertise*)

Tujuan dari Sistem Pakar adalah memindahkan kepakaran dari seorang pakar ke dalam komputer, kemudian ditransfer kepada orang lain yang bukan pakar. Proses ini melibatkan empat kegiatan, yaitu:

- a. Akuisisi pengetahuan (dari pakar atau sumber lain).
- b. Representasi pengetahuan (pada komputer).

- c. Inferensi Pengetahuan.
- d. Perpindahan pengetahuan ke pengguna.

4. Inferensi (*Inferencing*)

Inferensi adalah sebuah prosedur (program) yang mempunyai kemampuan dalam melakukan penalaran. Inferensi ditampilkan pada suatu komponen yang disebut mesin inferensi yang mencakup prosedur-prosedur mengenai pemecahan masalah. Semua pengetahuan yang dimiliki oleh seorang pakar disimpan pada basis pengetahuan oleh sistem pakar. Tugas mesin inferensi adalah mengambil kesimpulan berdasarkan basis pengetahuan yang dimilikinya.

5. Aturan-Aturan (*Rule*)

Kebanyakan *software* sistem pakar komersial adalah sistem yang berbasis *rule* (*rule-based systems*), yaitu pengetahuan disimpan terutama dalam bentuk *rule*, sebagai prosedur-prosedur pemecahan masalah.

6. Kemampuan Menjelaskan (*Explanation Capability*)

Fasilitas lain dari Sistem Pakar adalah kemampuannya untuk menjelaskan saran atau rekomendasi yang diberikannya. Penjelasan dilakukan dalam subsistem yang disebut subsistem penjelasan (*explanation*). Bagian dari sistem ini memungkinkan *system* untuk memeriksa penalaran yang dibuatnya sendiri dan menjelaskan operasi-operasinya.

Karakteristik dan kemampuan yang dimiliki oleh sistem pakar berbeda dengan sistem konvensional. Perbedaan ini ditunjukkan pada tabel 2.1.

Tabel 2.1 Perbedaan antara Sistem Konvensional dengan Sistem Pakar

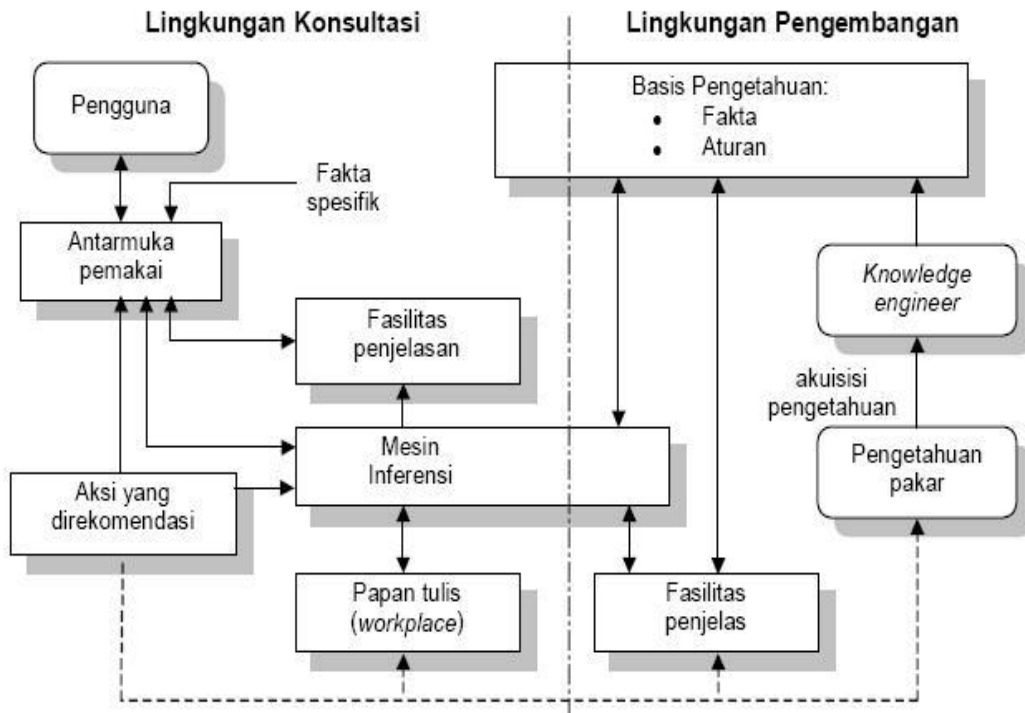
SISTEM KONVENSIONAL	SISTEM PAKAR
Informasi dan pemrosesannya biasanya jadi satu dengan program	Basis pengetahuan merupakan bagian terpisah dari mekanisme inferensi
Program tidak pernah salah (kecuali programnya yang salah)	Program bisa saja melakukan kesalahan
Biasanya tidak bisa menjelaskan mengapa suatu <i>input</i> data itu dibutuhkan atau bagaimana <i>output</i> itu diperoleh	Penjelasan adalah bagian terpenting dari sistem pakar
Pengubahan program cukup sulit dan merepotkan	Pengaturan pada aturan/kaidah dapat dilakukan dengan mudah
Sistem hanya akan bekerja jika sistem tersebut sudah lengkap	Sistem dapat bekerja hanya dengan beberapa aturan
Eksekusi dilakukan langkah demi langkah secara algoritmik	Eksekusi dilakukan pada keseluruhan basis pengetahuan secara heuristik dan logis
Menggunakan data	Menggunakan pengetahuan
Tujuan utamanya adalah efisiensi	Tujuan utamanya adalah efektivitas
Manipulasi efektif dari basis data yang besar	Manipulasi efektif dari basis pengetahuan yang besar
Menangkap, menambah, dan mendistribusikan akses ke data numerik atau informasi	Menangkap, menambah, dan mendistribusikan ke pertimbangan dan pengetahuan

Sumber : *T.Sutojo dkk, (2010:165-16)*

2.1.5 Struktur Sistem Pakar

Ada dua bagian penting dari Sistem pakar, yaitu lingkungan pengembangan (*development environment*) dan lingkungan konsultasi (*consultation environment*). Lingkungan pengembangan digunakan oleh pembuat sistem pakar

untuk membangun komponen-komponennya dan memperkenalkan pengetahuan ke dalam *knowledge base* (basis pengetahuan)



Gambar 2.1 Komponen-Komponen Sistem Pakar

Sumber : T. Sutojo dkk, (2011:167)

Keterangan:

1. Akuisisi Pengetahuan

Sub sistem ini digunakan untuk memasukkan pengetahuan dari seorang pakar dengan cara merekayasa pengetahuan agar bisa diproses oleh komputer dan meletakkannya ke dalam basis pengetahuan dengan format tertentu dalam bentuk representasi pengetahuan. Sumber-sumber pengetahuan bisa diperoleh dari pakar, buku, dokumen multimedia, basis data, laporan riset khusus, dan informasi yang terdapat di web.

2. Basis Pengetahuan (*Knowledge Base*)

Basis pengetahuan mengandung pengetahuan yang diperlukan untuk memahami, memformulasikan dan menyelesaikan masalah. Basis pengetahuan terdiri dari dua elemen dasar, yaitu:

- a. Fakta, misalnya situasi, kondisi, atau permasalahan yang ada.
- b. *Rule* (Aturan), untuk mengarahkan penggunaan pengetahuan dalam memecahkan masalah.

3. Mesin Inferensi (*Inference Engine*)

Mesin inferensi adalah sebuah program yang berfungsi untuk memandu proses penalaran terhadap suatu kondisi berdasarkan pada basis pengetahuan yang ada, memanipulasi dan mengarahkan kaidah, model, dan fakta yang disimpan dalam basis pengetahuan untuk mencapai solusi atau kesimpulan. Dalam prosesnya, mesin inferensi menggunakan strategi pengendalian, yaitu strategi yang berfungsi sebagai panduan arah dalam melakukan proses penalaran.

4. Daerah Kerja (*Blackboard*)

Untuk merekam hasil sementara yang akan dijadikan sebagai keputusan dan untuk menjelaskan sebuah masalah yang sedang terjadi. Sistem pakar membutuhkan ***Blackboard***, yaitu area pada memori yang berfungsi sebagai basis data. Tiga tipe keputusan yang dapat direkam pada *blackboard*, yaitu:

- a. Rencana: bagaimana menghadapi masalah
- b. Agenda: aksi-aksi potensial yang sedang menunggu untuk dieksekusi
- c. Solusi: calon aksi yang akan dibangkitkan.

5. Antarmuka Pengguna (*User Interface*)

Digunakan sebagai media komunikasi antara pengguna dan sistem pakar. Komunikasi disajikan dalam bahasa alami (*natural language*) dan dilengkapi dengan grafik, menu, dan formulir elektronik. Pada bagian ini akan terjadi dialog antara sistem pakar dengan pengguna.

6. Subsistem Penjelasan (*Explanation Subsystem/Justifier*)

Berfungsi memberi penjelasan kepada pengguna, bagaimana suatu kesimpulan dapat diambil. Kemampuan seperti ini sangat penting bagi pengguna untuk mengetahui proses pemindahan keahlian pakar maupun dalam pemecahan masalah.

7. Sistem Perbaiki Pengetahuan (*Knowledge Refining System*)

Kemampuan memperbaiki pengetahuan (*knowledge refining system*) dari seorang pakar diperlukan untuk menganalisis pengetahuan, belajar dari kesalahan masa lalu, kemudian memperbaiki pengetahuannya, sehingga dapat dipakai pada masa mendatang. Kemampuan evaluasi diri seperti itu diperlukan oleh program agar dapat menganalisis untuk mengambil kesimpulan. Dengan cara ini basis pengetahuan yang lebih baik dan penalaran yang lebih efektif akan dihasilkan.

8. Pengguna (*User*)

Pada umumnya pengguna sistem pakar bukanlah seorang pakar (*non-expert*) yang membutuhkan solusi, saran, atau pelatihan (*training*) dari berbagai permasalahan yang ada.

2.2 Teori Case Base Reasoning

Basis pengetahuan pada *Case Base Reasoning* adalah fakta-fakta berupa kasus-kasus sebelumnya yang pernah ada dan serangkaian alur untuk memeriksa,

menghitung serta menyimpulkan suatu solusi dari permasalahan atau kasus yang baru tahapan pada *Case Base Reasoning* ada 4 yaitu: ***Retrieve, Reuse, Revise***, dan ***retain***.

1. *Retrieve*

Melakukan pencarian tingkat kemiripan kasus inputan gejala yang diberikan pasien dengan kasus yang ada pada *knowledge base* untuk mencari kasus yang memiliki persamaan dengan basis kasus yang akan diproses.

2. *Reuse*

Pada tahap ini dilakukan perhitungan nilai similaritas (kemiripan) setiap kasus tertinggi yang sebelumnya telah dilakukan pencocokan dari kasus baru ke dalam kasus yang lama, sehingga dari hasil perhitungan nilai persamaan *similarity* didapatkan usulan solusi. Nilai *similarity* setiap kasus diperoleh dari: (Jumlah nilai gejala *similarity* terpilih x bobot)/ total nilai bobot gejala dalam data kasus lama).

3. *Revise*

Tahapan *revise* merupakan tahapan solusi kasus, misal saran obat dari hasil diagnosa. Pada proses validasi ini sistem akan melakukan *revise* melalui pertimbangan yang dimiliki oleh paramedis. Sistem akan menghasilkan solusi yang telah dikembangkan dan disimpan ke dalam sistem untuk dijadikan pengetahuan baru.

4. *Retain*

Pada proses *retain* solusi yang sudah diperbaiki akan disimpan sebagai data rekam medis pasien. Jika pakar menyatakan kasus baru tersebut sebagai kasus valid maka kasus baru tersebut dapat diubah ke dalam basis kasus.

Untuk menghitung kemiripan kasus, digunakan rumus berikut:

$$\text{Similarity}(p,q) = \frac{S_1 \times W_1 + S_2 \times W_2 + \dots + S_n \times W_n}{W_1 + W_2 + \dots + W_n} \quad (1)$$

Penelusuran pada aplikasi ini menggunakan teknik *Similarity(problem, case)* pada algoritma *k-nearest neighbor* sebagai berikut : Bobot parameter (w) : Gejala Penting = 1, Gejala Biasa = 0,5 $\text{Similarity}(T,S) = s_1 * w_1 + s_2 * w_2 + \dots + s_n * w_n / w_1 + w_2 + \dots + w_n$

Keterangan :

p : kasus baru

q : kasus yang ada dalam penyimpanan (*case*)

i : *weight* (bobot yang diberikan pada atribut ke-i)

s : *similarity* (nilai kemiripan)

w = *weight* (bobot yang diberikan)

f = Fungsi *similarity* atribut i antara kasus p dan kasus q

n = Jumlah atribut dalam setiap kasus

2.3 ATmega328

ATmega328 adalah mikrokontroler keluaran dari atmel yang mempunyai arsitektur RISC (*Reduce Instruction Set Computer*) yang dimana setiap proses eksekusi data lebih cepat dari pada arsitektur CISC (*Completed Instruction Set Computer*).

ATmega328 memiliki beberapa fitur antara lain :

1. 130 macam instruksi yang hampir semuanya dieksekusi dalam satu siklus *clock*.
2. 32 x 8-bit register serba guna.
3. Kecepatan mencapai 16 MIPS dengan clock 16 MHz.
4. 32 KB *Flash memory* dan pada arduino memiliki *bootloader* yang menggunakan 2 KB dari *flash memory* sebagai *bootloader*.
5. Memiliki EEPROM (*Electrically Erasable Programmable Read Only Memory*) sebesar 1KB sebagai tempat penyimpanan data semi permanent karena EEPROM tetap dapat menyimpan data meskipun catu daya dimatikan.
6. Memiliki SRAM (*Static Random Access Memory*) sebesar 2KB.
7. Memiliki pin I/O digital sebanyak 14 pin 6 diantaranya PWM (*Pulse Width Modulation*) output.
8. *Master / Slave SPI Serial interface*.

Mikrokontroler ATmega 328 memiliki arsitektur Harvard, yaitu memisahkan memori untuk kode program dan memori untuk data sehingga dapat

memaksimalkan kerja dan parallelism. Instruksi – instruksi dalam memori program dieksekusi dalam satu alur tunggal, dimana pada saat satu instruksi dikerjakan instruksi berikutnya sudah diambil dari memori program.

Konsep inilah yang memungkinkan instruksi – instruksi dapat dieksekusi dalam setiap satu siklus *clock*. 32 x 8-bit *register* serba guna digunakan untuk mendukung operasi pada ALU (*Arithmetic Logic unit*) yang dapat dilakukan dalam satu siklus. 6 dari register serbaguna ini dapat digunakan sebagai 3 buah register *pointer* 16-bit pada mode pengalamatan tidak langsung untuk mengambil data pada ruang memori data Ketiga register *pointer* 16-bit ini disebut dengan register X (gabungan R26 dan R27), register Y (gabungan R28 dan R29), dan register Z (gabungan R30 dan R31). Hampir semua instruksi AVR memiliki *format* 16-bit. Setiap alamat memori program terdiri dari instruksi 16-bit atau 32-bit. Selain register serba guna di atas, terdapat register lain yang terpetakan dengan teknik *memory mapped I/O* selebar 64 byte. Beberapa register ini digunakan untuk fungsi khusus antara lain sebagai register *control Timer/Counter, Interupsi, ADC, USART, SPI, EEPROM*, dan fungsi I/O lainnya. Register – register ini menempati memori pada alamat 0x20h – 0x5Fh.

2.4 Pemodelan

Menurut Rosa,(2016:135-136) Pemodelan adalah gambaran dari realita yang simpel dan dituangkan dalam bentuk pemetaan dengan aturan tertentu. Pemodelan dapat menggunakan bentuk yang sama dengan realitas misalnya jika seorang arsitek ingin sebuah gedung yang akan dibangun yang akan maka dia akan

memodelkannya dengan membuat sebuah maket (tiruan) arsitektur gedung yang akan dibangun dimana maket itu akan dibuat semirip mungkin dengan desain gedung yang akan dibangun agar arsitektur gedung yang diinginkan dapat terlihat.


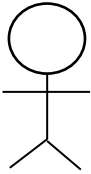
2.5 UML (*Unified Modeling Language*)


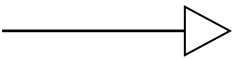
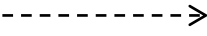
Menurut Rosa dan Shalahuddin, (2016:133-137) UML (*Unified Modeling Language*) adalah salah satu standar bahasa yang banyak digunakan didunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek. Pemodelan berorientasi objek biasanya dituangkan dalam dokumentasi perangkat lunak dengan menggunakan sebuah alat pemodelan berorientasi objek yang disebut UML, yang didalamnya terdapat diagram-diagram yang membantu proses pendekatan sistem dan pada tahap ini biasanya dapat dikenali tentang kendala dan permasalahan yang terjadi pada saat pembangunan sistem berorientasi objek. UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak.

2.5.1 *Use Case Diagram*

Menurut Rosa, (2016:155) *Use Case* atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use Case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu

Tabel 2.2 Simbol-simbol *Use Case Diagram*

Simbol	Deskripsi
<p><i>Use Case</i></p> 	<p>Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja diawal frase nama <i>use case</i>.</p>
<p>Aktor/<i>actor</i></p>  <p>nama actor</p>	<p>Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun symbol dari aktor adalah gambar orang; biasanya dinyatakan menggunakan kata benda diawal frase nama <i>actor</i></p>

<p>Asosiasi/<i>association</i></p> 	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu; mirip denga prinsip <i>inheritance</i> pada pemrograman berorientasi objek; biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan</p>
<p>generalisasi/<i>generalization</i></p> 	<p>Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya</p>
<p><<<i>include</i>>></p> 	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syaratdijalankan <i>use case</i> ini</p>


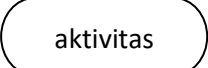
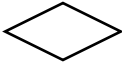


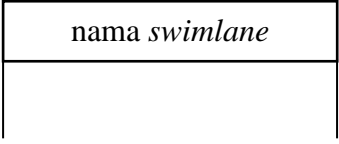
Sumber : Rosa A.S., dan M. Shalahuddin, (2016 : 156-157).

2.5.2 Activity Diagram

Diagram aktivitas atau *ativity diagram* menggambarkan aliran kerja (*workflow*) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada perangkat

lunak. *Activity diagram* menggambarkan berbagai aliran aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alur berawal, keputusan yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi

Tabel 2.3 Simbol-simbol *Activity Diagram*

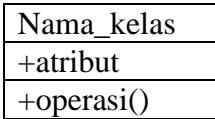
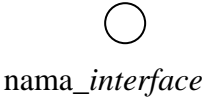


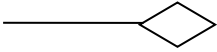
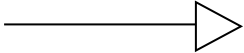

Simbol	Deskripsi
Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
Percabangan / <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
Penggabungan / <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
<i>Swimlane</i> 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

Sumber : Rosa A.S., dan M. Shalahuddin (2016 : 162-163).

2.5.3 Class Diagram

Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. (Rosa A.S., M. Shalahuddin, 2014)

Tabel 2.4 Simbol-simbol *Class Diagram*

Simbol	Deskripsi
Kelas 	Kelas pada struktur sistem
Antarmuka / <i>interface</i> 	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.
Asosiasi / <i>association</i> 	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
Asosiasi berarah / <i>directed association</i> 	Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
Agregasi / <i>aggregation</i> 	Relasi antarkelas dengan makna semua-bagian (<i>whole-part</i>)
Generalisasi 	Relasi antarkelas dengan makna generalisasi-spesialisasi (umum khusus)
Kebergantungan / <i>dependency</i> 	Relasi antarkelas dengan makna kebergantungan antar kelas

Sumber: Rosa A.S., dan M. Shalahuddin (2016 : 146-147).

2.6 Flowchart

Menurut Jogiyanto (Santoso, 2018 : 7) “*Flowchart* adalah salah satu bagan yang menjelaskan dengan rinci langkah – langkah dari proses suatu program”.

”*Flowchart* adalah bagan-bagan yang mempunyai arus yang menggambarkan langkah-langkah penyelesaian suatu masalah. *Flowchart* merupakan cara penyajian dari suatu algoritma”. (Ladjamudin, 2005:263).

Ada dua macam *flowchart* yang menggambarkan proses dengan komputer, yaitu:

1. Sistem *Flowchart*

Bagan yang memperlihatkan urutan proses dalam system dengan menunjukkan alat media *input*, *output* serta jenis media penyimpanan dalam proses pengolahan data.

2. Program *Flowchart*

Bagan yang memperlihatkan urutan instruksi yang digambarkan dengan simbol tertentu untuk memecahkan masalah dalam suatu program.

Flowchart membantu seorang analis dan *programmer* dalam memecahkan suatu masalah serta dapat membantu dalam menganalisis alternatif – alternatif dalam pengoperasiannya. Berikut ini simbol – simbol standart yang digunakan.

Fungsi *Flowchart* untuk menggambarkan sebuah proses agar mempermudah pemahaman dan mudah dilihat berdasarkan urutan langkahnya berdasarkan proses yang satu ke proses yang lainnya.

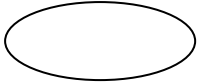
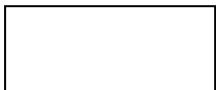

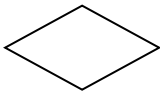

Flowchart disusun dengan simbol. Simbol ini dipakai sebagai alat bantu menggambarkan proses di dalam program. Simbol-simbol yang digunakan dapat dibagi menjadi 3 (tiga) kelompok, yakni sebagai berikut:

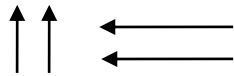
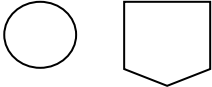
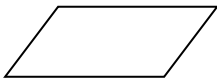
- a. *Flow Direction Symbols* (Simbol Penghubung/ Alur)
- b. *Processing Symbols* (Simbol Proses)
- c. *Input-Output Symbols* (Simbol *Input-Output*)

2.6.1 *Flow Direction Symbols*

Simbol yang digunakan untuk menghubungkan antara simbol yang satu dengan simbol yang lain. Simbol ini disebut juga *connecting line*. Simbol-simbol tersebut adalah sebagai berikut

Tabel 2.5 Simbol-simbol *Flow Direction Symbols*

Simbol	Keterangan
Terminal 	Untuk menggambarkan awal dan akhir proses aliran dokumen
<i>Processing</i> 	Dipakai untuk pengolahan aritmatika dan pemindahan data
<i>Preparation</i> 	Dipakai untuk memberikan nilai awal dari suatu <i>variable</i> atau <i>counter</i>
<i>Decision</i> 	Dipakai untuk mewakili operasi perbandingan logika
<i>Predefined process</i> 	Dipakai untuk proses yang detailnya dijelaskan secara terpisah, misalnya dalam bentuk <i>subroutine</i>






Garis alir 	Dipakai untuk menunjukkan aliran dari program
Penghubung 	Menunjukkan penghubung di halaman yang masih sama atau di halaman lainnya
<i>Input / Output</i> 	Mewakili data <i>input</i> atau <i>output</i>

Sumber : *Santoso, 2018 : 7*

2.6.2 Processing Symbols

Simbol yang menunjukkan jenis operasi pengolahan dalam suatu proses / prosedur. Simbol-simbol tersebut adalah sebagai berikut

Tabel 2.6 Processing Symbols






	Process	Simbol yang menunjukkan pengolahan yang dilakukan Komputer
	Decision	Simbol untuk kondisi yang akan menghasilkan beberapa kemungkinan jawaban / aksi
	Predefined Process	Simbol untuk mempersiapkan penyimpanan yang akan digunakan sebagai tempat pengolahan didalam storage
	Terminal	Simbol untuk permulaan atau akhir dari suatu program
	Manual Input	Simbol untuk pemasukan data secara manual on-line keyboard

Sumber : *Al-Bahra Bin Ladjamudin 2005:267*

2.6.3 Input-Output Symbols

Simbol yang menunjukkan jenis peralatan yang digunakan sebagai media *input* atau *output*. Simbol-simbol tersebut sebagai berikut

Tabel 2.7 *Input-Output Symbols*

	Process	Simbol yang menunjukkan pengolahan yang dilakukan Komputer
	Decision	Simbol untuk kondisi yang akan menghasilkan beberapa kemungkinan jawaban / aksi
	Predefined Process	Simbol untuk mempersiapkan penyimpanan yang akan digunakan sebagai tempat pengolahan didalam storage
	Terminal	Simbol untuk permulaan atau akhir dari suatu program
	Manual Input	Simbol untuk pemasukan data secara manual on-line keyboard

Sumber : Al-Bahra Bin Ladjamudin 2005:267

2.7 Aplikasi Pendukung

Adapun *software* yang digunakan dalam skripsi ini yaitu *Microsoft Visual Basic 2010* dan *Microsoft Access 2010*.

2.7.1 Visual Basic NET

Visual Basic NET adalah *visual basic* yang direkayasa kembali untuk digunakan pada *platform .NET* sehingga aplikasi yang dibuat menggunakan *Visual Basic .NET* dapat berjalan pada sistem komputer apapun, dan dapat mengambil data dari *server* dengan tipe apapun asalkan terinstal *.NET Framework*. Adapun beberapa kelebihan yang dimiliki VB .NET antara lain

1. Sederhana dan Mudah Dipahami

Bahasa yang digunakan VB .NET sangat sederhana sehingga lebih mudah dipahami bagi mereka yang masih awam terhadap dunia pemrograman.

2. Mendukung GUI (*Graphical User Interface*)

VB .NET bisa membuat *software* dengan antarmuka grafis yang *user friendly*.

3. Menyederhanakan *Deployment*

VB .NET mengatasi masalah *deployment* dari aplikasi berbasis *Windows* yaitu *DLL Hell* dan registrasi COM (*Component Object Model*). Selain itu tersedia *wizard* yang memudahkan dalam pembuatan *file setup*.

4. Menyederhanakan Pengembangan Perangkat Lunak

Ketika penulisan kode mengalami kesalahan maka VB .NET langsung menampilkan pesan kesalahan pada bagian *Message Windows* sehingga *programmer* dapat memperbaiki dengan cepat serta *wizard* yang lainnya yang mendukung *programmer* dalam pembuatan *software*.

5. Mendukung Penuh OOP

Memiliki fitur bahasa pemrograman berorientasi objek seperti *inheritance* (pewarisan), *encapsulation* (pembungkusan), *polymorphism* (banyak bentuk).

6. Mempermudah Pengembangan Aplikasi Berbasis Web

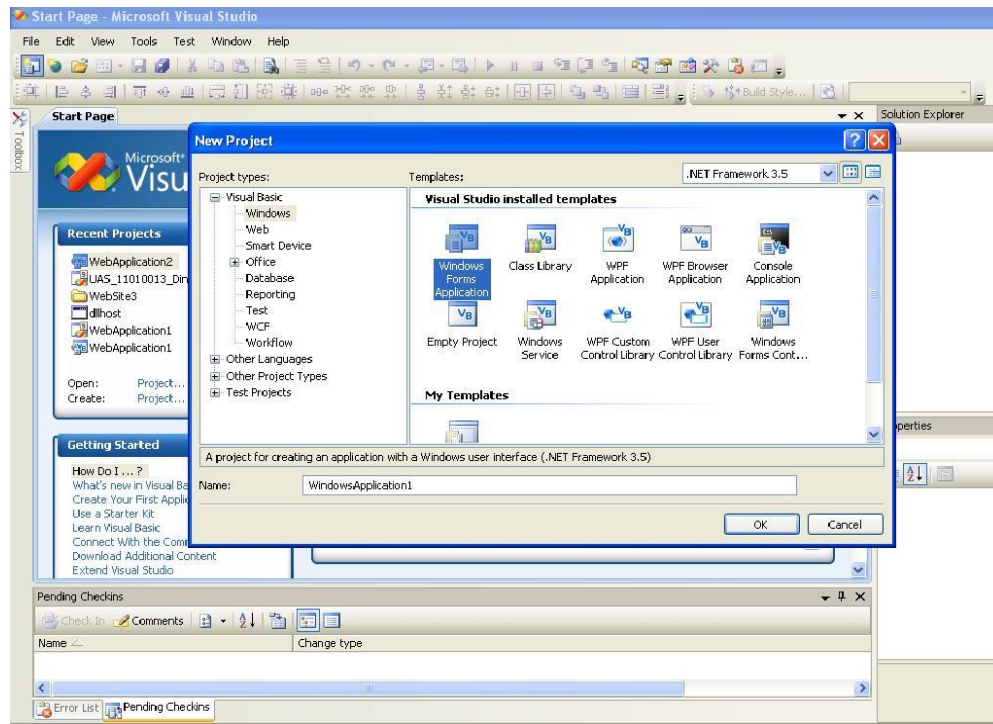
Disediakan desain *form Web*. Selain itu juga disediakan layanan Web XML sehingga memungkinkan suatu aplikasi “berkomunikasi” dengan aplikasi lainnya dari berbagai *platform* menggunakan protokol internet terbuka.

7. Migrasi ke VB .NET Dapat Dilakukan Dengan Mudah

Konversi aplikasi VB ke VB .NET dapat dijalankan dengan mudah.

8. Banyak digunakan oleh kalangan *programmer-programmer* diseluruh dunia.

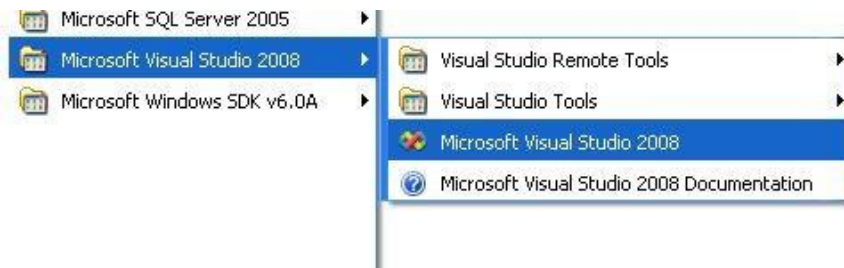
Jika memiliki suatu kendala maka keuntungannya dapat bertanya dan berkonsultasi dengan *programmer-programmer* lain yang ada diseluruh dunia melalui forum diinternet.



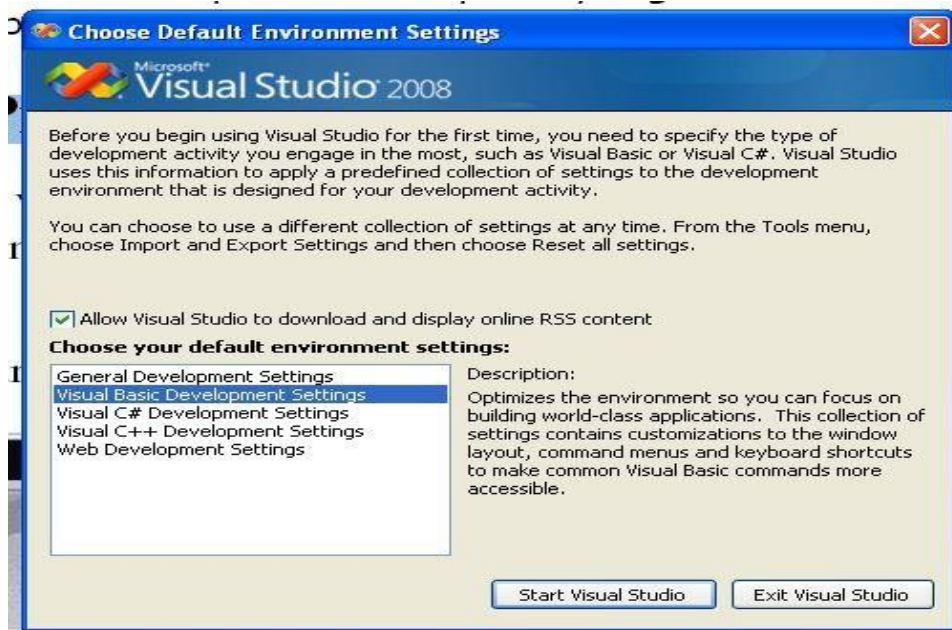
Gambar 2.2 Tampilan *Visual Basic*

Langkah awal dari *Visual Basic* adalah mengenal IDE (*Integrated Development Environment*) *Visual Basic* yang merupakan Lingkungan Pengembangan Terpadu bagi programmer dalam mengembangkan aplikasinya. Dengan menggunakan IDE programmer dapat membuat *user interface*, melakukan koding, melakukan *testing* dan *debuging* serta mengkompilasi program menjadi *executable*. Penguasaan yang baik akan IDE akansangat membantu programmer dalam mengefektifkan tugas-tugasnya sehingga dapat bekerja dengan efisien.

Salah satu cara untuk mengaktifkan IDE *Visual Basic* adalah menjalankannya dari Menu *Start*, pilih menu *all programs*, dan pilih *Microsoft Visual Studio 2008* dan klik *Microsoft Visual Basic 2008* Kemudian *Visual Basic Development Settings*

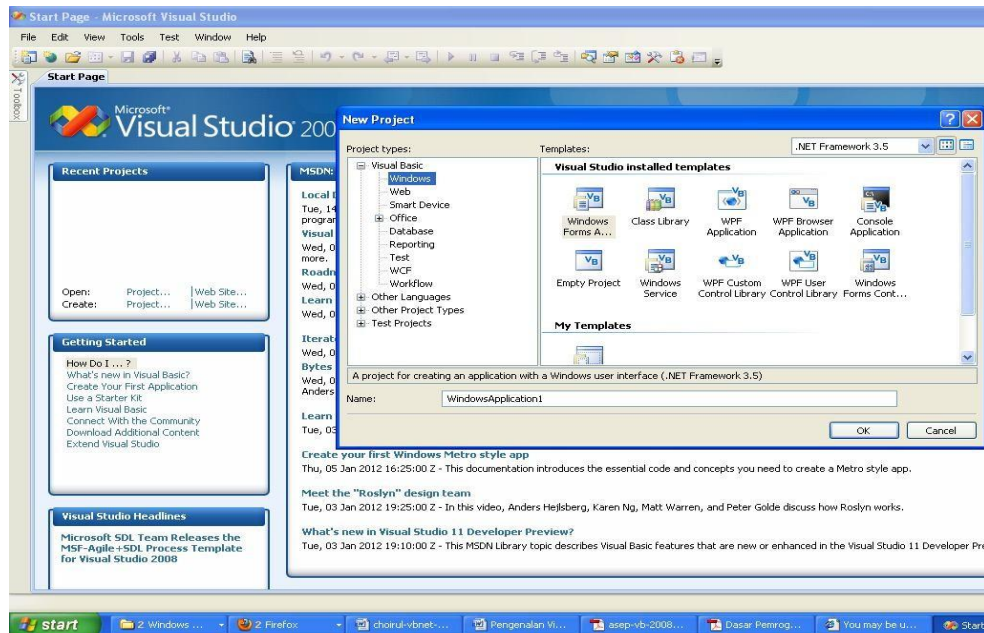


Gambar 2.3 Mengaktifkan IDE *Visual Basic2008*



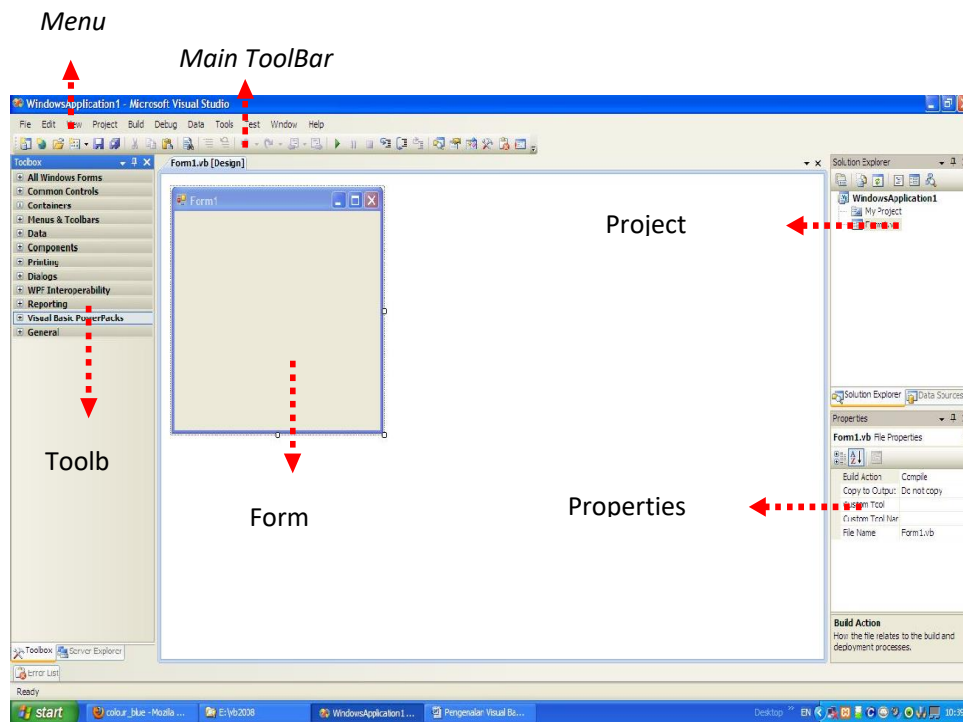
Gambar 2.4 Mengaktifkan IDE *Visual Basic 2008*

Sesaat anda aktif di IDE *Visual Basic*, pilih *File* kemudian *New Project* kemudian *Windows* lalu *Windows Form Applications*



Gambar 2.5 Dialog box *New Project* ditampilkan sesaat anda menjalankan IDE Jendela *Visual Basic 2008*

Setelah menuliskan nama *project* atau jika tidak ditulis atau diganti secara *default* nama projectnya adalah *WindowsApplication1* kemudian klik tombol OK. Berikutnya akan dihadapkan pada tampilan grafis dengan lingkungan IDE yang baru pada *Visual Basic Express Edition*, seperti gambar dibawah ini :



Gambar 2.6 Jendela *interface Visual Basic 2008*

Keterangan dari gambar diatas adalah sebagai berikut :

1. *Menu Bar*

Digunakan untuk memilih tugas-tugas tertentu seperti menyimpan *project*, membuka *project*, dan lain – lain.

2. *Main Toolbar*

Digunakan untuk melakukan tugas-tugas tertentu dengancepat.

3. *Jendela Toolbox*

Jendela ini berisi komponen-komponen yang dapat anda gunakan untuk mengembangkan *userinterface*.

4. *Jendela Form Design*

Jendela ini merupakan tempat anda untuk merancang *user interface* dari aplikasi anda.

5. *Jendela Properties Windows*

Merupakan daftar properti-properti objek yang sedang terpilih. Sebagai contohnya anda dapat mengubah warna tulisan (*foreground*) dan warna latar belakang (*background*).

6. *Jendela Project*

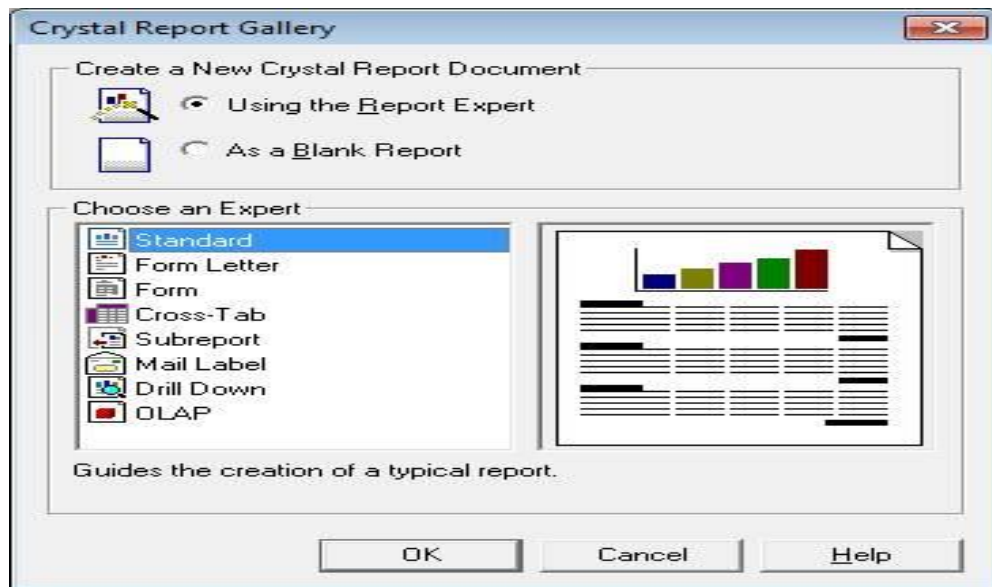
Jendela ini berisi gambaran dari semua modul yang terdapat dalam aplikasi.

2.7.2 *Crystal Report*

Crystal Report adalah salah satu program tunggal yang telah berintegrasi dengan *Microsoft Visual Basic*, sehingga menggunakan *Crystal Report* ini memungkinkan lagi seorang programmer menciptakan sebuah laporan yang lengkap. Dengan memiliki laporan yang didesain secara terpisah dan pada akhirnya nanti hasil *Crystal Report* dapat diaktifkan melalui *Microsoft Visual Basic*. *Crystal Report* dapat juga membaca sumber daya yang dihasilkan melalui *Microsoft Visual Basic*

Beberapa kelebihan *Crystal Report* antara lain sebagai berikut :

1. Dari segi pembuatan laporan, tidak terlalu rumit yang memungkinkan para *programmer* pemula dapat membuat laporan tanpa melibatkan banyak kode program.
2. Interaksi dengan bahasa pemrograman lain yang memungkinkan dapat digunakan oleh banyak *programmer* dengan masing-masing keahlian.
3. Fasilitas *impor* hasil laporan yang mendukung format data seperti *Microsoft Word*, *Microsoft Excel*, *Microsoft Access* dan sebagainya.



Gambar 2.7 Tampilan Menu Utama *Crystal Report*

2.7.3 Microsoft Access 2010

Microsoft Access 2010 atau disingkat *Access 2010* merupakan salah satu perangkat lunak yang tergolong *Relational Database Management System* (RDMS) yang banyak digunakan saat ini. Perangkat lunak ini sudah termasuk dalam aplikasi paket *Microsoft Office 2010*. *Access 2010* menyediakan banyak fasilitas yang berkaitan dengan pengelolaan *database*. Dengan fasilitas pada *access 2010* yang tersedia, kita dapat melakukan proses penyortiran, pengaturan data, pembuatan *tabel*, *query*, *form*, *report*, *pages*, *macros*, dan *moduls* yang sangat berguna dalam mengelola *database*. Untuk dapat membuat suatu *database* yang baik, diperlukan suatu rancangan yang tepat dan sesuai kebutuhan diantaranya ialah

1. Entitas dan *Relationship*

Hal yang sangat mendasar dan harus dipahami adalah pemodelan entitas dan *relationship*. Entitas adalah berbagai hal dalam dunia nyata yang informasinya disimpan dalam *database*. *Relationship* terdiri dari 3 derajat yang berbeda, yaitu:

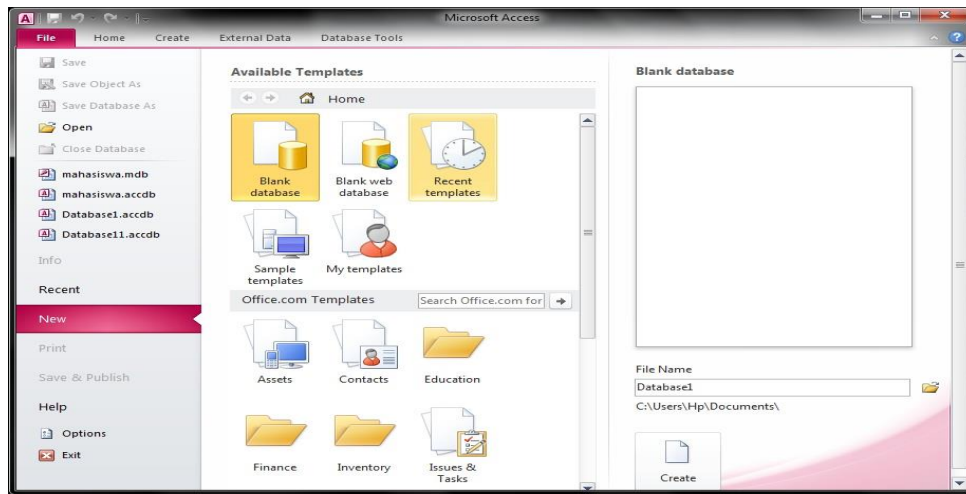
- *One-to-one* menghubungkan secara tepat dua entitas dengan satu kunci (*key*).
- *One-to-many (many-to-one)* merupakan hubungan antar entitas dimana kunci (*key*) pada satu tabel muncul berkali-kali pada tabel lainnya.
- *Many-to-many* merupakan hubungan antar entitas dimana kunci utama pada tabel pertama dapat muncul beberapa kali pada tabel kedua, dan sebaliknya.

2. Relasi atau Tabel

Relasi merupakan tabel yang mewakili entitas, dimana didalamnya terdapat kolom-kolom yang merupakan *attribute* dari entitas. Penamaan tabel juga disesuaikan dengan nama dari entitas agar mudah dipahami.

3. Kolom atau *Attribute*

Kolom atau *attribute* merupakan bagian dari tabel yang mewakili ciri dari suatu entitas.



Gambar 2.8 Tampilan Access 2010

2.8 Penelitian Sebelumnya

Adapaun penelitian sebelumnya adalah sebagai berikut:

No	Nama Penulis	Abstrak
1	1.Herowati 2.Agung Pambudi 3.Nurbayti	pendidikan di Indonesia pada saat ini sudah berkembang, sudah banyak bidang-bidang ilmu yang terlihat biasa tetapi sangat berguna jika dipelajari, contohnya ilmu dibidang budidaya perikanan, salah satu yang dapat dibudidayakan yaitu ikan hias air tawar, ikan hias ini sangat banyak digemari oleh para penggemar ikan hias. Tidak mudah memelihara dan merawat ikan hias air tawar ini, banyak sekali kendala yang disebabkan oleh faktor penyakit, lingkungan yang kurang bersih

		<p>dan organisme yang hidup disekitarnya, kemungkinan terburuk dari ikan yang terkena penyakit adalah kematian, hal tersebut sudah pasti sangat merugikan bagi para penjual dan para penggemar yang memelihara ikan hias air tawar tersebut.</p> <p>Metode pembelajaran untuk mendiagnosa penyakit ikan masih menggunakan cara manual yaitu dengan memeriksa ke laboratorium, hal tersebut membutuhkan waktu yang sangat lama, mendiagnosa gejala penyakit pada ikan membutuhkan keahlian seorang pakar yang ahli dibidang perikanan. Sistem pakar dibangun dengan mengadopsi kemampuan seorang pakar yang ahli dalam mendiagnosa penyakit ikan hias air tawar melalui gejala-gejala yang dihadapi. Fokus utama dari penelitian sistem pakar ini pada penyakit ikan hias air tawar, sistem pakar ini dibuat menggunakan pemrograman <i>Microsoft Visual Basic</i> dengan basis data <i>Microsoft Access</i>, sistem ini menggunakan <i>Metode Backward Chaining</i></p>
--	--	---

		<p>dengan Teknik <i>Breadth-First Search</i>, yaitu pencarian dengan cara menelusuri masalah secara mendalam sampai ditemukannya suatu solusi yang maksimal. <i>Output</i> dari sistem ini diharapkan dapat membantu para penjual, para penggemar ikan hias, ataupun para pelajar yang sedang mempelajari tentang ikan hias air tawar untuk mendiagnosa penyakit, dan juga memberikan solusi atau cara penanganan dari penyakit ikan hias air tawar secara cepat, tepat dan akurat.</p>
2	<p>1. Zulkarnain Lubis 2. Solly Aryza</p>	<p>Rancangan bangun miniatur rumah pintar 2 lantai dengan sensor gerak berbasis sms ini dibuat untuk mengetahui kondisi rumah dari jarak jauh melalui sms. Rancangan menggunakan mikrokontroler ATmega 8 sebagai pengendali, sensor gerak (PIR) sebagai pendeteksi gerak manusia, sensor gas dan api sebagai pendeteksi kebocoran gas dan kebakaran. Sebuah modem GSM digunakan untuk mengirim sms sebagai peringatan kepada <i>user</i> pada saat tidak berada</p>

		<p>di rumah. Sistem juga dilengkapi dengan pengendali jemuran kain yang akan mengatur proses penjemuran secara otomatis. Kontroler ATmega 8 digunakan untuk mengendalikan sistem secara keseluruhan. Setelah diuji coba hasil kerja alat cukup baik dalam merespon setiap sensor. Jika salah satu sensor mendeteksi gangguan melebihi batas yang ditentukan, buzzer akan diaktifkan kemudian sms akan dikirim ke nomor <i>user</i>. Dalam kondisi sinyal yang baik sms akan diterima dalam hitungan detik. Khusus untuk sensor gas jika rangkaian mendeteksi kebocoran gas selain buzzer dan sms diaktifkan rangkaian juga akan mengaktifkan ventilasi agar gas terhisap keluar.</p>
3	<p>1. Fitria Suryatini 2. Jaja Kustija</p>	<p>Robot bermanfaat untuk membantu manusia dalam mengerjakan pekerjaan yang berbahaya contohnya pemadam kebakaran. Oleh karena itu perlu dikembangkan robot cerdas pemadam api yang mampu mencari sumber api dan kemudian memadamkannya.</p>

		<p>Penelitian ini bertujuan membuat rancang bangun robot cerdas pemadam api menggunakan mikrokontroler ATmega128 sebagai pengendalinya. Proses pencarian sumber api dilakukan dengan cara memeriksa tiap ruangan apakah terdapat sumber api atau tidak. Pencarian titik api dilakukan dengan mendeteksi pancaran sinar <i>ultraviolet</i> yang dipancarkan api dengan menggunakan sensor <i>UVTron flame detector</i>. Untuk memadamkan api digunakan kipas yang digerakkan oleh motor DC. Robot bergerak secara otomatis menggunakan aplikasi dari motor DC. Robot menggunakan sensor PING <i>ultrasonic range finder</i> untuk memandu navigasi robot dalam pencarian ruangan, menghindari halangan, mendeteksi arah gerak dan untuk kembali ke posisi start robot. Hasil penelitian menunjukkan bahwa robot cerdas pemadam api dapat dibuat dengan menggunakan perangkat keras dan perangkat lunak yang dikontrol menggunakan mikrokontroler ATmega128. Dari pengujian yang telah</p>
--	--	--

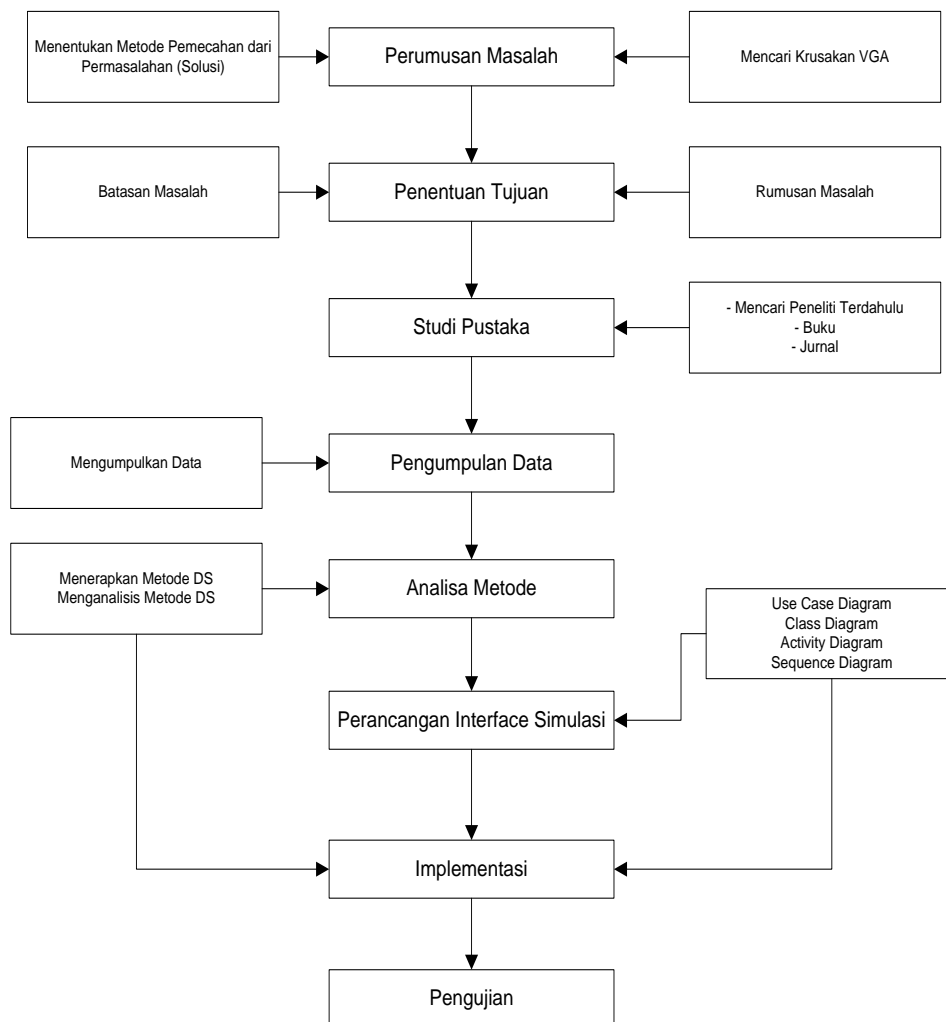
		<p>dilakukan dapat disimpulkan bahwa robot cerdas pemadam api ini dapat mendeteksi api hingga jarak 5 meter dan dapat memadamkan api dalam waktu rata-rata 38.6 detik.</p>
--	--	--

BAB III

METODE PENELITIAN

3.1 Metode Penelitian

Analisis sistem dilakukan untuk mendapatkan informasi penelitian terhadap sistem yang berjalan dan menentukan bagaimana sistem yang diusulkan agar menjawab permasalahan yang terjadi pada sistem yang berjalan.



Gambar 3.1 Tahapan Penelitian

3.2 Analisa Permasalahan

Dengan banyak fungsi dari ATMEGA328 tidak didukung dengan informasi kerusakan dari ATMEGA328 itu sendiri, maka sering juga terjadinya kerusakan pada ATMEGA328, seperti panas ATMEGA328 yang menghasilkan alat tidak berfungsi, terjadinya arus pendek listrik. Hal ini menjadi permasalahan dalam proses perawatan dan pemakaian ATMEGA328 bagi user tidak mengetahui bagaimana cara menjaga atau merawat ATMEGA328 itu sendiri.

3.3 Algoritma Sistem

Dalam menyelesaikan permasalahan yang terjadi tentang resiko atau Kerusakan yang ditimbulkan dalam penggunaan ATMEGA328 berdasarkan Tanda-Tanda yang terjadi pada perangkat maka dibutuhkan suatu sistem yang mampu mengadopsi proses dan cara berpikir seorang pakar dapat memiliki lebih dari satu Kerusakan. Adapun algoritma sistem diagnosa adalah sebagai berikut :

1. Menentukan Sumber Pengetahuan
2. Membuat Nilai Densitas Dari Kerusakan
3. Menerapkan Metode *Case Base Reasoning*
4. Menentukan Hasil Diagnosa

3.3.1 Menentukan Sumber Pengetahuan

Sumber data pengetahuan dari seorang pakar di PT. Nisa Indonesia Sentosa yang bernama Sasli Pranoto Simarmata, ST. ini tentunya menjadi acuan dasar sistem dalam menarik suatu kesimpulan, sehingga tabel pengetahuan ini sangat menentukan dalam proses perhitungan dan hasil diagnosa terhadap resiko yang

ditimbulkan bagi user yang menggunakan ATmega328. Berikut tabel pengetahuan yang akan digunakan dalam mendiagnosa resiko yang ditimbulkan dalam penggunaan ATmega328

Tabel 3.1 Jenis Resiko Penggunaan VG

No	Kode Resiko	Jenis Resiko
1	R01	Rusak Parah
2	R02	Rusak Perbaikan

Tabel 3.2 Sumber Pengetahuan Kerusakan Resiko ATmega328

No.	Tanda-Tanda Resiko Pada ATmega328	Kode Tanda
1	Tegangan operasi Rusak	K01
2	Pin input/output digital rusak	K02
3	Pin input daya analog rusak	K03
4	Arus DC untuk setiap pin input/output rusak	K04
5	Arus DC untuk 3.3V Pin rusak	K05
6	SRAM rusak.	K06
7	CLK Speed bermasalah	K07

Sumber : (PT. Nisa Indonesia Sentosa, Medan Sunggal)

3.3.2 Nilai Densitas

Setelah menentukan sumber pengetahuan mengenai Penyebab dan jenis resiko pengguna ATmega328, tahap selanjutna menentukan nilai densitas dari Tanda-Tanda dan prognosis resiko tersebut adalah sebagai berikut

Tabel 3.3 Basis Aturan Resiko ATmega328

No	Kode Kerusakan	Nama Kerusakan	Jumlah User	Nilai Densitas
1	K1	Tegangan operasi Rusak	40	0.8
2	K2	Pin input/output digital rusak	40	0.8
3	K3	Pin input daya analog rusak	20	0.4
4	K4	Arus DC untuk setiap pin input/output rusak	35	0.7
5	K5	Arus DC untuk 3.3V Pin rusak	30	0.6
6	K6	SRAM rusak.	40	0.8
7	K7	CLK Speed bermasalah	27	0.5

1. Tabel Keputusan

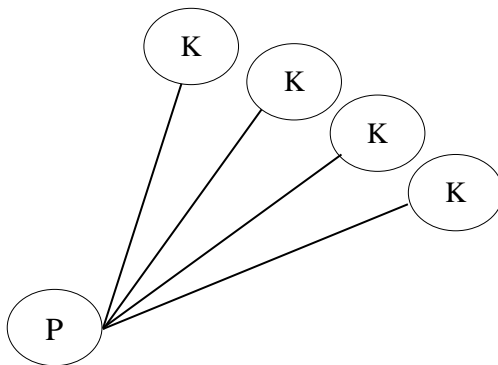
Dari gejala dan resiko yang diketahui maka dapat disimpulkan basis pengetahuan berupa hubungan antara gejala dan resiko yang ada pada user yang menggunakan ATmega328. Basis pengetahuan dapat dilihat pada tabel berikut :

Tabel 3.4 Data Keputusan

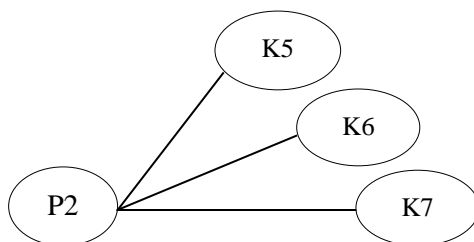
Kode Kerusakan	Jenis Kerusakan	Kerusakan
K1	Tegangan operasi Rusak	Rusak Perbaikan

K2	Pin input/output digital rusak	
K3	Pin input daya analog rusak	
K4	Arus DC untuk setiap pin input/output rusak	
K5	Arus DC untuk 3.3V Pin rusak	
K6	SRAM rusak.	Rusak Parah
K7	CLK Speed bermasalah	

2. Pohon Keputusan



Gambar 3.2 Pohon Keputusan Rusak Perbaikan (P1)



Gambar 3.3 Pohon Keputusan Rusak Parah (P2)

3. Data Sampel Kasus

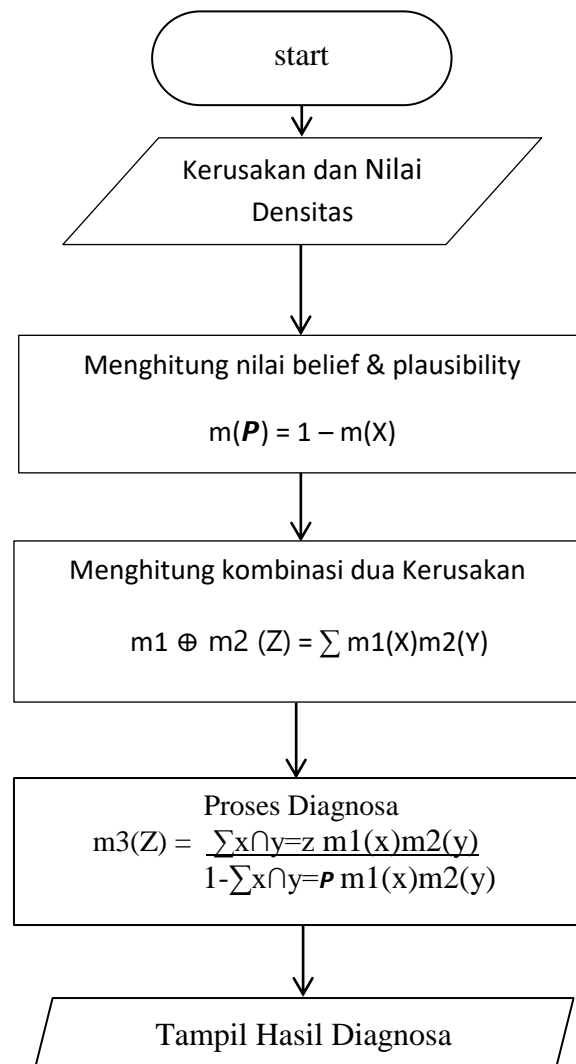
Data sampel kasus ini dihasilkan dari hasil analisa pada resiko penggunaan ATMega328. Berikut ini merupakan data sampel kasus :

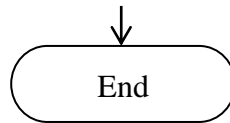
Tabel 3.5 Sampel Kasus

No	Gejala	Hasil Diagnosa Resiko
1	K5, K6, K7	Rusak Parah
2	K1, K2, K3, K4	Rusak Perbaikan

3.4 Flowchart Program

Flowchart Program merupakan keterangan yang lebih rinci mengenai bagaimana prosedur yang sesungguhnya yang dilakukan oleh program.





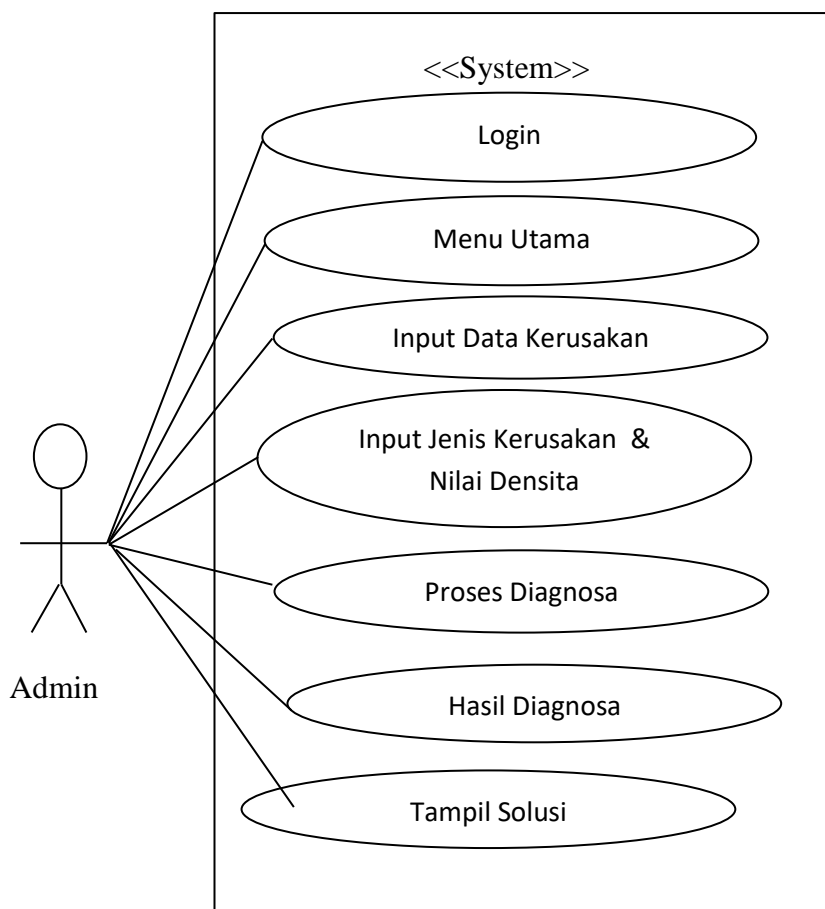
Gambar 3.4 Flowchart Program Menggunakan *Dempster-Shafer*

3.5 Perancangan Sistem

Perancangan sistem yang akan digunakan dalam penelitian ini adalah metode UML (Unified Modelling Language) yang akan dijelaskan melalui *Use Case*, *Activity Diagram* dan *Class Diagram*.

3.5.1 Use Case Diagram

Use Case Diagram menunjukkan hubungan interaksi antara aktor dengan *Use Case* didalam suatu sistem yang bertujuan untuk menentukan bagaimana aktor berinteraksi dengan sebuah sistem



Gambar 3.5 Use Case Diagram

1. Skenario *Use Case Login*

<i>Use Case</i>	: Login
Aktor	: Admin
Tujuan	: Melindungi data pada program agar tidak sembarangan orang mengakses menu admin dan mengubah data pada program misalkan data jenis Kerusakan, data jenis gejala dan data basis pengetahuan.
Deskripsi	:Admin menjalankan sistem, kemudian tampil halaman menu utama selanjutnya admin mengklik tombol login yang ada pada halaman menu utama dan sistem menampilkan halaman login. Selanjutnya admin memasukan username dan password. Jika berhasil sistem akan menampilkan menu admin, jika tidak kembali ke halaman login.

2. Skenario *Use Case Menu Utama*

Nama <i>Use Case</i>	: Menu Utama
Aktor	: Admin
Tujuan	:Menampilkan menu utama.
Deskripsi	:Admin melakukan proses login lebih dulu setelah itu tampil menu utama.

3. Skenario *Use Case Data Kerusakan*

Nama <i>Use Case</i>	: Mengelola Jenis Kerusakan dan Nilai Densitas
Aktor	: Admin

Tujuan :Mengelola jenis Kerusakan dengan menginput, menambah, mengurai atau mengedit jenis Kerusakan serta menambahkan nilai densitasnya.

Deskripsi : Admin masuk ke menu utama dan memilih dan menginput data Kerusakan.

4. Skenario *Use Case* Input Kerusakan

Nama *Use Case* : Input Kerusakan

Aktor : Admin

Tujuan : Memilih gejala sesuai dengan yang dialami user.

Deskripsi : Admin menjalankan sistem, kemudian memilih Kerusakan yang ditampilkan oleh sistem.

5. Skenario *Use Case* Proses Diagnosa

Nama *Use Case* : Proses Diagnosa

Aktor : Admin

Tujuan :Melakukan proses untuk mendapatkan hasil diagnosa yang dipilih.

Deskripsi :Setelah admin memilih gejala yang sesuai dengan yang dialami user, sistem akan menjalankan proses perhitungan Case Base Reasoning.

6. Skenario *Use Case* Tampil Solusi

Nama *Use Case* : Tampil Solusi

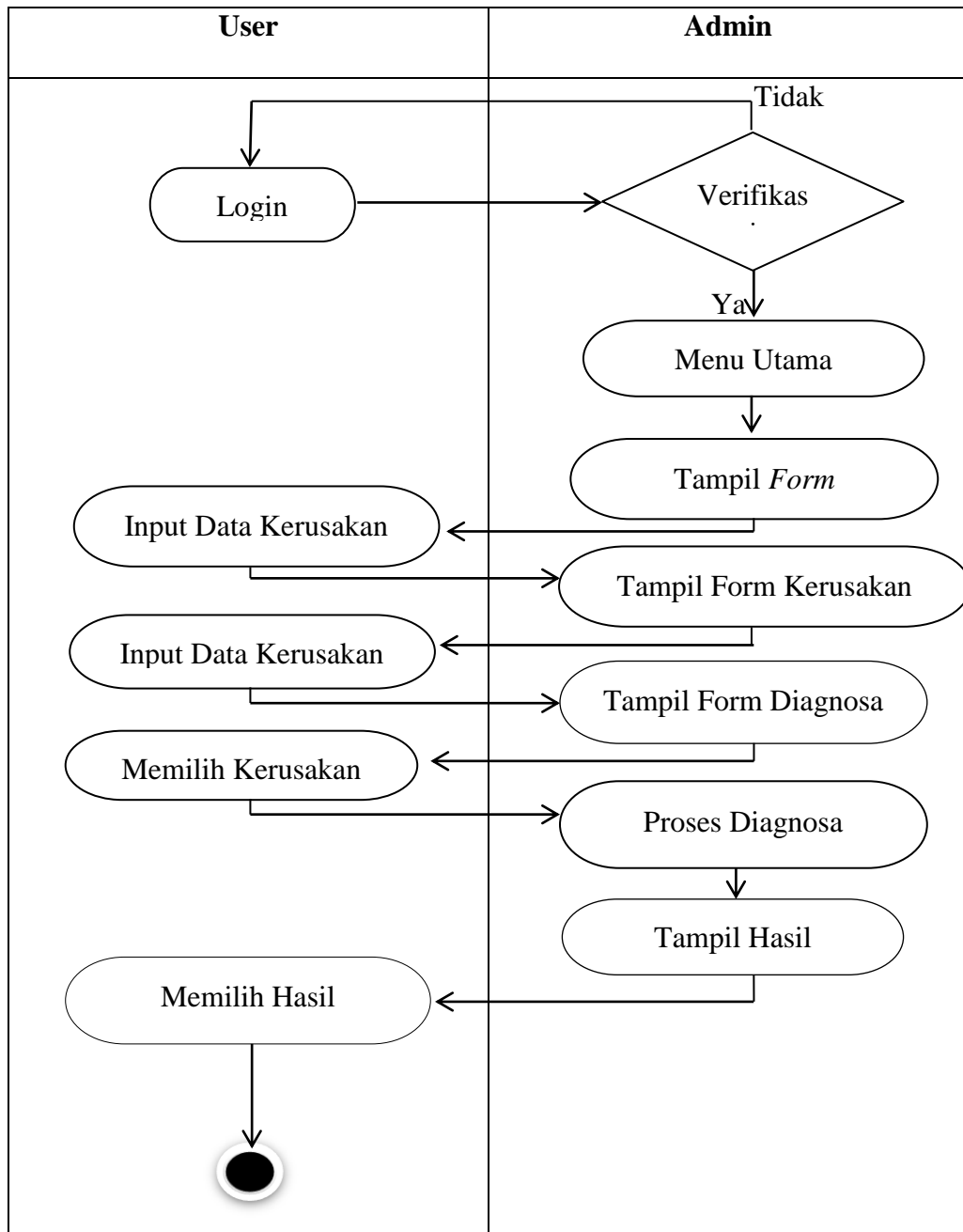
Aktor : Admin

Tujuan : Menunjukkan hasil diagnosa kepada *user*.

Deskripsi :Setelah menjalankan proses, sistem akan menampilkan hasil diagnosa dan memberikan solusi untuk penanganan lebih lanjut terhadap penyakit yang terdiagnosa.

3.5.2 Activity Diagram

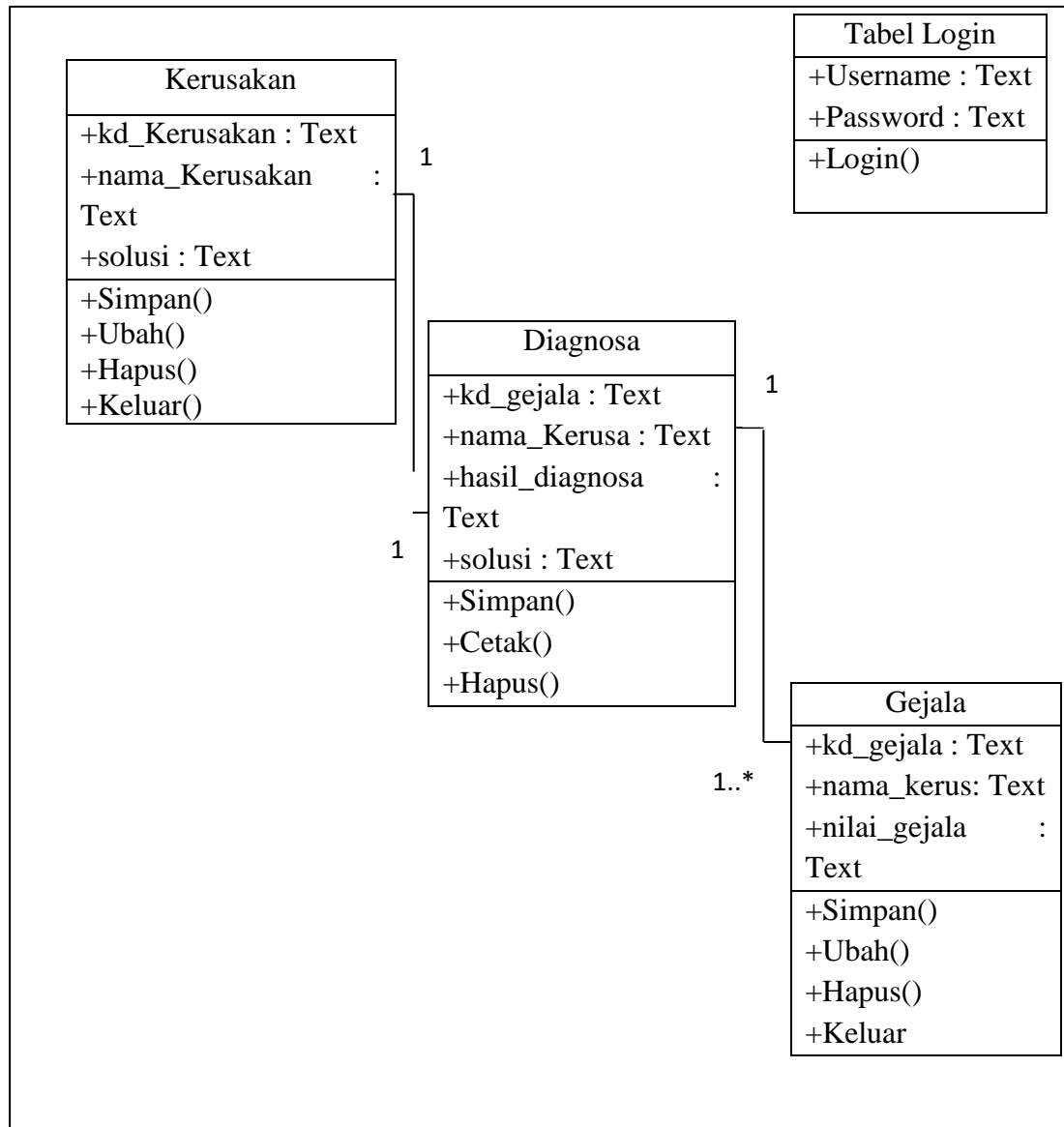
Diagram aktivitas menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor



Gambar 3.6 Activity Diagram Admin

3.5.3 Class Diagram

Class Diagram merupakan diagram yang menggambarkan sekelompok objek dari atribut, operasi dan relasi objek



Gambar 3.7 Class Diagram Resiko Menggunakan ATMega328

3.6 Rancangan Basis Data

Rancangan basis data digunakan untuk dapat melihat tabel atau *file* yang digunakan dalam memenuhi kebutuhan sebuah sistem. Rancangan basis data pada sistem pakar mendiagnosa resiko bagi user yang menggunakan ATMega328 dengan menggunakan metode *Case Base Reasoning* adalah sebagai berikut :

1. Tabel Login

Tabel 3.6 Tabel *Login*

No	Field Name	Data Type	Size	Description
1	user_name	Text	100	User Name
2	Password	Text	10	Password

2. Tabel Gejala

Tabel 3.7 Tabel Kerusakan

No	Field Name	Data Type	Size	Description
1	kd_kerusakan	Text	5	Kode Kerusakan
2	nama_kerusakan	Text	50	Nama Kerusakan
3	nilai_densitas	Number	Long Intenger	Nilai Densitas

3. Tabel Kerusakan

Tabel 3.8 Tabel Kerusakan

No	Field Name	Data Type	Size	Description
1	kd_Kerusakan	Text	5	Kode Kerusakan
2	nama_Kerusakan	Text	30	Nama Kerusakan
3	Solusi	Text	80	Solusi Kerusakan

4. Tabel Rule

Tabel 3.9 Tabel Rule

No	Field Name	Data Type	Size	Description
1	kd_rule	Text	3	Kode Rule
2	kd_Kerusakan	Text	4	Kode Kerusakan
3	kd_Kerusakan	Text	3	Kode Kerusakan

5. Tabel Diagnosa

Tabel 3.10 Tabel Diagnosa

No	Field Name	Data Type	Size	Description
1	kd_Kerusakan	Text	5	Kode Kerusakan
2	nama_Kerusakan	Text	50	Nama Kerusakan
3	tanggal	Date/Time		Tanggal

Tabel 3.11 Tabel Diagnosa (Lanjutan)

No	Field Name	Data Type	Size	Description
3	hasil_diagnosa	Text	50	Hasil Diagnosa
4	Solusi	Text	100	Solusi Kerusakan

3.7 Perancangan Antarmuka (*Interface*)

Perancangan masukan sistem merupakan gambaran dari *form* dari pada sistem yang akan dirancang. Berikut tampilan rancangan masukan pada Sistem Pakar Mendiagnosa Tingkat Resiko Bagi User Pengguna ATMEGA328.

1. Rancangan Form Menu Login

Menu utama adalah tampilan awal ketika *user* memasuki sistem. Halaman ini berisi tampilan luar tentang sistem pakar untuk mendiagnosa Kerusakan.

The diagram shows a window titled "Form Login" with standard window controls (minimize, maximize, close). Inside the window, there is a "Logo" box on the left. To the right of the logo are two input fields labeled "Username" and "Password". Below these fields are two buttons labeled "Login" and "Cancel".

Gambar 3.8 *Form* Menu Login

2. Rancangan *Form* Menu Utama

Form Buku Tamu adalah form adalah form yang digunakan untuk menginputkan siapa-siapa saja yang telah menggunakan sistem yang dirancang ini, sebelum pengunjung melakukan diagnosa, mereka wajib mengisi form ini.

Menu Utama				-	<input type="checkbox"/>	X
Data	Diagnosa	Laporan	Keluar			
Kerusakan	Case Base Reasoning	Laporan Hasil Diagnosa				
Kerusakan						
Rule						

Gambar 3.9 Form Menu Utama

3. Rancangan Form Kerusakan

Form Kerusakan digunakan oleh user. Pada form gejala ini user diharuskan untuk memilih Kerusakan sesuai dengan Kerusakan. Berikut adalah rancangan form diagnosa.

Form Kerusakan			-	<input type="checkbox"/>	X
Kode Kerusakan	<input type="text"/>	Tambah			
Nama Kerusakan	<input type="text"/>				
Nilai Densitas	<input type="text"/>				
Simpan	Ubah	Hapus	Batal	Keluar	
Kode Kerusakan	Nama Kerusakan	Densitas			

Gambar 3.10 Form Kerusakan

4. Rancangan Form Kerusakan

Form Hasi Diagnosa ini merupakan form untuk menampilkan hasil diagnosa berdasarkan Kerusakan yang dipilih sebelumnya.

<input type="button" value="-"/> <input type="button" value="X"/>		
Kode	<input type="text"/>	<input type="button" value="Tambah"/>
Nama Kerusakan	<input type="text"/>	
Solusi	<input type="text"/>	
<input type="button" value="Simpan"/>	<input type="button" value="Ubah"/>	<input type="button" value="Hapus"/>
<input type="button" value="Batal"/>		<input type="button" value="Keluar"/>
Kode Kerusakan	Nama Kerusakan	Solusi
<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>

Gambar 3.11 Form Kerusakan

5. Rancangan Form Rule

Perancangan Form Rule dibuat untuk merancang bentuk desain halaman web yang akan digunakan oleh admin untuk menuju ke form Kerusakan, form basis pengetahuan dan form kerusakan.

FormRule

- X

Kode Rule

Kode Kerusakan

Kode Kerusakan

Kode Pengetahuna

Nama

Nama Kerusakan

Simpan

Ubah

Hapus

Batal

Keluar

Kode Rule	Kode Kerusakan	Nama Kerusakan	Kode Kerusakan	Nama Kerusakan

Gambar 3.12 Form Rule

6. Rancangan Menu Diagnosa

Form diagnosa digunakan untuk melihat Kerusakan yang ada di database, menghapus Kerusakan, menambah Kerusakan dan mengubah Kerusakan.

Berikut adalah perancangan untuk halaman form Kerusakan.

Diagnosa		-	□	X
Nama	<input type="text"/>	Kemungkinan Diagnosa		
Tanggal	<input type="text"/>	Hasil Diagnosa		
Pilih Gejala Yang Dialami		<input type="text"/>	<input type="button" value="Diagnosa"/> <input type="button" value="Simpan"/> <input type="button" value="Batal"/> <input type="button" value="Keluar"/>	
<input type="checkbox"/>	[g01] xxxxxxxxxxxx			
<input type="checkbox"/>	[g02] xxxxxxxxxxxx			
<input type="checkbox"/>	[g03] xxxxxxxxxxxx			

Gambar 3.13 Form Diagnosa

7. Rancangan Laporan Hasil Diagnosa

Pada bagian ini dilengkapi dengan rancangan halaman login. Halaman login digunakan khusus untuk admin web yang dapat mengakses form Kerusakan, form gejala, form basis pengetahuan.

Logo	Sistem Pakar Diagnosa Kerusakan ATMEGA328				X
Laporan Hasil Diagnosa					
Nama Pengguna	Kode Kerusakan	Nama Kerusakan	Nilai Densitas	Solusi	
xxxxxxxx xxxxxxxxxxxxxxxxxxxx					

Gambar 3.14 Form Hasil Diagnosa

BAB IV

IMPELEMENTASI DAN PENGUJIAN

4.1 Kebutuhan Sistem

Kebutuhan sistem adalah kegiatan akhir dari proses penerapan sistem baru dimana sistem yang baru ini akan dioperasikan secara menyeluruh. Sebelum sistem benar-benar bias digunakan dengan baik, sistem harus melalui tahap pengujiannya terlebih dahulu untuk menjamin tidak ada kendala yang muncul pada saat sistem digunakan.

1.1.1 Kebutuhan Perangkat Keras

Perangkat keras adalah peralatan di sistem computer secara fisik terlihat dan dapat diolah. Adapaun perangkat keras yang digunakan pada implementasi sistem pakara dalah

1. *Processor* Minimal Intel Atom 1.66 GHz
2. RAM Minimal 1 GB
3. *Harddisk* Minimal 160 GB

1.1.2 Kebutuhan Perangkat Lunak

1. Aplikasi pemograman *Microsoft Visual Basic 2008*
2. *Xampp*
3. *Crystal Report 8.5*

4.2 Implementasi Sistem

Implementasi sistem merupakan hasil rancangan yang disebut menjadi hasil implementasi perancangan form kedalam bahasa pemrograman *Visual*. Berikut

hasil implementasi mendiagnosa kerusakan pada ATMEGA328 menggunakan aplikasi sistem pakar dengan metode *case base reasoning*. Penjelasan implementasi sistem dijabarkan pada langkah-langkah berikut :

4.2.1 Login

Seperti pada umumnya, sebuah aplikasi adanya sistem *login* akan memberikan kemudahan pengguna untuk menggunakan sistem yang telah dirancang. Berikut merupakan tampilan sistem *login* :

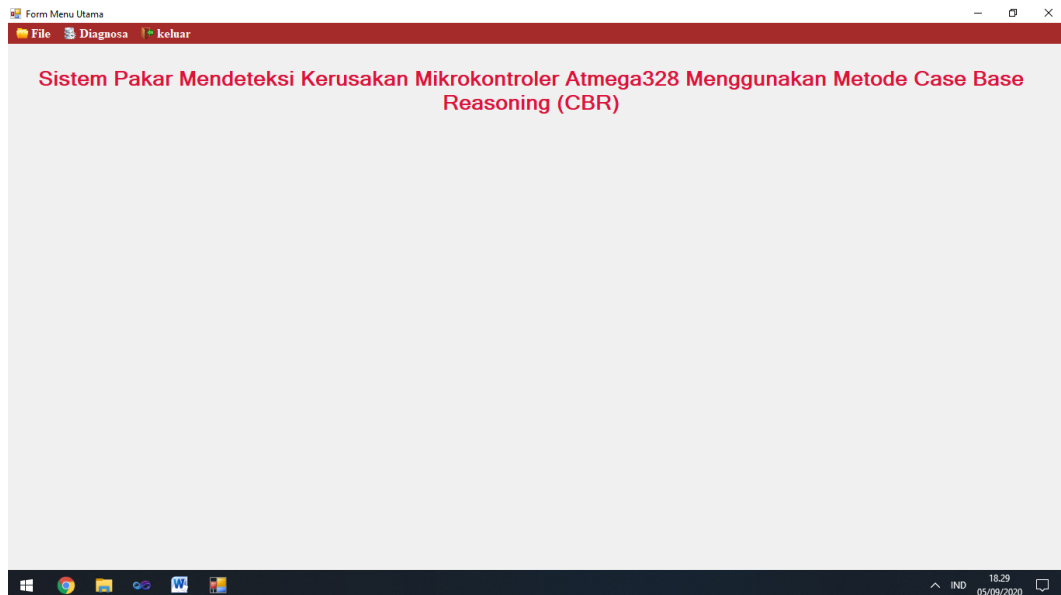


The image shows a screenshot of a login window titled "Form Login". The window has a red background and a white border. At the top center, it says "SILAHKAN LOGIN". Below this, there are two input fields. The first is labeled "USER" with a person icon and contains the text "dini". The second is labeled "PASSWORD" with a key icon and contains "####". At the bottom right, there are two buttons: "Login" with a green checkmark icon and "Cancel" with a red X icon.

Gambar 4.1 *Form Login*

4.2.2 Menu Utama

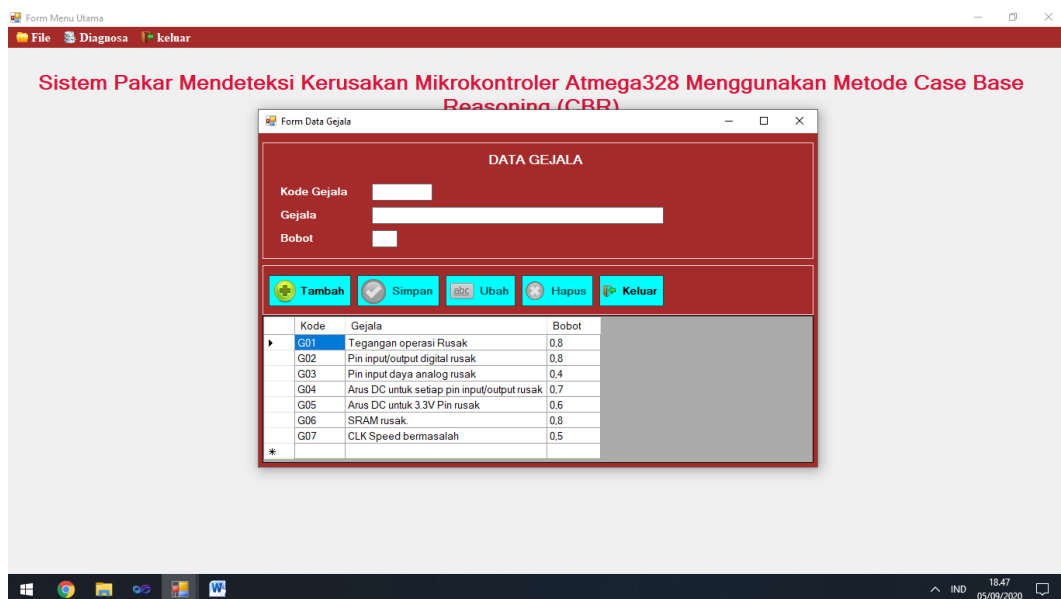
Form menu utama admin memilih menu awal. Menu utama bagi *admin* berupa submenu berupa *input* data gejala, data kasus, data pemilik ATMEGA328, *submenu* proses diagnosa, *submenu* laporan diagnosa dan *submenu* keluar. Adapun tampilan menu utama adalah sebagai berikut



Gambar 4.2 *Form* Menu Utama

4.2.3 *Form* Data Kerusan

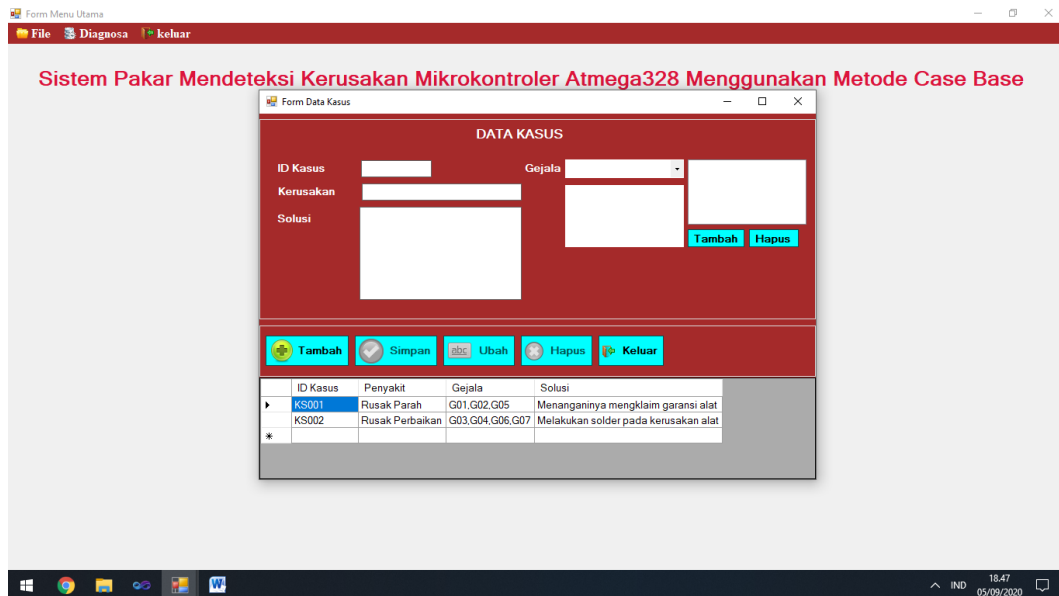
Form data Kerusakan adalah *form* yang berguna untuk simpan, ubah, hapus, dan keluar. Adapun tampilan dari menu kerusakan dari *form* data gejala adalah sebagai berikut



Gambar 4.3 *Form* Data Kerusakan

4.2.4 Form Data Kasus

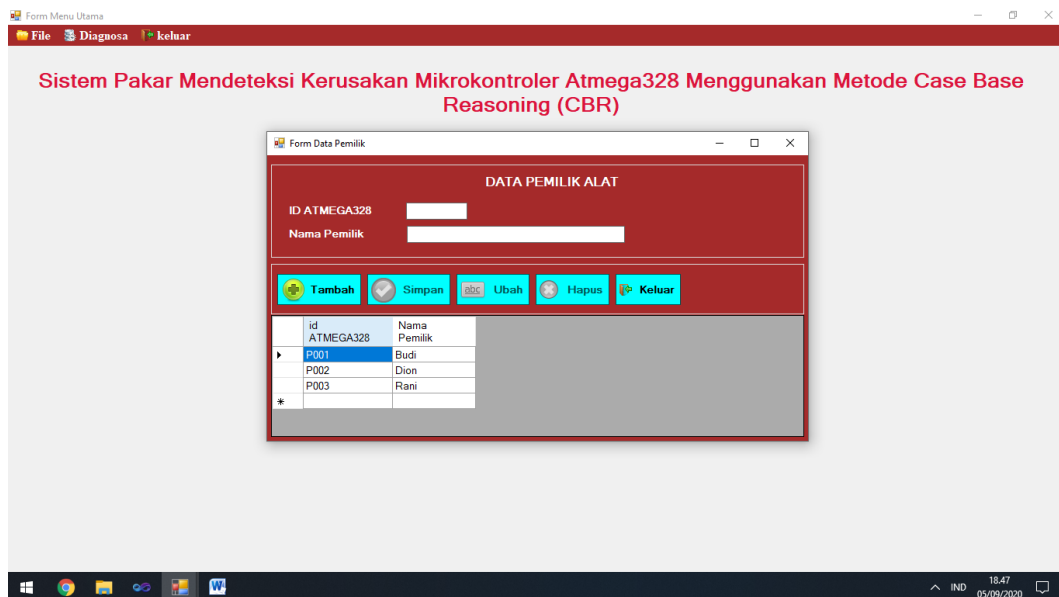
Form data Kasus adalah form yang berguna untuk simpan, ubah, hapus, dan keluar. Adapun tampilan menu data kasus dari form data kasus adalah sebagai berikut



Gambar 4.4 Form Data Kasus

4.2.5 Form Data Pemilik ATMEGA328

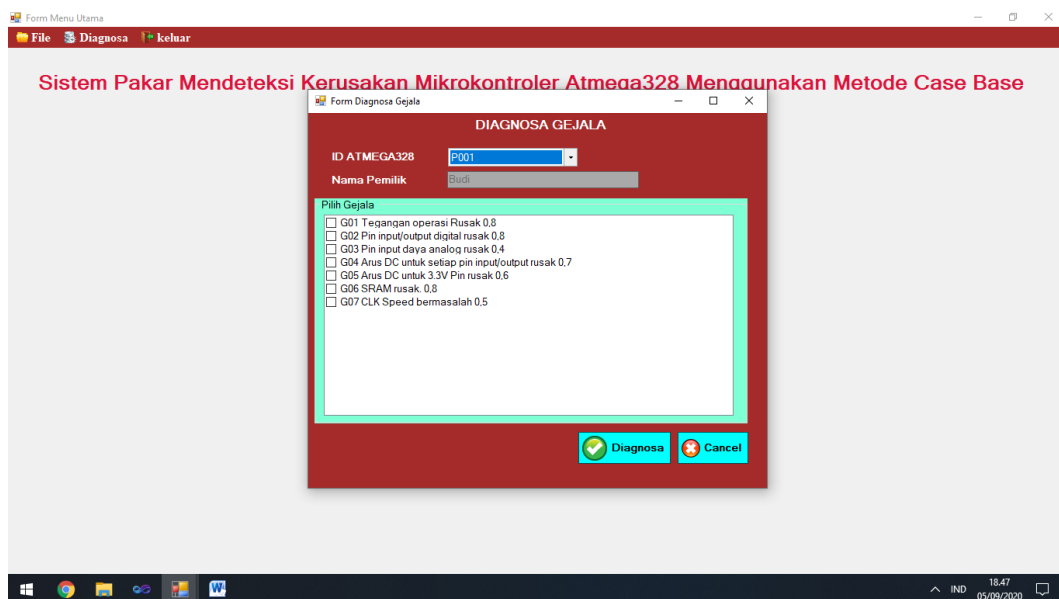
Form data pemilik ATMEGA328 adalah form yang berguna untuk simpan, ubah, hapus, dan keluar. Adapun tampilan menu pemilik dari form data pemilik ATMEGA328 adalah sebagai berikut



Gambar 4.5 Form Data Pemilik ATMEGA328

4.2.6 Form Diagnosa Kerusakan

Form Diagnosa Kerusakan adalah form yang berguna untuk proses diagnosa, dan keluar. Adapun tampilan dari menu diagnosa Kerusakan dari form diagnosa adalah sebagai berikut



Gambar 4.6 Form Diagnosa Kerusakan

4.2.7 Form Proses Diagnosa

Form proses diagnosa adalah *form* yang berguna untuk Lihat Hasil dan keluar. Adapun tampilan menu diagnosea dari *form* proses diagnosa adalah sebagai berikut

The screenshot shows a Windows application window titled "Form Proses Diagnosa". The window has a red title bar and a menu bar with "File", "Diagnosa", and "keluar". The main content area is titled "PROSES DIAGNOSA" and contains the following elements:

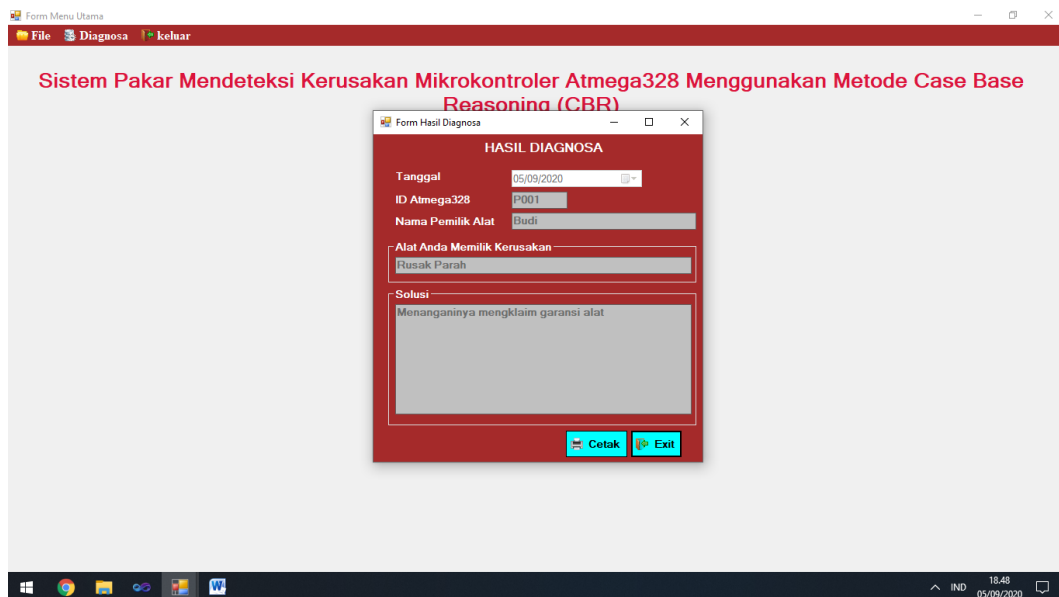
- Informasi Pemilik Alat:** ID Atmega328, P002
- Nama Pemilik:** Dion
- Gejala Baru:** G01
- Gejala Lama:** G01, G02, G05
- Gejala Sama/Tidak:** G01 (sama)
- Bobot:** 1
- Pilih Id Kasus:** KS001, KS002
- Similarity:** 1
- Nama Kerusakan:** Rusak Parah
- Buttons:** "Lihat Hasil" and "Keluar"

The window is set against a background that says "Sistem Pakar Mendet" and "Metode Case Base". The Windows taskbar at the bottom shows the search bar, taskbar icons, and system tray with the time 15:037 and date 26/10/2020.

Gambar 4.7 Form Proses Diagnosa Kerusakan Pada ATMEGA328

4.2.8 Form Hasil Diagnosa

Form Hasil diagnosa adalah *form* yang berguna untuk Lihat Simpan, Cetak dan keluar. Adapun tampilan menu Hasil diagnosa dari form Hasil diagnosa adalah sebagai berikut



Gambar 4.8 Form Hasil Diagnosa Kerusakan Pada ATMEGA328

4.3 Kelebihan dan Kelemahan Sistem

Kelebihan dan Kelemahan Sistem dijelaskan agar pengguna dapat mengetahui letak kelebihan dan kelemahan dari sistem yang telah dirancang dalam mendiagnosa kerusakan pada ATMEGA328 dengan menggunakan metode *Case Base Reasoning*.

1. Kelebihan Sistem

Adapun yang menjadi kelebihan pada sistem yang telah dirancang adalah sebagai berikut :

- a. Aplikasi sistem pakar dalam mendiagnosa kerusakan pada ATMEGA328 menggunakan metode *Case Base Reasoning* dapat mempermudah pemilik ATMEGA328 dalam melakukan proses diagnosa kerusakan pada ATMEGA328.
- b. Pengguna akan lebih mudah melihat hasil didalam proses diagnosa kerusakan pada ATMEGA328.

- c. Pengguna dapat langsung mencetak hasil laporan diagnosa kerusakan pada ATMEGA328 melalui *form* laporan.

2. Kelemahan Sistem

Adapun yang menjadi kelemahan pada sistem yang telah dirancang adalah sebagai berikut :

- a. Hasil ini hanya digunakan pada kasus sistem pakar dalam mendiagnosa kerusakan pada ATMEGA328 menggunakan metode *Case Base Reasoning*. Tidak dapat digunakan pada untuk Kerusakan lain.
- b. Sistem yang dibangun hanya berbasis *Dekstop*.
- c. Sistem tidak mampu untuk digunakan multi *user* karena sistem hanya dibangun untuk *single user* dan masih bersifat *offline*.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Dari hasil mendiagnosa kerusakan pada ATmega328 dengan menggunakan Metode *Case Base Reasoning* diperoleh kesimpulan sebagai berikut:

1. Kerusakan pada ATmega328 dapat didiagnosa berdasarkan gejala yang ditimbulkan.
2. Dalam merancang dan membangun aplikasi sistem pakar mendiagnosa kerusakan pada ATmega328 menggunakan metode *Case Base Reasoning* dilakukan dengan cara menerapkan konsep-konsep *flowchart*, UML (*Unified Modeling Language*) dan *Desktop Programming* didalamnya yang dimana *flowchart* dan UML tersebut merupakan gambaran arsitektur dari program yang dibuat.
3. Sistem pakar untuk mendiagnosa kerusakan pada ATmega328 dapat membuat sebuah sistem.
4. Dapat mengimplementasikan sistem pakar agar dapat digunakan dalam mendiagnosa kerusakan pada ATmega328.

5.2 Saran

Adapun saran yang dapat diberikan setelah melakukan pembahasan mengenai perangkat lunak Sistem Pakar untuk Mendiagnosa Kerusakan pada ATmega328 dengan menggunakan Metode *Case Base Reasoning* berbasis *desktop* adalah:

1. Sistem pakar dengan menggunakan metode *Case Base Reasoning* ini sebaiknya dapat difungsikan untuk banyak kalangan pengguna sehingga aplikasi ini dapat dikunjungi oleh banyak orang.
2. Sistem ini hendaknya dilakukan pengembangan secara terus-menerus untuk dapat memudahkan pengguna dalam mendiagnosa kerusakan pada ATmega328 tanpa harus berjumpa langsung dengan pakar.
3. Sistem ini diharapkan terus dikembangkan sehingga memenuhi standar pengembangan kerusakan pada ATmega328.

DAFTAR PUSTAKA

- Arianto, Jefri. "Studi Keandalan Sistem Distribusi 20 KV Berbasis GIS (Geographic Information Sistem) Dengan Menggunakan Metode RIA (*Reliability Index Assesment*)". Surabaya: 2015.
- Ashari, Avisena. "Macam-Macam Perubahan Energi Dan Contoh Perubahan Energi Disekitar Kita". Jakarta: 2021. Diakses pada 6 September 2021. dari: <https://bobo.grid.id/read/082561302/macam-macam-perubahan-energi-dan-contoh-perubahan-energi-di-sekitar-kita?page=all>.
- Aryza, S., Irwanto, M., Lubis, Z., Siahaan, A. P. U., Rahim, R., & Furqan, M. (2018). A Novelty Design Of Minimization Of Electrical Losses In A Vector Controlled Induction Machine Drive. In IOP Conference Series: Materials Science and Engineering (Vol. 300, No. 1, p. 012067). IOP Publishing.
- Badi. "Segitiga Daya". Jakarta: 2021. Diakses pada 6 September 2021. dari: <https://thecityfoundry.com/segitiga-daya/>.
- Beritajambi.co. "Pengertian Energi Dan Macam Satuan Energi". Jambi: 2017. Diakses pada 6 September 2021. dari: <https://beritajambi.co/read/2017/03/22/942/pengertian-energi--macam-dan-stauan-energi>.
- Energi, Ezkhel. "Sekring Putus". Jakarta: 2013. Diakses pada 6 September 2021. dari: <https://ezkhelenergy.blogspot.com/2013/11/fuse-cut-out.html>.
- Hajar, Ibnu dkk, "Analisa Nilai SAIDI SAIFI Sebagai Indeks Keandalan Penyedia Tenaga Listrik Pada Penyulang Chaya PT PLN (Persero) Area Ciputat". Jakarta: 2018.
- Hamdani, H., Tharo, Z., & Anisah, S. (2019, May). Perbandingan Performansi Pembangkit Listrik Tenaga Surya Antara Daerah Pegunungan Dengan Daerah Pesisir. In Seminar Nasional Teknik (Semnastek) Uisu (Vol. 2, No. 1, pp. 190-195).
- Nilakandi, Zuhroh. "Pengertian Energi Beserta Manfaat, Sifat Dan Jenis-jenis Energi". Jakarta: 2019. Diakses pada 6 September 2021. dari: <https://www.nesabamedia.com/pengertian-energi/>.
- Parta Setiawan. "Pengertian Energi". Jakarta: 2021. Diakses pada 25 Oktober 2021. dari: <https://www.gurupendidikan.co.id/pengertian-energi/>.
- PT PLN (Persero). 1983. SPLN No.52-3: Pola Pengamanan Sistem. Jakarta.
- PT PLN (Persero). 1985. SPLN NO.64 : Spesifikasi *Fuse Cut Out*. Jakarta.

- PT PLN (Persero).1985. SPLN No.59: Keandalan pada Sistem Distribusi 20 KV dan 6 KV. Jakarta.
- PT PLN (Persero).1986. SPLN NO.68-2 : Tingkat Jaminan Sistem Tenaga Listrik Bagian Dua: Sistem Distribusi. Jakarta.
- Putri, M., Wibowo, P., Aryza, S., & Utama Siahaan, A. P. Rusiadi.(2018). An implementation of a filter design passive lc in reduce a current harmonisa. *International Journal of Civil Engineering and Technology*, 9(7), 867-873.
- Putri, dkk. “Analisis Pengamanan Transformator Distribusi 400 kVA Dengan Fuse Cut Out”, Medan, 2019.
- Rahmat, Gheschik Safiur, “Evaluasi Indeks Keandalan Sistem Jaringan Distribusi 20 KV Di Surabaya menggunakan *Loop Restoration Scheme*”. Digilib ITS, Surabaya: 2013.
- Roger, C. Dugan. “Kualitas Daya Listrik”. Semarang: 2004. Universitas Muhammadiyah Semarang.
- Santoso, R. Nurhalim. “Evaluasi Tingkat Keandalan Jaringan 20 KV Pada Gardu Induk Bangkinang Dengan Menggunakan Metode FMEA (*Failure Mode Effect Analysis*)”. Riau: 2016.
- Saodah, Siti. “Evaluasi Keandalan Sistem Distribusi Tenaga Listrik Berdasarkan SAIDI dan SAIFI”. Yogyakarta: 2008. Institut Teknologi Nasional.
- Sopyandi, Andi. “Tipe-tipe Jaringan Distribusi Tegangan Menengah”. Jakarta: 2011. Diakses pada 6 September 2021. dari:
<https://electricdot.wordpress.com/2011/08/16/tipe-tipe-jaringan%20distribusi-tegangan-menengah>.
- Suhadi, dkk. “Teknik Distribusi Tenaga Listrik Jilid 1”. Direktorat Pembinaan Sekolah Menengah Kejuruan. Jakarta: 2008.
- Suswanto, Daman. “Sistem Distribusi Tenaga Listrik”. Padang : 2009. Fakultas Teknik Universitas Negeri Padang.
- Tarigan, A. D., & Pulungan, R. (2018). Pengaruh Pemakaian Beban Tidak Seimbang Terhadap Umur Peralatan Listrik. *RELE (Rekayasa Elektrikal dan Energi): Jurnal Teknik Elektro*, 1(1), 10-15.