

**GENERATOR SET UJI UNTUK DIAGNOSIS KESALAHAN
RANGKAIAN KOMBINASIONAL *MULTIPLEXER 4 LINE TO
1 LINE* MENGGUNAKAN METODE *ARTIFICIAL NEURAL
NETWORK***

(Tesis)

**Oleh
DENNY PRATAMA
NPM 2225031006**



**FAKULTAS TEKNIK
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2024**

**GENERATOR SET UJI UNTUK DIAGNOSIS KESALAHAN
RANGKAIAN KOMBINASIONAL *MULTIPLEXER 4 LINE TO 1 LINE*
MENGUNAKAN METODE *ARTIFICIAL NEURAL NETWORK***

Oleh

DENNY PRATAMA

Tesis

**Sebagai Salah Satu Syarat untuk Mencapai Gelar
MAGISTER TEKNIK ELEKTRO**

Pada

**Program Pascasarjana Magister Teknik Elektro
Fakultas Teknik Universitas Lampung**



**PROGRAM PASCASARJANA MAGISTER TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS LAMPUNG BANDAR LAMPUNG**

2024

ABSTRACT**SET-TEST GENERATOR FOR DIAGNOSIS FAULTS IN A 4-LINE TO 1-LINE MULTIPLEXER COMBINATIONAL CIRCUIT USING ARTIFICIAL NEURAL NETWORK**

By

Denny Pratama

An integrated circuit (IC) is built, among other things, using combinational digital circuits. Large-scale IC production requires a method for testing using sampling methods at specific time intervals. One method involves using a set-test generator. A set-test generator is built to detect errors in combinational digital circuits. In previous research, the set-test generator was made using an arrangement of electronic circuits. Each change in the test circuit required changes in the electronic circuits of the set-test generator, which influenced, among other things, the cost of system development. This research developed a set-test generator using a software-based neural network method. Therefore, if the input and output are within the scope of the microcontroller, any changes in the test circuit only require changes to the neural network model, directly impacting cost-efficiency in system development. As a result, this research has built a set-test generator system capable of accurately detecting error locations. The model's accuracy reaches 99.95%, making the prediction results of the model highly accurate. In addition to detecting single errors, the set-test circuit can also accurately detect double and triple error locations chosen.

Keywords: set-test generator, neural network, single error, multiple error, test circuit.

ABSTRAK

GENERATOR SET UJI UNTUK DIAGNOSIS KESALAHAN RANGKAIAN KOMBINASIONAL *MULTIPLEXER 4 LINE TO 1 LINE* MENGUNAKAN METODE *ARTIFICIAL NEURAL NETWORK*

Oleh

DENNY PRATAMA

Integrated circuit (IC) dibangun salah satunya menggunakan rangkaian digital kombinasional. Produksi IC dalam skala besar membutuhkan sebuah metode untuk melakukan pengujian dengan metode *sampling* dalam satuan waktu tertentu. Salah satunya menggunakan alat generator set uji. Generator set uji dibangun untuk melakukan pendeteksian kesalahan pada rangkaian digital kombinasional. Pada penelitian sebelumnya, generator set uji dibuat menggunakan susunan rangkaian elektronika. Setiap perubahan rangkaian uji membutuhkan perubahan rangkaian elektronika pada generator set uji. Sehingga berpengaruh salah satunya pada besarnya *cost* dalam pembuatan sistem. Penelitian ini membangun sebuah generator set uji dengan menggunakan metode *neural network* berbasis *software*. Sehingga apabila input dan output dalam cakupan mikrokontroler setiap perubahan rangkaian uji hanya dilakukan perubahan model *neural network* yang berdampak langsung pada *cost* pembangunan sistem yang lebih hemat. Hasilnya pada penelitian ini telah terbangun sistem generator set uji yang mampu mendeteksi lokasi kesalahan dengan tepat. Akurasi model mencapai 99,95 % membuat hasil prediksi dari model yang dibuat akurat. Selain dapat mendeteksi kesalahan tunggal, rangkaian set uji juga dapat melakukan pendeteksian kesalahan jamak ganda dan tiga lokasi kesalahan yang dipilih dengan akurat.

Kata Kunci: Generator set uji, *neural network*, kesalahan tunggal, kesalahan jamak, rangkaian uji

Judul Tesis

**: GENERATOR SET UJI UNTUK
DIAGNOSIS KESALAHAN RANGKAIAN
KOMBINASIONAL MULTIPLEXER 4 LINE
TO 1 LINE MENGGUNAKAN METODE
ARTIFICIAL NEURAL NETWORK**

Nama Mahasiswa

: Denny Pratama

Nomor Pokok Mahasiswa

: 2225031006

Program Studi

: Magister Teknik Elektro

Fakultas

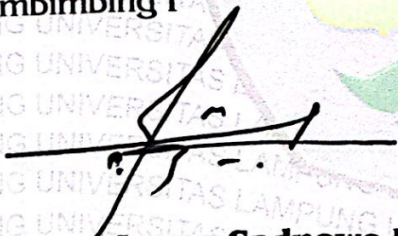
: Teknik

MENYETUJUI

1. Komisi Pembimbing

Pembimbing I

Pembimbing II


Dr. Eng. Ageng Sadnowo R., S.T., M.T.
NIP 19690228 199803 1 001


Dr. Sri Purwiyanti, S.T., M.T.
NIP 19731004 199803 2 001

2. Ketua Program Studi Magister Teknik Elektro

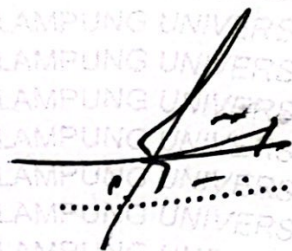

Misfa Susanto, S.T., M.T., Ph.D.
NIP 19710525 199903 1 001

MENGESAHKAN

1. Komisi Penguji

Ketua Komisi Penguji
(Pembimbing I)

: **Dr. Eng. Ageng S. R., S.T., M.T.**



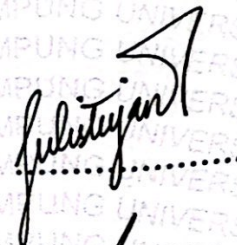
Sekretaris Komisi Penguji
(Pembimbing II)

: **Dr. Sri Purwiyanti, S.T., M.T.**



Anggota Komisi Penguji
(Penguji I)

: **Dr. Ir. Sri Ratna Sulistiyanti, M.T.**



Anggota Komisi Penguji
(Penguji II)

: **Dr. Eng. F.X. Arinto S., S.T., M.T.**



2. Dekan Fakultas Teknik



Dr. Eng. Ir. Helmy Fitriawan, S.T., M.Sc.

NIP 19750928/200112 1 002



Direktur Program Pascasarjana

Prof. Dr. Ir. Murhadi, M.Si.

NIP 19640326 198902 1 001

Tanggal Lulus Ujian Tesis :

LEMBAR PERNYATAAN

Saya menyatakan dengan sesungguhnya bahwa tesis yang saya susun sebagai syarat untuk memperoleh gelar Magister Teknik dari Program Pascasarjana Magister Teknik Elektro seluruhnya merupakan hasil karya sendiri.

Adapun bagian-bagian tertentu dalam penulisan tesis yang saya kutip dari hasil karya orang lain telah dituliskan sumbernya secara jelas sesuai dengan norma, kaidah dan etika penulisan ilmiah.

Tesis dengan judul “Generator Set Uji untuk Diagnosis Kesalahan Rangkaian Kombinasional *Multiplexer 4 Line to 1 Line* Menggunakan Metode *Artificial Neural Network*” dapat diselesaikan berkat bimbingan dan motivasi dari pembimbing-pembimbing saya, yaitu :

1. Dr. Eng. Ageng Sadnowo Repelianto, S.T., M.T.
2. Dr. Sri Purwiyanti, S.T., M.T.

Saya Ucapkan terima kasih yang sebesar besarnya kepada semua pihak, khususnya kedua dosen pembimbing dan Bapak/Ibu Dosen Program Studi Magister Teknik Elektro Universitas Lampung yang banyak memberikan ilmu pengetahuan, bimbingan dan motivasi.

Apabila dikemudian hari ditemukan seluruh atau sebagian tesis ini bukan hasil karya sendiri atau adanya plagiat dalam bagian-bagian tertentu, saya bersedia menerima sanksi pencabutan gelar akademik yang saya sandang dan sanksi-sanksi lainnya sesuai dengan peraturan perundangan yang berlaku.

Bandar Lampung, 6 Agustus 2024



Denny Pratama

NPM. 2225031006

RIWAYAT HIDUP



Penulis dilahirkan di Bandar Lampung, pada tanggal 23 Desember 1995. Penulis merupakan anak ke-1 dari 3 bersaudara, dari pasangan Bapak Nasrudin dan Ibu Siti Hawa.

Penulis pertama kali mengenyam pendidikan di TK Swadhipa Natar. Penulis melanjutkan tingkat sekolah dasar di SD Negeri Bumi Sari lulus tahun 2007. Sekolah Menengah Pertama diselesaikan di SMP Negeri 3 Natar Kec. Natar, lulus tahun 2009. Sekolah Menengah Kejuruan diselesaikan di SMK Negeri 2 Bandar Lampung, lulus tahun 2013. Penulis melanjutkan pendidikan ke jenjang perguruan tinggi di Institut Teknologi Sumatera pada Program Studi Teknik Elektro tahun 2015 melalui jalur Ujian Seleksi Bersama Masuk Perguruan Tinggi Negeri (SBMPTN).

Selama menimba ilmu di Institut Teknologi Sumatera penulis aktif mengikuti organisasi kemahasiswaan internal kampus yakni Himpunan Mahasiswa Teknik Elektro dan Informatika (HMEI) dan Keluarga Mahasiswa Institut Teknologi Sumatera (KM ITERA). Selama berkegiatan organisasi, penulis pernah menyandang sebagai Presiden Mahasiswa ITERA pada tahun 2017/2018.

Pada Tahun 2020, Penulis menyelesaikan program studi S1 Teknik Elektro ITERA dengan menulis sebuah skripsi yang berjudul: “Sistem Rancang Bangun *Smarthome Upgrade System Control and Automatic Xtra* (SUSI CANTIX) Berbasis *Internet of Things*”. Pada November 2020, Penulis mulai bekerja di Institut Teknologi Sumatera sebagai Tenaga Kependidikan pada unit kerja Laboratorium Program Studi

Fisika dibawah UPT Laboratorium Terpadu Institut Teknologi Sumatera.

Pada Tahun 2022 penulis terdaftar sebagai Mahasiswa Program Pascasarjana Magister Teknik Elektro Universitas Lampung. Dan pada tahun 2023 penulis melakukan penelitian pada bidang mikrokontroler dengan judul tesis “Generator Set Uji untuk Diagnosis Kesalahan Rangkaian Kombinasional *Multiplexer 4 Line to 1 Line* Menggunakan Metode *Artificial Neural Network*” dibawah bimbingan Bapak Dr. Eng. Ageng Sadnowo Repelianto, S.T., M.T. dan Ibu Dr. Sri Purwiyanti, S.T., M.T.

Bandar Lampung, 6 Agustus 2024

Penulis

DENNY PRATAMA



PERSEMBAHAN

Dengan Ridho Allah SWT, teriring shalawat kepada Nabi Muhammad SAW

Karya Tulis ini kupersembahkan untuk:

Ayah dan Ibuku Tercinta

Bapak Nasrudin dan Ibu Siti Hawa.

Keluarga Kecil Tersayang

Istriku Kintan Mutiara Putri, S.T., Kedua anakku tersayang Jennaira Alana Dhuha dan Jazlinn Nahda Sabina, Bapak Erna Kumala Hasan, A.Md, Ibu Sumarni, A.Md, Arya Dwi Apriliyanto, S.T, M Fadly Nazwa Putra, Kartika Dwi Wulandari, S.Si dan Ghaitsa Zahira Shofa.

Dosen Teknik Elektro

Yang selalu membimbing, mengajarkan, memberikan saran, baik secara akademis maupun non akademis

Teman- teman kebanggaanku

Rekan – rekan Jurusan Teknik Elektro

Sahabat-sahabatku

Yang selalu membantu, memberikan semangat, mendukung menuju keberhasilan, serta berbagi cerita suka duka dalam berkeluh kesah

Keluarga Besar Magister Teknik Elektro 2022

Yang selalu memberi semangat, dukungan dalam proses yang sangat panjang, dan selalu berdiri bersama dalam perjuangan menuju kesuksesan

Almamaterku

Universitas Lampung

Bangsa dan Negaraku

Republik Indonesia

Terima kasih untuk semua yang telah diberikan kepadaku. *Jazzakallah Khairan.*



MOTTO

*"Jangan Pernah Ingin Kembali Ke Masa Lampau Meskipun
Terlihat Lebih Baik, Hanya Perjuangan dan Pengorbanan
Yang Akan Membangun Diri Dikemudian Hari"*

*"Sebaik Apapun Perkataan, Jika Tidak Ada Aksi, Maka Tidak
Akan ada Artinya. Maka Bangun dan Bergegaslah Merangkai
Mimpi yang Didapat dari Tidur Semalam"*

— Denny Pratama

SAN WACANA

Assalamu'alaikum Wr. Wb.

Syukur Alhamdulillahirobbilalamin, Penulis haturkan puji syukur atas kehadiran Allah SWT yang senantiasa melimpahkan rahmat dan hidayah, serta inayah-Nya kepada penulis sehingga penulis dapat menyelesaikan laporan Tesis dengan mempersembahkan judul tesis “**Generator Set Uji untuk Diagnosis Kesalahan Rangkaian Kombinasional Multiplexer 4 Line to 1 Line Menggunakan Metode Artificial Neural Network**” dengan sebaik-baiknya.

Shalawat beriring salam selalu tercurah kepada junjungan seluruh alam Nabi Muhammad SAW, sahabatnya, serta para pengikutnya yang selalu istiqomah di atas jalan agama islam hingga hari ajal menjemput.

Dalam penyusunan tugas akhir ini penulis banyak mendapat bimbingan, motivasi dan bantuan baik moral maupun materi oleh banyak pihak. Untuk itu dengan sepuh ketulusan hati penulis mengucapkan terima kasih kepada:

1. Ibu Prof. Dr. Ir. Lusmeilia Afriani, D.E.A., I.P.M., selaku Rektor Universitas Lampung.
2. Bapak Prof. Dr. Ir. Murhadi, M.Si., selaku Direktur Program Pascasarjana Fakultas Teknik Universitas Lampung.
3. Bapak Dr. Eng. Helmy Fitriawan, S.T., M.Sc., selaku Dekan Fakultas Teknik, Universitas Lampung.
4. Ibu Herlinawati, S.T., M.T., selaku Ketua Jurusan Teknik Elektro Universitas Lampung.
5. Bapak Misfa Susanto, S.T., M.T., Ph.D. selaku Ketua Program Studi Magister Teknik Elektro Universitas Lampung.
6. Bapak Dr. Eng. Ageng Sadnowo Repelianto, S.T., M.T., selaku dosen pembimbing utama tesis yang banyak memberikan waktu, ide pemikiran dan

semangat serta motivasi bagi penulis.

7. Ibu Dr. Sri Purwiyanti, S.T., M.T., selaku pembimbing kedua tesis, yang telah banyak memberikan waktu, pengalaman, motivasi dan pemikiran bagi penulis.
8. Ibu Dr. Ir. Sri Ratna Sulistiyanti, S.T., M.T., selaku dosen penguji utama sekaligus yang telah banyak memberikan kritik, saran dan motivasi yang bermanfaat bagi penulis.
9. Bapak Dr. Eng. F.X. Arinto Setyawan, S.T., M.T., selaku dosen penguji kedua yang telah banyak memberikan kritik dan saran yang bermanfaat bagi penulis.
10. Seluruh Dosen Program Studi Magister Teknik Elektro Universitas Lampung, berkat ilmu yang telah diajarkan kepada penulis selama penulis menjalani masa studi di perkuliahan.
11. Seluruh Tenaga Pendidik Program Studi Magister Teknik Elektro Fakultas Teknik Universitas Lampung yang telah banyak membantu kepada penulis, sehingga dapat menyelesaikan tugas akhir ini.
12. Seluruh teman-teman Program Studi Magister Teknik Elektro UNILA angkatan 2022 untuk kebersamaan yang telah dijalani. Tiada kata yang dapat penulis utarakan untuk mengungkapkan perasaan senang dan bangga menjadi bagian dari angkatan 2022.
13. Kedua orang tua, ayah dan ibu yang selalu menyayangi, mendidik, membimbing, berkorban serta mendoakan penulis.
14. Istriku Kintan Mutiara Putri yang selalu berdoa, memberi dukungan moral dan materiil mengenai kebutuhan terkait perkuliahan hingga kehidupan agar semua berjalan lancar.
15. Seluruh Dosen Program Studi Fisika Institut Teknologi Sumatera yang telah memberikan kesempatan kepada penulis untuk melanjutkan pendidikan jenjang Magister.
16. Abang Perdana Agung Nugraha, bapak Ardika, mas Taufik Agung Santoso, mas Rosyid, mas Ading Atma Gamilang, mas Okta Ferli Suryadi serta mas Priyan yang membantu penyusunan penelitian penulis.
17. Serta semua pihak yang telah membantu dalam penyusunan tugas akhir ini yang tidak bisa penulis sebutkan satu-persatu.

Akhir kata, Penulis menyadari bahwa Tesis ini masih jauh dari kesempurnaan, baik dari segi isi maupun cara penyajiannya. Oleh karena itu, Penulis sangat mengharapkan saran serta kritik yang bersifat membangun dari pembaca. Akhir kata sedikit harapan penulis semoga karya sederhana ini dapat berguna dan bermanfaat bagi kita semua. Aamiin Allahumma Aamiin.

Wassalamu'alaikum Wr. Wb.

Bandar Lampung, 6 Agustus 2024

Penulis,

Denny Pratama

DAFTAR ISI

	Halaman
HALAMAN JUDUL.....	I
ABSTRACT.....	III
ABSTRAK	IV
LEMBAR PENGESAHAN	VI
RIWAYAT HIDUP.....	VII
SAN WACANA.....	XI
DAFTAR ISI.....	XIV
DAFTAR GAMBAR	XVI
DAFTAR TABEL	XVIII
DAFTAR LAMPIRAN	XIX
BAB I PENDAHULUAN	1
1. 1. Latar Belakang.....	1
1. 2. Rumusan Masalah.....	2
1. 3. Tujuan Penelitian	3
1. 4. Manfaat Penelitian	3
1. 5. Batasan masalah.....	3
1. 6. Hipotesis	4
1. 7. Sistematika Penulisan	4
BAB II TINJAUAN PUSTAKA.....	6
2. 1. Penelitian Terdahulu	6
2. 2. Landasan Teori	10
2. 2. 1. Rangkaian Digital Kombinasional <i>Multiplexer</i>	10
2. 2. 2. Gangguan Tunggal pada Rangkaian Kombinasional.....	11
2. 2. 3. Pohon Diagnosa	14
2. 2. 4. <i>Artificial Neural Network (ANN)</i>	18

2. 2. 5. <i>Keras Framework</i>	23
BAB III METODE PENELITIAN.....	24
3. 1. Waktu dan Tempat Penelitian.....	24
3. 2. Alat dan Komponen Penelitian	24
3. 3. Prosedur Penelitian	24
3. 4. Tahap Pelaksanaan Penelitian.....	25
3. 4. 1. Studi Literatur	25
3. 4. 2. Observasi	25
3. 4. 3. Perancangan Pengujian Generator Set Uji.....	26
3. 4. 4. Penyusunan Dokumen	26
3. 5. Prosedur Pengujian Generator <i>Set Test</i>	26
3. 6. Konsep Perancangan Jaringan Saraf Tiruan	27
3. 7. Susunan Rangkaian Pengujian.....	29
3. 8. Perancangan <i>Generator Set Test</i>	34
3. 9. <i>Experimental Setup</i>	38
3. 10. Diagram Alir Penelitian.....	41
BAB IV HASIL DAN PEMBAHASAN	43
4. 1. Hasil.....	43
4. 1. 1. Perancangan Pohon Diagnosa.....	43
4. 1. 2. Penyusunan <i>Dataset Input Output</i>	53
4. 1. 3. Perancangan Struktur <i>Neural Network</i>	54
4. 1. 4. Implementasi <i>Experimental Setup</i>	65
4. 2. Pengujian Sistem.....	69
4. 2. 1. Pengujian Struktur <i>Neural Network</i>	69
4. 2. 2. Pengujian pada Rangkaian <i>Circuit Under Test</i>	74
BAB V KESIMPULAN DAN SARAN.....	86
5. 1. Kesimpulan	86
5. 2. Saran	86
DAFTAR PUSTAKA	89
LAMPIRAN.....	93

DAFTAR GAMBAR

Gambar	Halaman
Gambar 2. 1. <i>Multiplexer Dual 4 Line to 1 Line</i>	10
Gambar 2. 2. Hubungan Antara Gangguan, Kesalahan dan Kegagalan.	12
Gambar 2. 3. Karakteristik Gangguan.....	13
Gambar 2. 4. Gangguan <i>Stuck Logika</i> (a) <i>Stuck at 0</i> dan (b) <i>Stuck at 1</i>	13
Gambar 2. 5. Hasil Pengujian Lengkap.	16
Gambar 2. 6. Pemilihan Pengujian F^*	17
Gambar 2. 7. (a) Pengujian $F^*_{1(3)}$ dan (b) Pengujian $F^*_{1(3)}$	17
Gambar 2. 8. (a) Pengujian $F^*_{10(1)}$ dan (b) Pengujian $F^*_{11(1)}$	17
Gambar 2. 9. Pohon Diagnosa <i>AND Gate</i>	18
Gambar 2. 10.(a) Jaringan Saraf Manusia, (b) Manipulasi Teknologi Jaringan Saraf.	20
Gambar 3. 1. Perbandingan Perbedaan <i>Generator Set Test</i> (a) Menggunakan Rangkaian Digital Elektronika, (b) Menggunakan <i>Neural Network</i>	27
Gambar 3. 2. <i>Generator Set Uji Menggunakan Neural Network</i>	28
Gambar 3. 3. (a) <i>Input Output Generator Set Uji</i> , (b) Representasi <i>Input Output GST</i> pada <i>Neural Network</i>	28
Gambar 3. 4. Rangkaian <i>multiplexer dual 4 line to 1 line IC 74153</i>	30
Gambar 3. 5. Potongan 1 Rangkaian <i>Multiplexer Dual 4 Line to 1 Line IC 74153</i>	31
Gambar 3. 6. Kondisi <i>Sa1</i> dan <i>Sa0</i>	32
Gambar 3. 7. Diagram Pengkabelan <i>CUT Multiplexer Dual 4 Line to 1 Line IC74153</i>	33
Gambar 3. 8. <i>PCB CUT Multiplexer Dual 4 Line to 1 Line IC 74153</i>	34
Gambar 3. 9. Diagram Pengkabelan Catu Daya.	35
Gambar 3. 10. Diagram Pengkabelan Mikrokontroler <i>Arduino Mega</i>	36
Gambar 3. 11 Tampak Atas <i>Generator Set Uji</i>	38

Gambar 3. 12. Diagram <i>Experimental Setup</i>	39
Gambar 3. 13. <i>Experiment Setup</i> Lengkap.....	40
Gambar 3. 14. Diagram Alir Penelitian	41
Gambar 3. 15. Diagram Alir Pengujian Pohon Diagnosa.	42
Gambar 3. 16. Diagram Alir Pengujian <i>Neural Network</i>	42
Gambar 4. 1. Pengujian Level 1 Pemilihan Pengujian F^*	48
Gambar 4. 2. Pengujian Level 2 (a) Pengujian $F^*(112)_0$ dan (b) Pengujian $F^*(112)_1$	49
Gambar 4. 3. Pengujian Level 3 (a) Pengujian $F^*(75)_{00}$ dan (b) Pengujian $F^*(75)_{01}$	50
Gambar 4. 4. Pengujian level 3 (a) Pengujian $F^*(90)_{10}$ dan (b) Pengujian $F^*(90)_{11}$	51
Gambar 4. 5. Pohon Diagnosa Rangkaian Digital Kombinasional IC 74153.	52
Gambar 4. 6. <i>Dataset Input Output</i>	55
Gambar 4. 7. Grafik Perbandingan Antara Perulangan <i>Dataset</i> Terhadap Persentase Prediksi.....	59
Gambar 4. 8. Grafik Perbandingan Antara Jumlah <i>Hidden Layer</i> dan <i>Neuron</i> Terhadap Persentase Prediksi.....	63
Gambar 4. 9. Struktur <i>Neural Network</i> yang Dibangun.	65
Gambar 4. 10. Rangkaian Generator Set Uji.....	66
Gambar 4. 11. Rangkaian <i>Circuit Under Test</i>	68
Gambar 4. 12. <i>Flowchart</i> Pengujian Sistem.	69
Gambar 4. 13. Hasil <i>Output</i> Prediksi dari Model <i>Neural Network</i>	70
Gambar 4. 14 Grafik <i>Threshold</i>	71
Gambar 4. 15. <i>Output</i> Setelah Diberi Nilai <i>Threshold</i>	72
Gambar 4. 16. Hasil Evaluasi Perbandingan Kesamaan pada Kolom.	72
Gambar 4. 17. Persentase Kesamaan.	74
Gambar 4. 18. Titik Uji Kesalahan Tunggal.	75
Gambar 4. 19. Lokasi kesalahan 1 $Sa0$ dan 9 $Sa1$	80
Gambar 4. 20. Lokasi Kesalahan 10 $Sa1$ dan 23 $Sa0$	81
Gambar 4. 21. Lokasi Kesalahan 5 $Sa1$ dan 25 $Sa1$	82

DAFTAR TABEL

Tabel	Halaman
Tabel 2. 1. <i>Output</i> Gerbang <i>AND</i> pada Kondisi Normal dan Saat Kondisi <i>Stuck</i>	14
Tabel 2. 2. Kelompok Kesalahan	14
Tabel 2. 3. Operasi <i>XOR</i> pada Pengujian {1, 2, 3}	15
Tabel 3. 1. Prosedur Pengujian Generator <i>Set Test</i>	26
Tabel 3. 2. Tabel Kebenaran Pohon Diagnosa	29
Tabel 3. 3. Tabel Kebenaran <i>IC 74153</i>	31
Tabel 3. 4. Koneksi Pin Mikrokontroler	37
Tabel 4. 1. Tabel Kesalahan Rangkaian Digital Kombinasional <i>IC 74153</i>	44
Tabel 4. 2. Kelompok Kesalahan Rangkaian Digital Kombinasional <i>IC 74153</i> .	46
Tabel 4. 3. Pengujian <i>XOR</i>	47
Tabel 4. 4. Persamaan <i>Input Output</i> Pohon Diagnosa <i>IC 74153</i>	53
Tabel 4. 5. <i>Dataset Input Output</i>	54
Tabel 4. 6. Parameter Awal Model <i>Neural Network</i>	55
Tabel 4. 7. Jumlah <i>Neuron</i> pada Masing-Masing <i>Hidden Layer</i>	64
Tabel 4. 8. Tabel Pengujian Kesalahan Tunggal.....	76
Tabel 4. 9. Deteksi <i>Error</i> Kelompok 1 dan 2.....	81
Tabel 4. 10. Deteksi <i>Error</i> Kelompok 2 dan 3.....	82
Tabel 4. 11. Deteksi <i>Error</i> Kelompok 3 dan 1.....	83
Tabel 4. 12. Deteksi <i>Error</i> Kelompok Kesalahan 1, 2 dan 3	83
Tabel 4. 13. Deteksi <i>Error</i> Kelompok Kesalahan 2 pada 3 Kesalahan.....	84

DAFTAR LAMPIRAN

Lampiran	Halaman
Lampiran 1. Prosedur Lengkap Pembuatan Pohon Diagnosa.....	93
Lampiran 2. Struktur Lengkap <i>Neural Network</i>	121
Lampiran 3. Persamaan <i>Input Output</i> Pohon Diagnosa <i>IC 74153</i>	127
Lampiran 4. <i>Dataset Input Output</i>	129
Lampiran 5. <i>Souce Code Generator Set Test</i>	131
Lampiran 6. <i>Experimet Setup</i>	138

BAB I PENDAHULUAN

1. 1. Latar Belakang

Pengujian produksi masal komponen *Integrated Circuit* atau sering disebut dengan IC membutuhkan komponen pengujian yang efektif. Pengujian kesalahan pada produksi adalah proses diagnosis untuk memastikan bahwa hasil produksi memiliki kualitas yang terjamin sesuai standar persentase yang telah ditetapkan oleh perusahaan. Diagnosis kesalahan merupakan topik penelitian yang sedang dikembangkan beberapa dekade terakhir. Bagian ini merupakan hal penting untuk menjamin kualitas dan keamanan dari suatu produk[1]. Berbagai kasus keadaan abnormal seperti kejadian alam, kecelakaan fisik, kegagalan peralatan dan kesalahan operasi merupakan keadaan yang sulit dihindarkan[2]. Oleh sebab itu diperlukan perangkat pengujian untuk mempermudah aktifitas sampling produk.

Berbagai tahapan dapat dilakukan untuk menjaga sistem agar tetap terjaga dari kerusakan seperti proses pengawasan dan proses perawatan, tetapi sering kali muncul kerusakan yang disebabkan oleh keadaan abnormal sehingga perlu dilakukan proses deteksi dari kesalahan[2], [3]. Cara yang paling sederhana dan banyak digunakan dalam pendeteksian penentuan titik kesalahan adalah membuat *bank data* atau catatan kerusakan yang pernah terjadi untuk dapat disesuaikan apabila terjadi kesalahan yang sama di kemudian hari. Selain membuat *bank data*, dapat juga dilakukan dengan melakukan pemodelan dari sistem sehingga dari model yang dibuat dapat mencakup kesalahan yang timbul[3]. Seiring perkembangan zaman, proses pendeteksian sering kali dikaitkan dengan sinyal yang keluar dan dibandingkan dengan keadaan sebelumnya. Identifikasi dan penemuan titik kesalahan sering juga disebut dengan diagnosis kesalahan[4]. Adapun tugas yang harus dilakukan untuk diagnosis kesalahan adalah mengetahui jenis kesalahan seperti ukuran kesalahan, lokasi dan waktu pendeteksian[5].

Diagnosis merupakan hal penting dalam penentuan kerusakan. Tahapan ini membutuhkan pengembangan teknologi supaya proses dapat dilakukan dengan waktu yang cepat dan waktu yang efektif, salah satunya dengan melibatkan *Artificial Intelligence* atau kecerdasan buatan. Kecerdasan buatan dapat melakukan prediksi *output* dengan melakukan beberapa tahapan pengerjaan[6]. *Artificial Neural Network* (ANN) atau Jaringan Saraf Tiruan merupakan salah satu bagian dari *Artificial Intelligence* yang saat ini banyak digunakan manfaatnya dalam penentuan prediksi *output*[6]. Tujuan utama dari penggunaan jaringan saraf tiruan untuk mengembangkan algoritma matematika yang berjalan berdasarkan sistem kerja otak manusia[7]. Jaringan saraf tiruan bersifat empiris, tetapi dapat memberikan solusi praktis yang akurat untuk sebuah permasalahan pada suatu model[7]. Rangkaian digital kombinasional merupakan rangkaian yang disusun minimal menggunakan dua tahap sederhana. Penggabungan dua tahap sederhana ini terdiri dari beberapa gerbang logika yang disusun menjadi satu kesatuan[8]. Rangkaian logika berpotensi memiliki kesalahan atau kegagalan yang perlu dilakukan diagnosis. Teknik diagnosis kesalahan pada sebuah rangkaian dapat diklasifikasikan menjadi teknik sebab akibat[9].

Penelitian ini memanfaatkan *neural network* guna membangun sebuah generator set uji dari suatu pohon diagnosa pengujian yang digunakan untuk melakukan proses diagnosis kesalahan dengan menjalankan tahapan pengujian pohon diagnosa pada rangkaian digital kombinasional. Pemanfaatan jaringan saraf tiruan digunakan sebagai pengganti perangkat elektronik manual sehingga lebih efisien terhadap *hardware* dalam sistem yang dibangun. Sistem akan tetap dapat gunakan selama *input* dan *output* yang digunakan masih dalam cakupan, pengguna hanya melakukan *update* nilai bobot untuk dapat menggunakan sistemnya kembali. Sehingga dengan sistem yang dibangun, dapat melakukan penghematan waktu pengujian serta menurunkan penggunaan biaya yang dilakukan untuk melakukan pembuatan generator set uji.

1. 2. Rumusan Masalah

Rumusan masalah yang diangkat pada penelitian ini adalah

1. Bagaimana tahapan menentukan pohon diagnosa yang digunakan dalam mencari lokasi kesalahan?

2. Bagaimana membangun struktur *neural network* yang sesuai dengan fungsi set uji rangkaian kombinasional *multiplexer 4 line to 1 line*?
3. Bagaimana membangun generator set uji menggunakan jaringan saraf tiruan untuk uji rangkaian digital kombinasional?

1. 3. Tujuan Penelitian

Tujuan yang akan dicapai pada penelitian ini adalah

1. Membangun sebuah generator set uji yang dapat bekerja sesuai dengan pohon diagnosa pengujian.
2. Melakukan implementasi pohon diagnosa pengujian menggunakan struktur jaringan saraf tiruan.
3. Melakukan analisis efisiensi waktu pengujian dari kerja generator set uji dengan cara melakukan perbandingan dengan pengujian yang dilakukan dengan cara manual.

1. 4. Manfaat Penelitian

Penggunaan metode *neural network* pada pengujian rangkaian digital kombinasional mampu meningkatkan efisiensi penggunaan komponen *hardware* pada rangkaian set uji yang akan berpengaruh terhadap biaya pembangunan sistem.

1. 5. Batasan masalah

Adapun batasan terhadap masalah pada penelitian ini adalah

1. Simulasi gangguan secara tunggal adalah gangguan tunggal dan bersifat tetap atau gangguan permanen pada rangkaian kombinasional multiplexer 4 line to 1 line.
2. Perangkat rangkaian digital kombinasional yang digunakan adalah IC SN 74153.
3. Generator set uji yang dibangun menggunakan mikrokontroler *Arduino Mega*.
4. Struktur *neural network* dibangun menggunakan bahasa pemrograman *Python* dengan menggunakan *package Keras* sebagai *tools* untuk membangun struktur *neural network* dengan variasi parameter *dataset input output*, jumlah *neuron* dan jumlah *hidden layer*.

1. 6. Hipotesis

Terbangunnya sistem diagnosis kesalahan menggunakan pohon diagnosa untuk mendeteksi kesalahan dan lokasi kesalahan pada rangkaian digital kombinasional *multiplexer 4 line to 1 line IC SN 74153*. Melalui struktur yang dibangun, dapat diketahui tingkat akurasi prediksi pada rangkaian set uji. Dapat dilakukan analisis kinerja terhadap rangkaian pengujian.

1. 7. Sistematika Penulisan

Supaya penulis dan pembaca dapat memahami isi dari karya tulis ini, terdapat sistematika penulisan sebagai berikut.

BAB I PENDAHULUAN

Bab ini berisi latar belakang penelitian mengenai diagnosa kesalahan serta alasan penelitian ini dapat dikembangkan, perumusan masalah, tujuan yang ingin dicapai pada penelitian seperti terbangunnya sebuah generator set uji hingga dapat dilakukan analisis terhadap kinerjanya, batasan masalah terhadap penelitian serta yang menjadi sebuah manfaat apabila penelitian ini dilakukan yaitu pengurangan komponen pada generator set uji untuk meningkatkan efisiensi penggunaan *hardware*.

BAB II TINJAUAN PUSTAKA

Bab ini menjelaskan mengenai dasar teori rangkaian digital kombinasional, proses terjadinya gangguan pada sebuah rangkaian digital kombinasional, model gangguan, informasi mengenai tabel kesalahan, pohon diagnosa dan *neural network* juga dilampirkan. Bab ini juga dilengkapi dengan penelitian terdahulu yang berasal dari sumber karya tulis ilmiah mengenai pembahasan rangkaian digital kombinasional, perkembangan *neural network*, klasifikasi dan prosedur penentuan struktur *neural network* yang berguna dalam penyusunan penelitian.

BAB III METODE PENELITIAN

Bab ini menjelaskan mengenai waktu dan tempat pelaksanaan penelitian, alat dan komponen yang digunakan dalam pembuatan sistem diagnosis kesalahan rangkaian digital kombinasional *multiplexer dual 4 line to 1 line IC SN 74153*, penggunaan *neural network* sebagai generator set uji untuk menjalankan perintah pengujian dengan prinsip kerja pohon diagnosa. *Neural network* digunakan sebagai pengganti

hardware agar lebih efisien. Selain itu, bab ini juga membahas mengenai struktur *neural network* dibangun serta langkah pengujian sistem generator set uji.

BAB IV HASIL DAN PEMBAHASAN

Bab ini menjelaskan mengenai generator set yang dibangun pada rangkaian digital kombinasional *multiplexer dual 4 line to 1 line IC SN 74153* dapat bekerja sesuai dengan pohon diagnosa, dilakukan pengujian terhadap kecepatan proses klasifikasi kesalahan, dilakukan pengujian *double* gangguan untuk melihat respon generator set uji terhadap pengujian di luar kapasitas pengujian serta melakukan pengujian seberapa besar tingkat akurasi sistem yang dibangun dalam melakukan prediksi.

BAB V KESIMPULAN

Pohon diagnosa yang terbangun dalam pembuatan sistem akan diuji antara pohon diagnosa dan *neural network* yang dibangun. Selanjutnya, dilakukan pengujian tingkat akurasi prediksi untuk melihat seberapa besar tingkat kepercayaan sistem yang dibangun sehingga dapat digunakan. Selain dari kesimpulan yang menjawab tujuan, terdapat saran yang disertakan untuk keberlanjutan penelitian ke depan.

BAB II

TINJAUAN PUSTAKA

2. 1. Penelitian Terdahulu

Sebelum melakukan penelitian, dibutuhkan peninjauan terhadap penelitian yang pernah dilakukan sebelumnya. Penelitian sebelumnya digunakan sebagai sarana pencarian referensi seputar topik permasalahan yang akan dibahas pada penelitian ini yaitu diagnosis kesalahan pada rangkaian kombinasional. Adapun beberapa penelitian terdahulu yang digunakan sebagai referensi utama dalam menulis adalah sebagai berikut :

Pertama, penelitian yang dilakukan oleh Ageng Sadnowo Repelianto pada tahun 1996 dengan judul Penggunaan Metoda Tabel Kesalahan untuk Diagnosis Kesalahan Rangkaian Digital Kombinasional. Luaran penelitian tersebut adalah untuk melakukan deteksi kesalahan yang ada dan mengetahui titik letak kerusakan dengan menggunakan metoda tabel kesalahan dengan objek potongan IC SN 74153. Menggunakan pohon diagnosa yang terbangun dari tabel kesalahan dapat melakukan diagnosis kesalahan pada rangkaian dengan mendeteksi ada atau tidaknya kesalahan. Apabila terdapat kesalahan, dapat ditunjukkan letak lokasi kesalahan berada. Pada penelitian ini dilakukan pengujian sebanyak 10 tingkat pengujian untuk mendeteksi tanpa kesalahan, hasilnya terdapat 4096 variasi pengujian dengan tingkat efisiensi pengujian sebesar 99,76%. Pohon diagnosa dapat mempercepat melakukan diagnosis kesalahan yang terjadi pada sebuah rangkaian. Sistem mendeteksi rata-rata kesalahan pada tingkat efisiensi sebesar 47,42 % untuk kesalahan tunggal[10].

Kedua, penelitian yang dilakukan oleh Rizkima Akbar Setiawan pada tahun 2023 dengan judul penelitian Rancang Bangun Generator Set Uji untuk Diagnosis Kesalahan Rangkaian *Multiplexer Quad 2 Line To 1 Line* IC 74157 Menggunakan Metode Tabel Kesalahan Berbasis *Arduino Nano Atmega328*. Penelitian ini

membuat generator set uji untuk melakukan diagnosis kesalahan tunggal yang terjadi pada sebuah rangkaian logika kombinasional yang dilakukan secara otomatis. Generator set akan melakukan identifikasi terhadap kesalahan, kemudian menentukan pada titik mana lokasi kesalahan berada. IC yang digunakan pada penelitian ini adalah potongan IC 75157. Hasilnya, penelitian ini memiliki tingkat efisiensi sebesar 68,75% hingga 81,25% dengan panjang eksperimen berjumlah 5 tingkat. Efisiensi yang didapatkan ketika melakukan pengujian lengkap pada potongan IC 75157 menghasilkan tingkat efisiensi waktu dalam proses pengujian sebesar 98,05% sampai 98,83% dengan pengujian terpanjang 20 level untuk mendeteksi tanpa kesalahan[11].

Ketiga, Penelitian yang dilakukan oleh Seongmin Heo dan Jay H. Lee yang dipublikasikan pada tahun 2018 dengan judul *Fault Detection and Classification Using Artificial Neural Networks*. Penelitian ini melakukan proses diagnosis kesalahan pada sistem proses *Tennessee Eastman*. Melakukan deteksi kesalahan, serta mencari lokasi kesalahan menggunakan jaringan saraf tiruan merupakan hal yang sedang dikembangkan. Penelitian melakukan proses klasifikasi dengan melakukan perbandingan jumlah *layer* yang digunakan dalam proses diagnosis kesalahan. Dilakukan perbandingan tingkat akurasi diagnosis pada 2, 3 dan 4 *layer* dengan struktur yang berbeda. Hasilnya menunjukkan semakin banyak *layer* dan *neuron* maka tingkat akurasi semakin tinggi karena jumlah data yang dilakukan *training* meningkat sehingga hasil semakin akurat. penambahan jumlah *layer* dan *neuron* berpengaruh kepada tingkat akurasi hingga menunjukkan sekitar 97,26% pada level tertentu. Hasil menunjukkan akurasi yang baik dan menjanjikan, ada beberapa hal yang perlu dilakukan pada penelitian selanjutnya. Pertama, karakteristik kesalahan perlu dianalisis secara mendalam untuk meningkatkan proses diagnosis. Kedua, dampak augmentasi perlu ditinjau lebih lanjut terhadap perubahan akurasi. Terakhir, perlu dilakukan pengujian terhadap pemilihan jenis jaringan saraf untuk melakukan deteksi kesalahan dan klasifikasi[12].

Keempat, penelitian yang dilakukan oleh Muhammad Uzair dan Noreen Jamil pada tahun 2020 dengan judul *Effects of Hidden Layers on the Efficiency of Neural Networks*. Penelitian ini membahas mengenai pengaruh dari *hidden layer* dan jumlah *neuron* yang digunakan dalam sebuah sistem. Penelitian dilakukan

dengan cara membandingkan penelitian yang menggunakan jaringan saraf terhadap dua parameter yaitu, kompleksitas waktu dan tingkat akurasi. Kompleksitas waktu merupakan berapa banyak waktu yang digunakan jaringan saraf untuk mempelajari keseluruhan masalah kemudian menghasilkan dalam rentang waktu yang diperlukan. Biasanya sistem akan berjalan selaras dengan jumlah *hidden layer* dan *neuron*-nya. Semakin banyak jumlah *hidden layer* dan *neuron*, maka akan semakin membutuhkan waktu yang banyak untuk menghasilkan *output*. Selanjutnya tingkat akurasi, faktor ini akan mempengaruhi tingkat kebenaran dalam melakukan penelitian. Jaringan yang memiliki lapisan tersembunyi akan memiliki tingkat akurasi yang cenderung tinggi. Akan tetapi perlu dipertimbangkan berapa level maksimal penggunaan *hidden layer* dan *neuron* yang digunakan supaya sistem dapat optimal berjalan. Penelitian ini membandingkan 10 penelitian terkait untuk dianalisis tingkat kompleksitas waktu dan tingkat akurasi yang digunakan. Hasilnya menunjukkan bahwa penentuan jumlah *hidden layer* dan *neuron* perlu diperhitungkan secara matang karena akan berpengaruh kepada tingkat kompleksitas waktu dan akurasi. Proses lama pelatihan akan dipengaruhi langsung pada jumlah *hidden layer* dan *neuron*. Selain parameter tersebut, perlu diperhitungkan *overfitting* dan *underfitting* yang terjadi pada sistem. Sistem akan berjalan bergantung pada analisis dari peneliti sehingga perkiraan jumlah *hidden layer* dan *neuron* tidak dapat ditentukan pada sebuah parameter tertentu[13].

Kelima, penelitian yang dilakukan oleh Oludare Isaac Abiodun dkk dengan judul *State-of-the-art in Artificial Neural Network Applications: A survey* yang di publish pada tahun 2018. Jurnal survei ini menjelaskan secara terperinci mengenai aplikasi yang dapat dikembangkan menggunakan jaringan saraf tiruan. Banyak kelompok keilmuan yang dapat dikembangkan pada jaringan saraf termasuk pada bidang komputasi dan teknologi. Pada penelitian ini dijelaskan mengenai penggunaan dan *feedforward* dan *feedback propagation* akan berpengaruh pada akurasi, kecepatan pemrosesan, latensi, toleransi kesalahan, volume dan kinerja yang dilakukan. Hasilnya adalah banyak model dari jaringan saraf yang ditampilkan pada penggunaan sehari-hari di berbagai bidang yang dibahas pada penelitian ini meliputi keamanan komputer, ilmu kedokteran, bisnis, keuangan, bank, asuransi, pasar saham, pembangkit listrik, manajemen, industri nuklir, eksplorasi mineral,

pertambangan, prediksi kualitas fraksi minyak mentah, prediksi hasil panen, pengolahan air, dan kebijakan. Besarnya cakupan yang dapat menggunakan jaringan saraf membuat proses pengembangan dapat beradaptasi dengan baik. Pengembangan ANN terletak pada karakteristik pemrosesan informasi hingga kekuatan pembelajaran, paralelisme tinggi, toleransi kesalahan, nonlinier, toleransi terhadap derau yang rendah, dan kemampuan generalisasi[14].

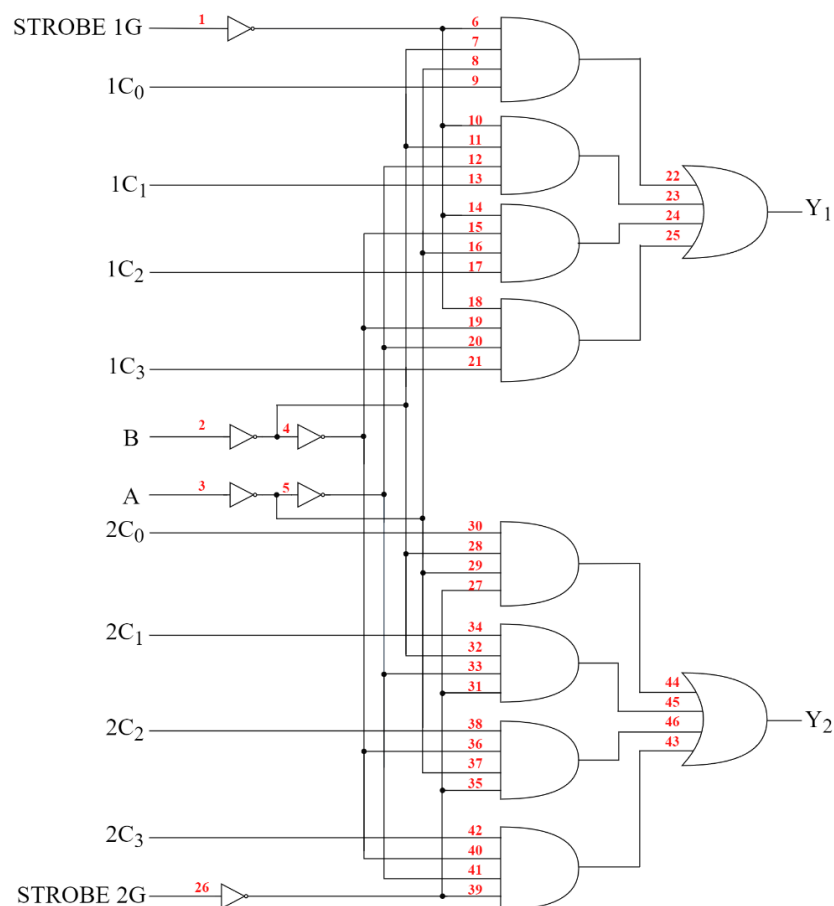
Keenam, penelitian yang dilakukan oleh Yudhi Andrian dan M. Rhifky Wayahdi pada tahun 2014 dengan judul Analisis Perbandingan Metode *Perceptron* dan *Backpropagation* dalam Mengenali Gerbang Logika. Penelitian ini membandingkan *neural network* menggunakan *back propagation* dan *perceptron*. Implementasi yang dilakukan pada gerbang logika *AND*, *OR*, *NOR*, *NAND*, *Ex-NOR*, *Ex-OR*. Penelitian membandingkan dua metode *neural network* tersebut untuk melakukan *training* data membaca *output* pada gerbang logika. Hasilnya, pada *back propagation* dapat mengenali pola data *input* dan *output* dengan baik, sedangkan *perceptron* kurang baik dalam mengenali pola *input* dan target biner terutama pada gerbang *Ex-NOR* dan *Ex-OR*. Metode *perceptron* mengenali pola lebih cepat dibandingkan *back propagation* khususnya dalam fungsi *OR* dan *AND* dengan jumlah iterasi 20. Metode *perceptron* dan *back propagation* tidak dapat mengenali gerbang logika dengan kondisi *input* bipolar target bipolar maupun data *input* biner dengan target bipolar (*hybrid*)[15].

Secara garis besar, penelitian mengenai diagnosis kesalahan pada rangkaian digital kombinasional sudah dilakukan. Akan tetapi, dengan menambahkan kecerdasan buatan dalam sistem diagnosis kesalahan merupakan salah satu kebaruan dalam penelitian yang dilakukan. Jaringan saraf yang digunakan sebagai metode yang dilakukan pada diagnosis kesalahan rangkaian kombinasional. Penggunaan jaringan syaraf tiruan pada diagnosis kesalahan dapat menghemat *cost* dalam penggunaan komponen karena tidak perlu mengganti komponen elektronik apabila rangkaian digital kombinasional masih dalam cakupan *input* dan *output* yang sama.

2. 2. Landasan Teori

2. 2. 1. Rangkaian Digital Kombinasional *Multiplexer*

Rangkaian digital kombinasional merupakan rangkaian gabungan antar gerbang logika dengan target nilai *output* yang dipengaruhi dengan nilai *input* yang diberikan. Rangkaian logika kombinasional akan bekerja sesuai dengan prinsip dasar gerbang logika dan bernilai *output* sesuai dengan masukan yang diberikan pada waktu tersebut dikarenakan sebuah rangkaian kombinasional tidak memiliki memori untuk menyimpan informasi[14]. Prinsip kerja dari sebuah rangkaian kombinasional dapat diilustrasikan pada sebuah persamaan aljabar *boolean*. Aljabar *boolean* merupakan salah satu fungsi yang memiliki nilai penjabarannya di ibaratkan hanya pada dua kondisi yaitu benar atau *true* atau bernilai 1 dan salah atau *false* atau bernilai 0[16]. Salah satu aplikasi pada rangkaian digital kombinasional adalah rangkaian *multiplexer*. Berikut pada Gambar 2. 1 merupakan *multiplexer 4 line to 1 line*.

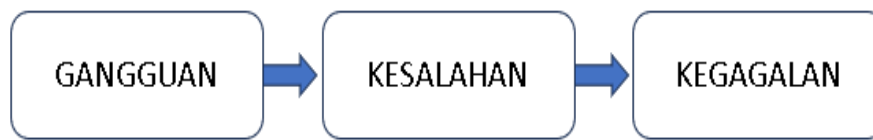


Gambar 2. 1. *Multiplexer Dual 4 Line to 1 Line*.

Prinsip kerja dari *multiplexer* adalah memperkecil jumlah keluaran menjadi satu jalur yang berasal dari masukkan lebih dari satu jalur yang dikendalikan oleh *select*. *Select* akan memilih informasi mana yang akan diteruskan menuju keluaran[17][18]. *Multiplexer* yang digunakan pada penelitian ini adalah *multiplexer 4 line to 1 line* dimana *multiplexer* memiliki masukan empat jalur dengan keluaran satu jalur. *Multiplexer* yang disajikan pada Gambar 2. 1 merupakan *multiplexer* dengan identitas IC 74153 dimana terdapat dua rangkaian identik yang menyusun rangkaian digital kombinasional *multiplexer* tersebut. Jumlah masukan dalam IC tersebut adalah delapan, dengan jumlah keluaran sebanyak 2 akan tetapi jika dilakukan pemotongan maka dapat di bagi menjadi memiliki 4 masukan dan 1 keluaran. IC *multiplexer* ini memiliki manfaat dibidang kontrol sebagai saklar dan dapat juga digunakan sebagai cara untuk melakukan efisiensi dalam mengirimkan data. Pemilihan salah satu jalur yang diteruskan pada sisi output dapat berisikan banyak informasi didalamnya seperti kondisi, pengkodean serta identitas yang sudah ditentukan[17], [18].

2. 2. 2. Gangguan Tunggal pada Rangkaian Kombinasional

Gangguan merupakan suatu kerusakan dalam bentuk fisik. Kerusakan dapat terjadi pada perangkat keras maupun lunak pada *Integrated Circuit*. Gangguan merupakan salah satu bentuk dari sebuah kesalahan. Kondisi ini tentu akan mempengaruhi keadaan sebenarnya dalam suatu sistem. Berdasarkan perancangan toleransi gangguan, gangguan terbagi menjadi tiga bagian yang saling berkaitan yaitu gangguan, kesalahan dan kegagalan[10], [11]. Bagian ini memiliki hubungan keterkaitan satu sama lain. Gangguan adalah penyebab terjadinya kesalahan, kesalahan adalah penyebab terjadinya kegagalan kerja sistem pada sebuah sistem. Apabila hasil yang muncul tidak sesuai dengan keadaan normal, maka terdapat kegagalan yang disebabkan oleh gangguan dari sistem tersebut. Sehingga kegagalan bukan adalah sifat asli, melainkan hanya keputusan terakhir yang muncul dari sistem. Berikut adalah keterkaitan gangguan, kesalahan dan kegagalan.

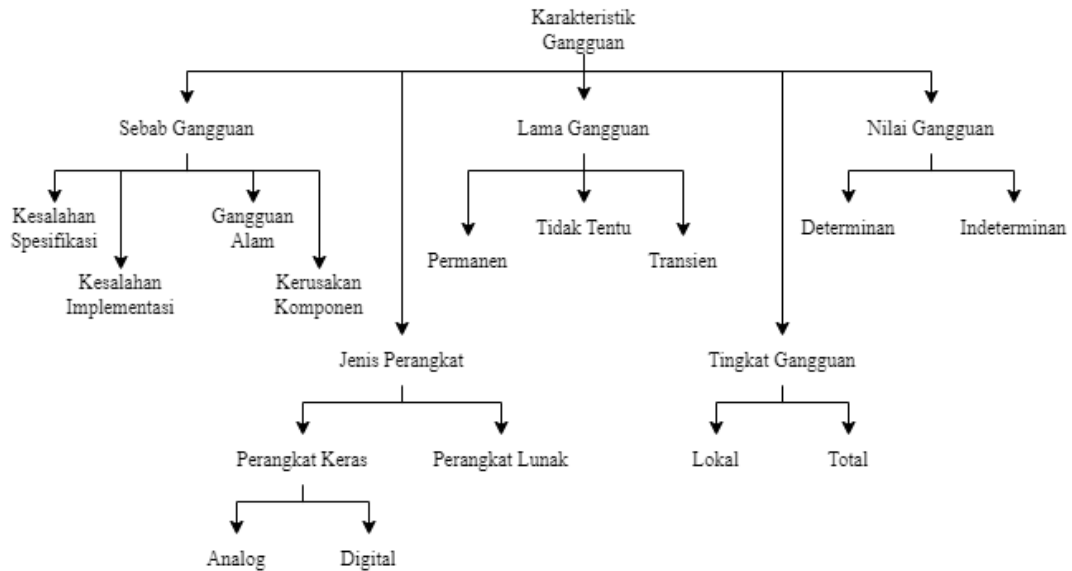


Gambar 2. 2. Hubungan Antara Gangguan, Kesalahan dan Kegagalan.

Gambar 2. 2 adalah ilustrasi kegagalan yang penyebab mendasarnya adalah gangguan. Sumber gangguan yang dapat menyebabkan kegagalan bersumber dari kesalahan spesifikasi, implementasi, komponen dan faktor luar atau faktor alam. Hal pertama yang dapat menyebabkan terjadinya gangguan adalah kesalahan spesifikasi. Kesalahan spesifikasi dapat menyebabkan ketidakcocokan algoritma yang akan berpengaruh pada perangkat keras dan perangkat lunak sistem. Penyebab kedua adalah implementasi, hal ini sering terjadi pada proses koneksi jika menggunakan banyak komponen berdekatan sehingga sering tertukar atau *short* karena tertempel antar komponen. Selain pada perangkat keras, perangkat lunak juga dapat menjadi faktor penyebab kesalahan dari implementasi seperti proses pembuatan *code* yang tidak sesuai sehingga fungsi tidak dapat dijalankan dengan semestinya. Selanjutnya kesalahan ketiga adalah komponen, biasanya terjadi karena sistem pabrikasi yang kurang baik sehingga komponen bekerja di luar ambang toleransi yang seharusnya. Sehingga tidak bisa dijalankan sesuai dengan target utamanya. Yang terakhir adalah kesalahan yang terjadi karena gangguan luar, gangguan luar dapat dipengaruhi oleh cuaca, suhu, kelembaban, medan elektromagnetik, kegagalan operasi hingga bencana alam.

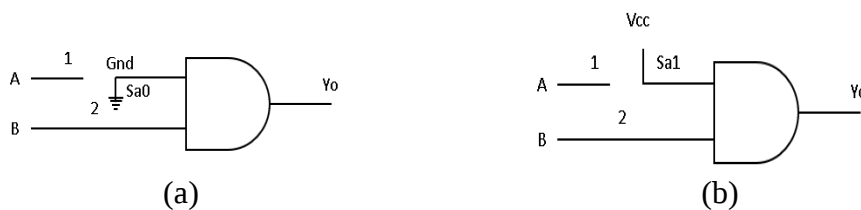
Karakteristik gangguan dalam sebuah perangkat dapat dibedakan menjadi beberapa bagian, salah satunya adalah lamanya gangguan terjadi. Gangguan permanen adalah salah satu jenis gangguan durasi, sehingga apabila tidak dilakukan perbaikan pada bagian yang rusak maka sistem tidak akan bekerja secara normal kembali. Akibat dari gangguan permanen yaitu sistem kerja dari sebuah sistem tidak bekerja dengan baik sehingga perangkat tidak dapat digunakan. Salah satu contoh terjadinya kerusakan permanen adalah rusaknya jalur atau sebuah gerbang logika pada suatu rangkaian kombinasional yang disebabkan oleh kerusakan fisik atau komponen pada perangkat elektronika itu sendiri. Selanjutnya adalah gangguan transien, biasanya gangguan ini disebabkan oleh adanya petir yang

terjadi. Petir dapat mengganggu sistem selama kurun waktu tertentu sehingga ketika petir itu tidak datang, maka keadaan akan berjalan normal. Terakhir, gangguan tidak menentu yaitu gangguan yang datang dan menghilang dalam kurun waktu yang tidak menentu. Gambar 2.3 merupakan jenis dan karakteristik gangguan.



Gambar 2. 3. Karakteristik Gangguan.

Model gangguan *stuck* logika merupakan salah satu model yang dapat digunakan dalam proses pencarian titik kesalahan karena sangat efektif dan sederhana. Model gangguan ini masuk kepada gangguan permanen apabila tidak diperbaiki maka akan selalu dalam keadaan abnormal. Model gangguan seperti ini dapat dikategorikan pada dua kondisi yaitu gangguan *stuck at 0* ($Sa0$) dan gangguan *stuck at 1* ($Sa1$). Berikut merupakan konsep dasar mengenai model gangguan $Sa0$ dan $Sa1$.



Gambar 2. 4. Gangguan *Stuck* Logika (a) *Stuck at 0* dan (b) *Stuck at 1*.

Gambar 2. 4 (a) menunjukkan kondisi $Sa0$ setiap *input* yang diberikan pada *input* A akan bernilai 0 sehingga akan mempengaruhi *output* yang diberikan pada gerbang AND tersebut. Gambar 2. 4 (b) menunjukkan kondisi $Sa1$ setiap *input* yang

diberikan pada *input* A akan bernilai 1. Berikut adalah tabel kebenaran dari dua keadaan pada Gambar 2.4.

Tabel 2. 1. *Output* Gerbang *AND* pada Kondisi Normal dan Saat Kondisi *Stuck*

Uji	A	B	y_0	y_{10}	y_{11}	y_{20}	y_{21}
0	0	0	0	0	0	0	0
1	0	1	0	0	1	0	1
2	1	0	0	0	0	0	0
3	1	1	1	0	1	0	1

Kondisi pada Tabel 2.1. merupakan tabel keseluruhan yang muncul ketika diberikan kondisi *stuck* di setiap jalur. Hasil respon model gangguan *stuck* logika ini disebabkan oleh gangguan fisik yang menyebabkan kerusakan permanen. Secara khusus diasumsikan bahwa gangguan keadaan *stuck* merupakan gangguan tunggal atau hanya terjadi satu keadaan dalam satu pengujian. Total kerusakan yang berpotensi dari setiap n jalur memiliki 2^n potensi kerusakan jalur secara tunggal.

Hasil y_0 merupakan hasil keadaan normal dari gerbang *AND*, sedangkan y_{10} , y_{11} , y_{20} , y_{21} secara berturut adalah keluaran ketika dalam keadaan gangguan *Sa0* dan *Sa1* pada jalur yang memiliki potensi gangguan. Terdapat dua *output* yang memiliki kondisi yang sama sehingga dikategorikan dalam kelompok kesalahan yang sama. Kesalahan dikelompokkan apabila terdapat hasil yang identik sama dari setiap percobaan kondisi *stuck*. Terdapat 4 kelompok kesalahan pada gerbang *AND* yang diilustrasikan yaitu f_0 adalah *output* tanpa kesalahan, f_1 adalah *output* yang kerusakan *Sa0* pada jalur 1 dan *Sa0* pada jalur 2, f_2 adalah *output* kerusakan *Sa1* pada jalur 1, dan f_3 adalah kerusakan *Sa1* pada jalur 2. Berikut adalah Tabel 2.2 yang merupakan tabel kelompok kesalahan pada gerbang *AND*.

Tabel 2. 2. Kelompok Kesalahan

Uji	A	B	$f_0 = \{y_0\}$	$f_1 = \{y_{10}, y_{20}\}$	$f_2 = \{y_{11}\}$	$f_3 = \{y_{21}\}$
0	0	0	0	0	0	0
1	0	1	0	0	1	0
2	1	0	0	0	0	1
3	1	1	1	0	1	1

2. 2. 3. Pohon Diagnosa

Pohon diagnosa adalah sebuah grafik petunjuk yang memiliki simpul berupa pengujian. Terdapat dua hasil dalam setiap pengujian yang menyatakan proses

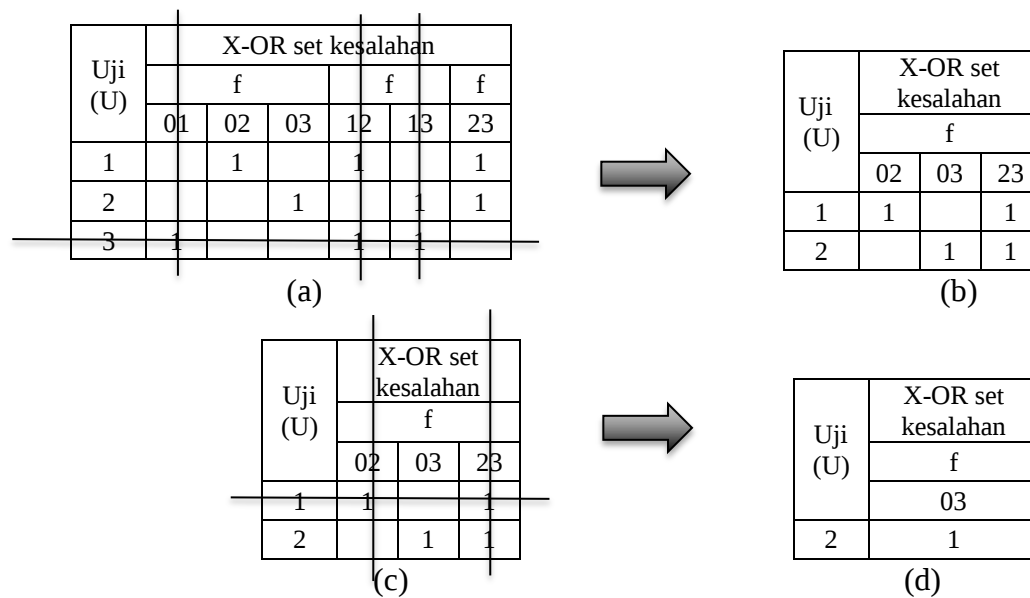
pengujian akan berlanjut atau berhenti. Pengujian akan berlanjut pada pengujian selanjutnya apabila hasil masih dalam cakupan tertentu, sedangkan pengujian akan berhenti ketika pengujian menunjukkan sebuah lokasi kesalahan[4]. Selanjutnya ditentukan set pengujian lengkap minimum dengan menguji masing-masing kelompok kesalahan dengan cara melakukan operasi *XOR gate*. Kemudian mencari nilai selisih terkecil dari setiap baris dan melakukan pengelompokan nilai yang seragam dan melakukan eliminasi pada tahapan selanjutnya[10]. Berikut contoh tahapan pembuatan pohon diagnosa pada gerbang *AND*.

Setelah menjadi sebuah kelompok kesalahan, selanjutnya dilakukan pengoperasian menggunakan gerbang *XOR* untuk mengetahui set pengujian lengkap. Baris pengujian (0) tidak dilakukan pengoperasian *XOR* karena *redundant* atau memiliki nilai yang sama. sehingga dilakukan hanya pada pengujian {1, 2, 3}. berikut pada Tabel 2.3 disajikan hasil pengoperasian set pengujian lengkap gerbang *AND* dengan melakukan pengujian *XOR*.

Tabel 2. 3. Operasi *XOR* pada Pengujian {1, 2, 3}

Uji (U)	A	B	X-OR set kesalahan					
			f			f		f
			01	02	03	12	13	23
0	0	0						
1	0	1		1		1		1
2	1	0			1		1	1
3	1	1	1			1	1	

Setelah mendapatkan hasil pengujian awal, dilakukan eliminasi dengan kriteria identik dengan baris lain yang dominan 1, tetapi tidak ada pasangan identik dalam kolom tersebut dan dilakukan proses eliminasi terhadap baris dan kolom. Pada Tabel 2.3 terlihat pada pengujian $f0 \otimes f1$ terdapat identik yang dominan pada horizontal, tetapi pada vertikal hanya tunggal nilai 1 sehingga baris dan kolom tersebut dapat dilakukan eliminasi sehingga yang menjadi pengujian pertama adalah 3.



Gambar 2. 5. Hasil Pengujian Lengkap.

Gambar 2.5 (a) merupakan eliminasi pertama pada pengujian 3. pengujian 3 dipilih karena masuk pada kriteria yaitu terdapat baris identik 1 pada kolom tersebut sehingga pengujian dipilih menjadi urutan pertama yang menghasilkan Gambar 2.5 (b). Setelah dilakukan eliminasi pertama, selanjutnya dilakukan eliminasi kembali pada Gambar 2.5 (c) adalah proses eliminasi baris identik pada pengujian ke 1 dengan hasil eliminasi pada Gambar 2.5 (d). Secara berturut yang menjadi urutan pengujian pada pohon diagnosa adalah pengujian {3, 1, 2}. Pohon diagnosa disusun berdasarkan urutan pengujian dengan melakukan pencarian nilai selisih terkecil antara nilai 0 dan 1 pada tabel dengan Persamaan 2.1 berikut.

$$Ri = |wi_0 - wi_1| \quad (2.1)$$

Keterangan :

Ri : Selisih antara jumlah 0 dan 1 dalam baris

wi_0 : jumlah dari angka 0 dalam baris

wi_1 : jumlah dari angka 1 dalam baris

Setelah mendapatkan urutan pengujian lengkap, tahapan selanjutnya adalah membuat pohon diagnosis dengan melakukan proses penjadwalan pengujian dengan level minimal yang telah dirancang pada penentuan set pengujian lengkap. Kemudian dipilih pengujian awal dengan jumlah Ri terkecil menggunakan Persamaan 2. 1. Gambar 2. 6 baris pengujian (3) dipilih karena pada baris awal dan

jumlah R_i yang sama dengan baris lain sehingga pemilihan bisa dilakukan pada salah satu baris.

$F^* =$					
f_0	f_1	f_2	f_3	U	R_i
1	0	1	1	3	2
0	0	1	0	1	2
0	0	0	1	2	2

Gambar 2. 6. Pemilihan Pengujian F^* .

Keterangan :

U : Urutan pengujian

R_i : Selisih antara nilai 0 dan 1

Pengujian (3) menghasilkan pengujian pada level 1 yaitu $F^*_{0(3)}$ dan $F^*_{1(3)}$. $F^*_{0(3)}$ menghasilkan sebuah lokasi kesalahan yaitu f_1 dan $F^*_{1(3)}$ dipilih pengujian selanjutnya yaitu pengujian (1) dengan selisih jumlah 0 dan 1 ($R_i = 1$)

$F^*_{0(3)} =$	
f_1	U
0	1
0	2

(a)

$F^*_{1(3)} =$				
f_0	f_2	f_3	U	R_i
0	1	0	1	1
0	0	1	2	1

(b)

Gambar 2. 7. (a) Pengujian $F^*_{1(3)}$ dan (b) Pengujian $F^*_{1(3)}$.

Berdasarkan Gambar 2.7 (a) menunjukkan lokasi kesalahan sedangkan yaitu pada f_1 (b) proses pengujian selanjutnya yaitu pengujian (1) dengan $R_i = 1$.

$F^*_{10(1)} =$			
f_0	f_3	U	R_i
0	1	2	0

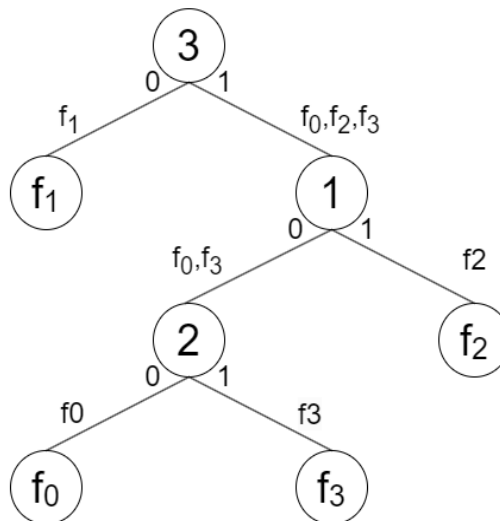
(a)

$F^*_{11(1)} =$	
f_2	U
1	1
0	2

(b)

Gambar 2. 8. (a) Pengujian $F^*_{10(1)}$ dan (b) Pengujian $F^*_{11(1)}$.

Gambar 2. 8 (a) menunjukkan nilai pengujian selanjutnya adalah pengujian (2) dengan nilai 0 adalah f_0 dan nilai 1 adalah f_3 . Sedangkan Gambar 2. 8(b) menunjukkan lokasi kesalahan f_2 . Secara berturut, akan menghasilkan sebuah pohon diagnosa seperti disajikan pada Gambar 2.9 berikut.



Gambar 2. 9. Pohon Diagnosa *AND Gate*.

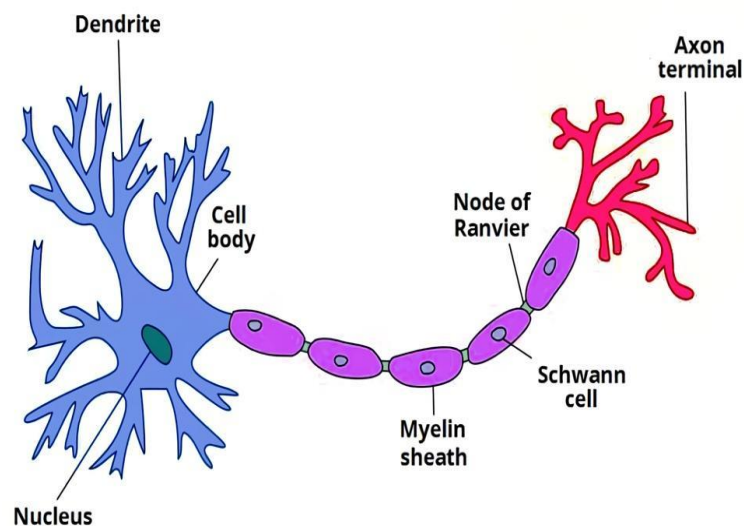
Pohon diagnosa *AND gate* memiliki 3 level pengujian dengan penentuan lokasi kesalahan sesuai dengan kelompok kesalahan yang dibuat pada kelompok kesalahan. f_0 atau tanpa kesalahan atau menandakan pengujian normal harus berada di pengujian terakhir karena setiap kerusakan akan terlebih dahulu di deteksi secara keseluruhan. Selain itu terdapat ketentuan bahwa f_0 tidak boleh memiliki nilai *output* yang sama dengan kelompok kesalahan lain.

2. 2. 4. Artificial Neural Network (ANN)

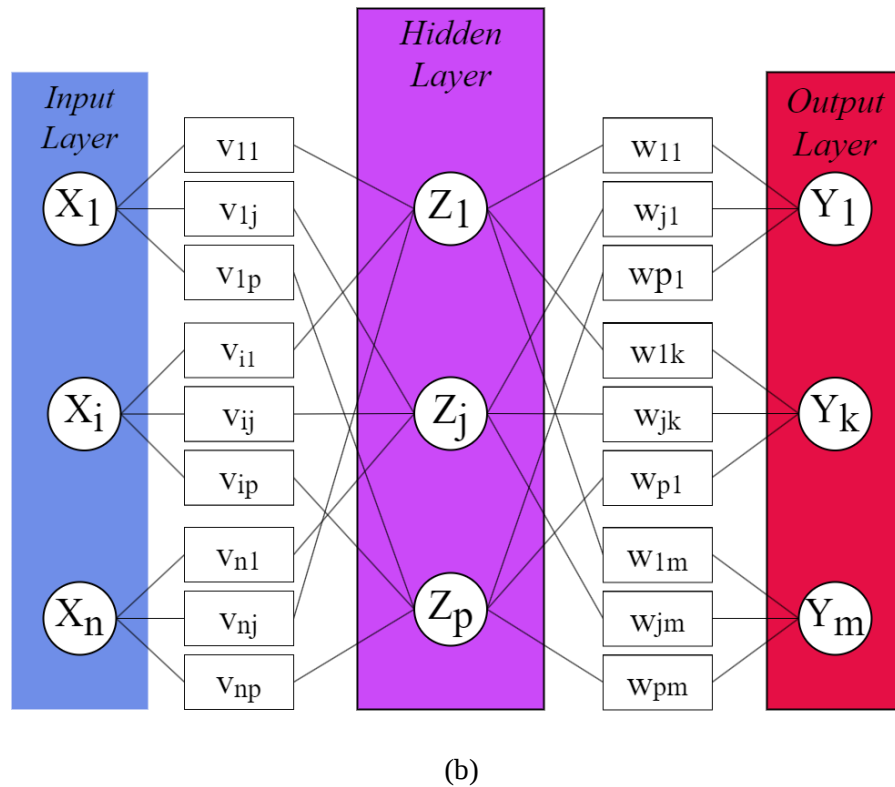
Jaringan saraf tiruan atau sering disebut dengan ANN adalah sebuah paradigma pemrosesan informasi yang didasari oleh jaringan saraf pada sistem biologis manusia[13]. Pemanfaatannya digunakan pada permodelan klasifikasi, pengelompokan, pengenalan pola dan prediksi membuat jaringan saraf semakin berkembang di banyak cabang keilmuan seperti bidang pertanian, sains, ilmu, dan pendidikan [19]. ANN terbukti menjadi model yang sangat baru yang diterapkan pada pemecahan masalah dan pembelajaran mesin karena model yang dibangun dapat menyesuaikan dengan kebutuhan sistem yang dibangun. Adapun komponen di dalam sebuah jaringan saraf terdiri dari *input*, *weight* dan *output*. *Input* merupakan nilai masukan yang digunakan saat pembelajaran maupun saat pengenalan suatu objek. *Weight* atau bobot merupakan beban yang direpresentasikan pada suatu nilai yang selalu berubah nilainya setiap kali diberikan

input pada saat proses pembelajaran berlangsung. *Output* merupakan luaran hasil dari pengenalan suatu objek.

Input layer atau lapisan *input* akan berinteraksi dengan lingkungan luar yang dikategorikan dalam sebuah nilai *input* yang diberikan. Kemudian informasi mulai diproses oleh *hidden layer* untuk didapatkan parameternya. Setelah mendapatkan nilai parameter yang sesuai, maka dapat ditampilkan pada *output layer* atau lapisan *output*. Pemilihan kriteria lapisan tersembunyi merupakan tugas yang sangat sulit karena dalam beberapa kondisi akan menjadi sebuah nilai *overfitting* dan *underfitting*. Pada pemilihan metode jaringan saraf, penentuan jumlah *layer* perlu diperhatikan dengan melakukan *testing* dan membaca banyak literatur untuk mendapatkan tingkat akurasi yang baik pada lapisan keluaran. Kondisi *overfitting* terjadi apabila jumlah lapisan tersembunyi menjadi sangat besar dibandingkan dengan kompleksitas yang membuat dampak buruk terhadap efisiensi pengujian sistem. Artinya model tidak dapat beradaptasi untuk menebak variabel baru yang akan dilakukan pengujian. Kondisi *underfitting* terjadi ketika jumlah lapisan tersembunyi dalam jaringan menjadi kurang dari kompleksitas masalah karena jaringan hampir tidak dapat menangani masalah sehingga tidak dapat menebak model dengan tepat. Kondisi ini disebut dengan *undertraining* atau kekurangan pelatihan. Pada dasarnya setiap pengujian *neural network* perlu dilakukan dengan banyak percobaan sehingga model dapat dijalankan sesuai dengan ketentuan yang diharapkan dari setiap penelitian [13].



(a)



Gambar 2. 10.(a) Jaringan Saraf Manusia, (b) Manipulasi Teknologi Jaringan Saraf.

Penelitian ini menggunakan jaringan saraf tiruan dengan metode *back propagation*. *Back propagation* dipilih dikarenakan pada metode ini terdapat proses evaluasi atau perbaikan nilai bobot penyusun yang dilakukan dengan melakukan tahapan propagasi mundur. Tahapan pelatihan suatu model ANN menggunakan *back propagation* terdiri dari tiga fase yaitu propagasi maju (*feedforward*), propagasi mundur (*backpropagation*) dan modifikasi bobot. fase propagasi maju merupakan suatu tahapan perhitungan keseluruhan mulai dari *layer* masukan hingga *layer* keluaran dengan fungsi aktivasi yang ditentukan. Fase propagasi mundur merupakan perbaikan nilai nilai *error* dari bobot untuk diperkecil mulai dari *output* hingga *input*. Fase modifikasi bobot merupakan tahapan penyesuaian nilai bobot baru dengan cara menghitung selisih *error* nilai propagasi mundur dengan nilai bobot lama.

Tahapan proses penyelesaian menggunakan algoritma *backpropagation* seperti pada Gambar 2. 10 (b) dengan fungsi aktivasi *relu* sebagai berikut.

1. Inisiasi semua nilai bobot dengan sebuah bilangan

2. Jika kondisi keluaran belum terpenuhi, sistem akan melakukan langkah 3—10 .
3. Setiap data pelatihan akan melakukan langkah 4—9 yang terdiri dari propagasi maju, propagasi mundur dan koreksi bobot

Fase 1 Propagasi Maju

4. Masing-masing unit masukan akan menerima sinyal yang akan diteruskan ke unit tersembunyi (*hidden layer*).
5. Hitung komponen keluaran pada unit tersembunyi kemudian dilakukan aktivasi menggunakan Persamaan 2. 2 dan Persamaan 2. 3.

$$z_{net\ j} = v_{0j} + \sum_{i=1}^p x_i \cdot v_{ij} \quad (2. 2)$$

Keterangan:

$z_{net\ j}$: Net input untuk neuron j

v_{0j} : bias yang digunakan

x_i : nilai input dari i

v_{ij} : bobot penghubung input i menuju neuron j

$$z_j = f(z_{net\ j}) = \max (0, z_{net\ j}) \quad (2. 3)$$

Keterangan:

z_j : Output aktivasi neuron j

$\max (0, z_{net\ j})$: fungsi aktivasi *relu*, ketika nilai output negatif bernilai 0 dan ketika output bernilai positif maka nilai tersebut menjadi output.

6. Hitung komponen keluaran jaringan di unit Y_k menggunakan Persamaan 2. 4 dan di aktivasi menggunakan Persamaan 2. 5

$$Y_{net\ k} = w_{0j} + \sum_{j=1}^m z_j \cdot w_{jk} \quad (2. 4)$$

Keterangan:

$Y_{net\ k}$: Net output untuk neuron k

w_{0j} : bias yang digunakan pada sisi output

z_j : nilai input dari neuron j (neuron sebelumnya)

w_{jk} : bobot penghubung input j menuju output

$$y_k = f(Y_{net\ k}) = \max (0, Y_{net\ k}) \quad (2. 5)$$

Keterangan:

Y_k : output aktivasi neuron k

$\max(0, z_{netj})$: fungsi aktivasi *relu*

Fase 2 Propagasi Mundur

7. Hitung faktor δ unit keluaran berdasarkan kesalahan di setiap unit keluaran Y_k menggunakan Persamaan 2. 6

$$\delta_k = (t_k - y_k)f'(Y_{net\ k}) = (t_k - y_k) y_k (1 - y_k) \quad (2. 6)$$

Keterangan:

δ_k : Gradien kesalahan k pada sisi *output*

t_k : Target *output neuron k*

y_k : Output aktual *neuron k*

$Y_{net\ k}$: Net input untuk *neuron k* dilapisan *output*

8. Hitung faktor δ unit tersembunyi berdasarkan perbaikan kesalahan di setiap unit tersembunyi Z_j menggunakan Persamaan 2. 7.

$$\delta_{net\ j} = \sum_{k=1}^m \delta_k \cdot w_{jk} \quad (2. 7)$$

Keterangan:

$\delta_{net\ j}$: Gradien kesalahan pada neuron j pada sisi *hidden layer*

δ_k : Gradien kesalahan *neuron k* pada lapisan *output*

w_{jk} : Bobot penghubung antara *neuron j* ke lapisan *output k*

Faktor δ *hidden layer*

$$\delta_j = \delta_{net\ j} f'(Z_{net\ j}) = \delta_{net\ j} z_j (1 - z_j) \quad (2.8)$$

Hitung nilai suku perubahan bobot (yang akan dipakai untuk merubah v_{ij})

$$\Delta V_{ij} = \alpha \delta_j x_i \quad (2.9)$$

Keterangan:

δ_j : Gradien kesalahan pada neuron j selama proses *backpropagation*

ΔV_{ij} : Gradien kesalahan pada neuron j pada sisi *hidden layer*

α : *Learning rate*

Fase 3 Modifikasi Bobot

9. Menghitung keseluruhan nilai perubahan bobot

Perubahan nilai bobot yang menuju ke unit keluaran

$$W_{jk}(\text{baru}) = w_{jk}(\text{lama}) + \Delta w_{jk} \quad (k = 1, 2, \dots, m ; j = 0, 1, 2, \dots, p)$$

Perubahan nilai bobot yang menuju ke unit tersembunyi

$$v_{ij}(\text{baru}) = v_{ij}(\text{lama}) + \Delta v_{ij} \quad (j = 1, 2, \dots, p ; i = 0, 1, 2, \dots, n)$$

10. Berhenti

2. 2. 5. Keras Framework

Keras merupakan sebuah perangkat *Application Programming Interface (API)* dimana merupakan sekumpulan program runtutan yang dibuat untuk memudahkan proses pembuatan jaringan *neural network*. Implementasi *Keras* dapat menggunakan bahasa pemrograman *python*. Selain *Keras*, pada saat membuat sebuah pemrograman *neural network* digunakan sebuah *package* bantuan seperti *pandas* dan *TensorFlow* [20]. *Pandas* merupakan *package* perangkat lunak menggunakan bahasa pemrograman *python* untuk memanipulasi serta melakukan analisa terhadap data yang diberikan. Implementasi penggunaan *pandas* pada sistem *neural network* adalah penggabungan, pemisahan, pembersihan dan transformasi data. Data yang diterima dalam bentuk *CSV*, *Excel*, *SQL* dan *JSON*. Selain *pandas*, digunakan juga *package seaborn* yang digunakan untuk pengolahan tampilan seperti grafik, *plotting*, *scatter*, *bar plots* serta *heatmaps*. Selain itu, yang digunakan untuk pengolahan dan pengembangan program adalah *TensorFlow*. *TensorFlow* merupakan sebuah kumpulan pustaka *opensource* yang dikembangkan oleh *google* untuk memfasilitasi pembangunan *neural network*.

Keras akan mengatur segala kebutuhan pembangunan *neural network* seperti *dataset*, jumlah *neuron*, lapisan *hidden layer* yang digunakan serta metode apa yang digunakan dalam membangun *neural network*. Dengan dibantu tools lain seperti *pandas*, *seaborn* dan *TensorFlow* dapat dibangun sebuah model dan dilakukan visualisasi sehingga dapat digunakan dan dikembangkan sesuai dengan kebutuhan. Implementasi *neural network* menggunakan kelompok *package* ini memudahkan pengguna untuk melakukan perhitungan, pemodelan serta melakukan analisa dan evaluasi terhadap suatu model.

BAB III

METODE PENELITIAN

3. 1. Waktu dan Tempat Penelitian

Perencanaan, implementasi dan pengujian penelitian ini dilaksanakan di Laboratorium Elektronika dan Kendali Teknik Elektro, Universitas Lampung.

3. 2. Alat dan Komponen Penelitian

Guna menyelesaikan penelitian dengan baik, perlu melengkapi alat dan komponen menjadi satu kesatuan dengan baik. Alat dan komponen yang digunakan sebagai berikut.

1. 1 unit *PC core i3* sesuai dengan spesifikasi pemrograman
2. 1 unit Mikrokontroler *Arduino Mega Pro Mini 2560* sebagai perangkat utama pengolahan data.
3. 2 buah *project board* untuk melakukan proses perancangan .
4. 2 buah IC 4072 yang digunakan sebagai gerbang *OR*.
5. 4 buah IC 7404 yang digunakan sebagai gerbang *NOT*.
6. 2 buah IC 4082 yang digunakan sebagai gerbang *AND*
7. 1 paket kabel *jumper* untuk melakukan koneksi antar perangkat.
8. *Jumper set* untuk peubah kondisi.
9. Multimeter *digital* sebagai alat ukur utama.
10. *Charging modul* TP4056.
11. Baterai *lithium ion 18650*.
12. Modul *step down XL 6009*.
13. *Voltage regulator LM 7805*.

3. 3. Prosedur Penelitian

Mekanisme yang dilakukan dalam penelitian ini adalah *Research and Development* atau disebut dengan (*R&D*). Penelitian ini merupakan penelitian lanjutan yang dikembangkan dengan metode baru dari penelitian yang pernah

dilakukan. Penelitian sebelumnya melakukan pendeteksian kesalahan pada rangkaian kombinasional menggunakan metode tabel kesalahan berbasis *Arduino nano ATMEGA328*. Penelitian tersebut melakukan diagnosis kesalahan pada rangkaian *IC 75157* dengan tingkat efisiensi waktu 68,75% sampai dengan 82,25%. Pengembangan yang dilakukan pada penelitian ini adalah menggunakan *neural network* untuk menjalankan fungsi pohon diagnosa.

3. 4. Tahap Pelaksanaan Penelitian

Penelitian dilakukan dengan mengacu terhadap beberapa tahapan yang dilakukan guna menyelesaikan penelitian dengan hasil yang baik. Adapun beberapa tahapan antara lain.

3. 4. 1. Studi Literatur

Satu tahapan penting yang dilakukan pada awal penelitian adalah studi literatur. Studi literatur merupakan tahapan pencarian sumber terkait penelitian sehingga dapat diketahui posisi penelitian yang akan dilakukan. Posisi penelitian dikaitkan dengan kebaruan terhadap penelitian yang dijalankan. Selain itu studi literatur juga digunakan untuk mencari referensi terkait penelitian yang pernah dilakukan. Penelitian ini menggunakan kebaruan dengan menambahkan jaringan saraf yang digunakan dalam melakukan diagnosis kesalahan. Sehingga, proses pencarian literasi meliputi beberapa objek antara lain, diagnosis kesalahan, diagnosis kesalahan menggunakan *neural network*, penentuan jumlah *neuron* dan *layer* pada *neural network* melalui hasil publikasi karya tulis penelitian.

3. 4. 2. Observasi

Setelah melakukan studi literatur, selanjutnya melakukan observasi. Observasi dilakukan dengan melakukan pencocokan terhadap hasil atau isi dari karya tulis yang dijadikan referensi untuk selanjutnya dikembangkan menjadi model penelitian. Tahap ini dimulai dengan melakukan pembuatan tabel kesalahan dan pohon diagnosa hingga menentukan model yang digunakan pada jaringan saraf yang digunakan.

3. 4. 3. Perancangan Pengujian Generator Set Uji

Pengujian rangkaian digital kombinasional diperlukan untuk mencapai target dan toleransi yang dapat dibandingkan dengan penelitian lain. Pengujian ini meliputi tiga bagian utama yaitu pengujian fungsional, pengujian kesalahan dan pengujian paramater. Pengujian fungsional merupakan rangkaian pengujian yang dilakukan untuk membuktikan bahwa sebuah rangkaian kombinasional memiliki kerusakan yang sama dengan tabel kerusakan yang telah direncanakan. Pengujian kesalahan merupakan pengujian yang dilakukan untuk melakukan deteksi kerusakan yang terjadi serta dapat menunjukkan lokasi terjadinya kerusakan. Pengujian dilakukan dengan cara menguji seberapa cepat proses klasifikasi sistem berjalan untuk menemukan lokasi kesalahan, kemudian dilakukan pengujian *double* gangguan untuk melihat bagaimana respon dapat berjalan.

3. 4. 4. Penyusunan Dokumen

Dokumen disusun untuk memuat hasil analisis pengujian diagnosis kesalahan dengan metode tabel kesalahan menggunakan *neural network* yang dilakukan secara komprehensif. Dokumen disusun sesuai dengan pengujian masing-masing komponen pengujian.

3. 5. Prosedur Pengujian Generator Set Test

Prosedur pengujian merupakan runtutan pengujian dalam melakukan diagnosis kesalahan pada *AND gate* pada Gambar 2. 9. Tahapan ini dilakukan untuk melakukan pengecekan lokasi kesalahan berada melalui tahapan pengujian yang mengacu pada pohon diagnosa. Tabel 3.1 menunjukkan jumlah tingkatan pengujian yang dilakukan serta menunjukkan dimana lokasi kesalahan berada. Tabel 3. 1 tersebut juga menunjukkan tingkatan pengujian yang dilakukan yang digunakan untuk melakukan validasi terhadap percobaan implementasi yang dilakukan.

Tabel 3. 1. Prosedur Pengujian Generator Set Test

Lokasi Kesalahan	Pengujian	Tingkat Pengujian			Keterangan
		1	2	3	
f1 Jalur 1 (Sa 0) Jalur 2 (Sa 0)	U PD	3 0			Terdeteksi f1
f2 Jalur 1 (Sa 1)	U PD	3 1	1 1		Terdeteksi f2

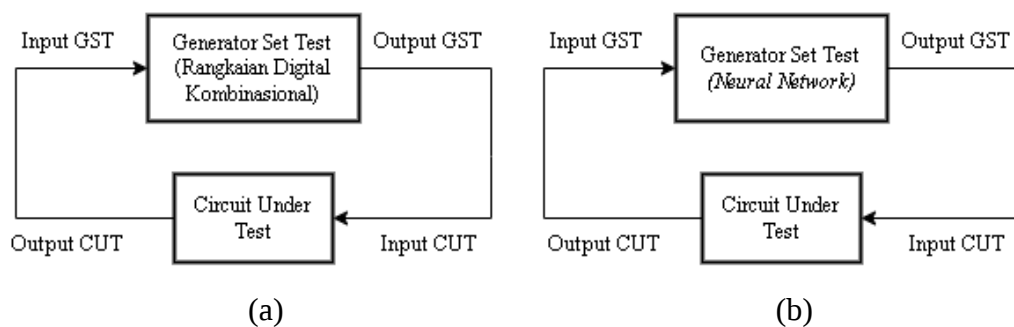
Lokasi Kesalahan	Pengujian	Tingkat Pengujian			Keterangan
		1	2	3	
f3	U	3	1	2	Terdeteksi f3
Jalur 2 (Sa 1)	PD	1	0	1	
f0	U	3	1	2	Tidak ada kesalahan
Tidak ada kesalahan	PD	1	0	0	

Keterangan :

U : Pengujian, PD : Pohon Diagnosa

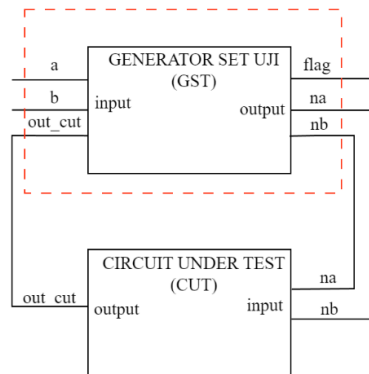
3. 6. Konsep Perancangan Jaringan Saraf Tiruan

Generator set test (GST) atau generator set uji yang dibangun berdasarkan prinsip kerja dari pohon diagnosa dapat dibangun menggunakan rangkaian elektronik digital kombinasional dan dibangun menggunakan *neural network*. Pembangunan generator set uji berbasis rangkaian elektronik digital hanya dapat bekerja pada prinsip pohon diagnosa *IC circuit under test (CUT)* yang dilakukan pengujian. Artinya, Ketika berganti *CUT* yang digunakan, maka pohon diagnosa akan berganti dan generator set uji akan diganti dengan generator set uji yang baru. Sehingga akan terjadi pemborosan biaya pembangunan generator set uji pengujian. Berbeda dengan *GST* yang dibangun menggunakan rangkaian elektronik, generator set uji yang dibangun menggunakan *neural network* hanya melakukan perubahan bobot apabila berubah *IC* dalam *CUT* selama *input* dan *output* dalam cakupan generator set uji. Penggunaan *neural network* dalam Pembangunan generator set uji akan melakukan penghematan dalam biaya Pembangunan sistemnya. Selanjutnya pada Gambar 3. 1 berikut disajikan ilustrasi perbedaan yang terletak pada penggunaan rangkaian digital kombinasional dengan *neural network* pada Pembangunan sistem generator set uji.



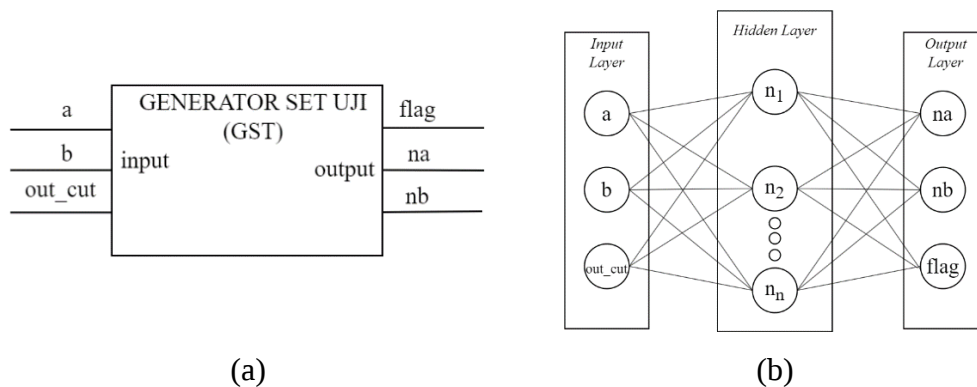
Gambar 3. 1. Perbandingan Perbedaan *Generator Set Test* (a) Menggunakan Rangkaian Digital Elektronika, (b) Menggunakan *Neural Network*.

Pada dasarnya, tidak ada perbedaan prinsip kerja antara generator set uji yang dibangun menggunakan rangkaian digital kombinasional dengan *neural network*. Sehingga *neural network* bisa dijadikan solusi dalam pembangunan generator set uji dikarenakan terdapat penghematan biaya dalam pembangunan sistem. Tahapan yang dilakukan dalam pembangunan sistem yang pertama adalah melakukan pemotongan pada sisi generator set uji yang dibangun menggunakan *neural network* seperti disajikan pada Gambar 3. 2. yang mengacu pada prinsip kerja pohon diagnosa gerbang and seperti disajikan pada Gambar 2. 9.



Gambar 3. 2. Generator Set Uji Menggunakan *Neural Network*.

Sisi *input* dari generator set uji adalah *a*, *b* dan *out_cut*. Nilai *a* dan *b* merupakan masukan pengujian yang dilakukan sedangkan *out_cut* adalah keluaran natural dari *circuit under test*. Selanjutnya pada sisi *output* terdapat *na*, *nb* dan *flag*, *na* dan *nb* merupakan pengujian lanjutan dan *flag* merupakan kondisi apakah selanjutnya berupa pengujian atau merujuk pada sebuah lokasi kesalahan. Jika dilakukan pemotongan sisi generator set uji maka seperti disajikan pada Gambar 3. 3 (a).



Gambar 3. 3. (a) *Input Output* Generator Set Uji, (b) Representasi *Input Output* GST pada *Neural Network*.

Sisi *input* dan *output* dari generator set uji merujuk pada sebuah persamaan *input output* yang dapat direpresentasikan menggunakan *neural network* yang disajikan pada Gambar 3. 3(b). Persamaan *input output* menggunakan pohon diagnosa dapat di tuliskan seperti pada Tabel 3. 2 berikut.

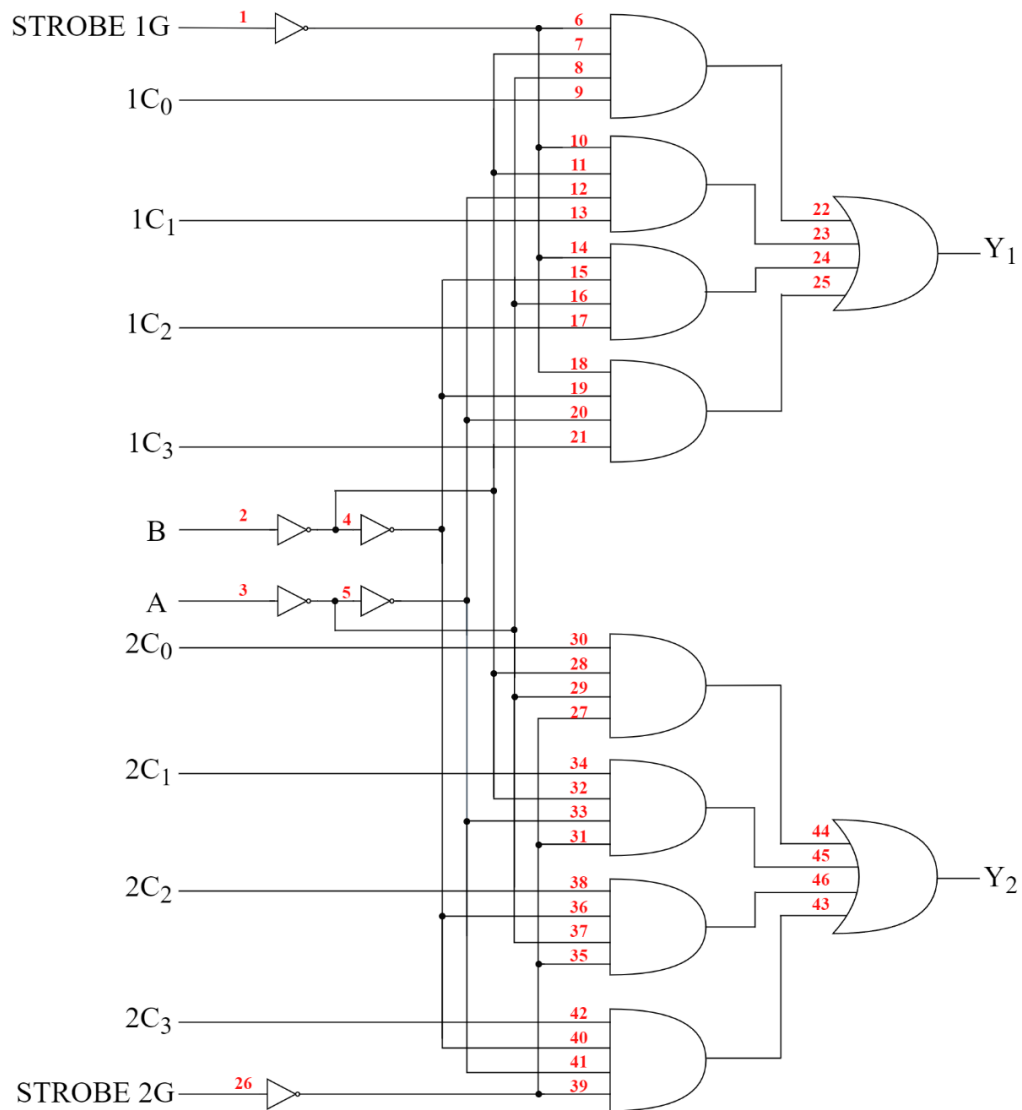
Tabel 3. 2. Tabel Kebenaran Pohon Diagnosa

Input			Output		
<i>out_cut</i>	a	b	na	nb	<i>flag</i>
0	0	0	X	X	0
0	0	1	1	0	0
0	1	0	0	0	1
0	1	1	0	1	1
1	0	0	X	X	0
1	0	1	1	0	1
1	1	0	1	1	1
1	1	1	0	1	0

Isi yang terdapat pada Tabel 3. 2 merupakan rangkaian pengujian yang dilakukan menggunakan pohon diagnosa dengan komponen a dan b merupakan variabel set uji dan *out_cut* merupakan *output* dari *circuit under test* atau rangkaian yang akan diuji ketiga komponen tersebut merupakan bagian *input*. Selanjutnya pada bagian *output* na dan nb merupakan variabel yang akan menunjukkan pengujian selanjutnya. *Flag* merupakan identitas apakah pengujian akan berlanjut atau berhenti. Jika *flag* bernilai 0 maka pengujian akan berlanjut, apabila *flag* bernilai 1 maka sudah menunjukkan sebuah lokasi kesalahan dan pengujian akan berhenti. Persamaan tabel kebenaran di atas merupakan untuk pohon diagnosa pada Gambar 2. 9 yaitu pohon diagnosa gerbang *AND*.

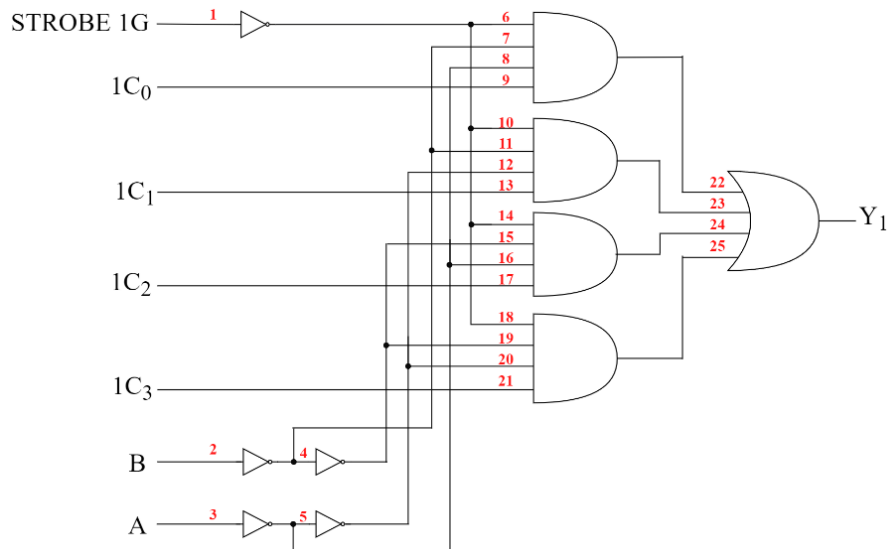
3. 7. Susunan Rangkaian Pengujian

Susunan rangkaian yang akan digunakan pada percobaan ini adalah rangkaian kombinasional *multiplexer dual 4 line to 1 line IC 74153*. Rangkaian terdiri dari 16 gerbang logika yaitu delapan gerbang *AND*, enam gerbang *NOT* dan dua gerbang *OR*. Terdapat 43 lokasi kesalahan dengan posisi seperti ditampilkan pada Gambar 3. 4. Pemetaan seluruh lokasi kesalahan dalam cakupan akan memudahkan dalam penentuan lokasi kesalahan.



Gambar 3. 4. Rangkaian *multiplexer dual 4 line to 1 line IC 74153*.

Rangkaian digital kombinasional dapat disederhanakan seperti Gambar 3. 5 dikarenakan masukan $\{1C_0, 1C_1, 1C_2, 1C_3\}$ dengan keluaran $\{Y_1\}$ identik dengan masukan $\{2C_0, 2C_1, 2C_2, 2C_3\}$ dengan keluaran $\{Y_2\}$ maka dapat dilakukan penyederhanaan menjadi potongan pada rangkaian *multiplexer dual 4 line to 1 line IC74153* seperti disajikan pada Gambar 3. 5. Berikut pada Persamaan 3. 1 merupakan persamaan *boolean* pada rangkaian *multiplexer dual 4 line to 1 line IC74153* yang digunakan dalam membuat sebuah tabel kesalahan.



Gambar 3. 5. Potongan 1 Rangkaian *Multiplexer Dual 4 Line to 1 Line IC 74153*.

$$Y_1 = \overline{G1}.B.A.1C_0 + \overline{G1}.B.A.1C_1 + \overline{G1}.B.A.1C_2 + \overline{G1}.B.A.1C_3 \quad (3.1)$$

Persamaan *boolean* mengacu pada data sheet dari *multiplexer dual 4 line to 1 line IC 74153*. Adapun persamaan identik potongan kedua disajikan pada Persamaan 3. 2.

$$Y_2 = \overline{G1}.B.A.2C_0 + \overline{G1}.B.A.2C_1 + \overline{G1}.B.A.2C_2 + \overline{G1}.B.A.2C_3 \quad (3.2)$$

Berdasarkan persamaan boolean pada persamaan 1 dan 2 maka menghasilkan sebuah tabel kebenaran yang dapat diacu untuk penentuan tahapan pembuatan pohon diagnosa. Berikut pada Tabel 3. 3 merupakan tabel kebenaran dari rangkaian *multiplexer dual 4 line to 1 line IC 74153*.

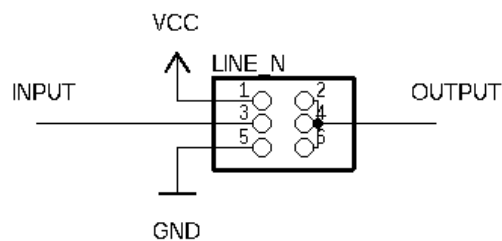
Tabel 3. 3. Tabel Kebenaran *IC 74153*

Select Input		Data Input				Strobe	Output
B	A	C ₀	C ₁	C ₂	C ₃	G	Y
X	X	X	X	X	X	H	L
L	L	L	X	X	X	L	L
L	L	H	X	X	X	L	H
L	H	X	L	X	X	L	L
L	H	X	H	X	X	L	H
H	L	X	X	L	X	L	L
H	L	X	X	H	X	L	H
H	H	X	X	X	L	L	L
H	H	X	X	X	H	L	H

Keterangan :

L : Low, H : High , X : Diabaikan

Tahapan implementasi rangkaian pengujian selanjutnya dinamakan *circuit under test (CUT)*. Rangkaian *circuit under test* dibangun menggunakan masing-masing gerbang logika penyusun IC 74153 diantaranya IC 7404 sebagai gerbang *NOT*, IC 4082 sebagai gerbang *AND* dengan 4 *input* 1 *output* dan IC 4072 sebagai gerbang *OR* dengan 4 *input* dan 1 *output*. Masing masing jalur penghubung IC diberikan sebuah percabangan kondisi keadaan normal, *stuck at 0 (Sa0)* maupun *stuck at 1 (Sa1)* yang akan merepresentasikan gangguan pada jalur tersebut. Titik percabangan kondisi posisinya sesuai dengan Gambar 3. 4 dan 3. 5 yang ditandai dengan angka berwarna merah. Pada implementasi, pencabangan dibuat menggunakan kabel penghubung sebagai peubah kondisi *Sa0*, normal dan *Sa1*. Selain itu, menggunakan *connector pin header* sebagai pin koneksi antara *circuit under test* dengan generator set uji. Koneksi *Sa1* dan *Sa0* disajikan pada Gambar 3. 6 berikut.

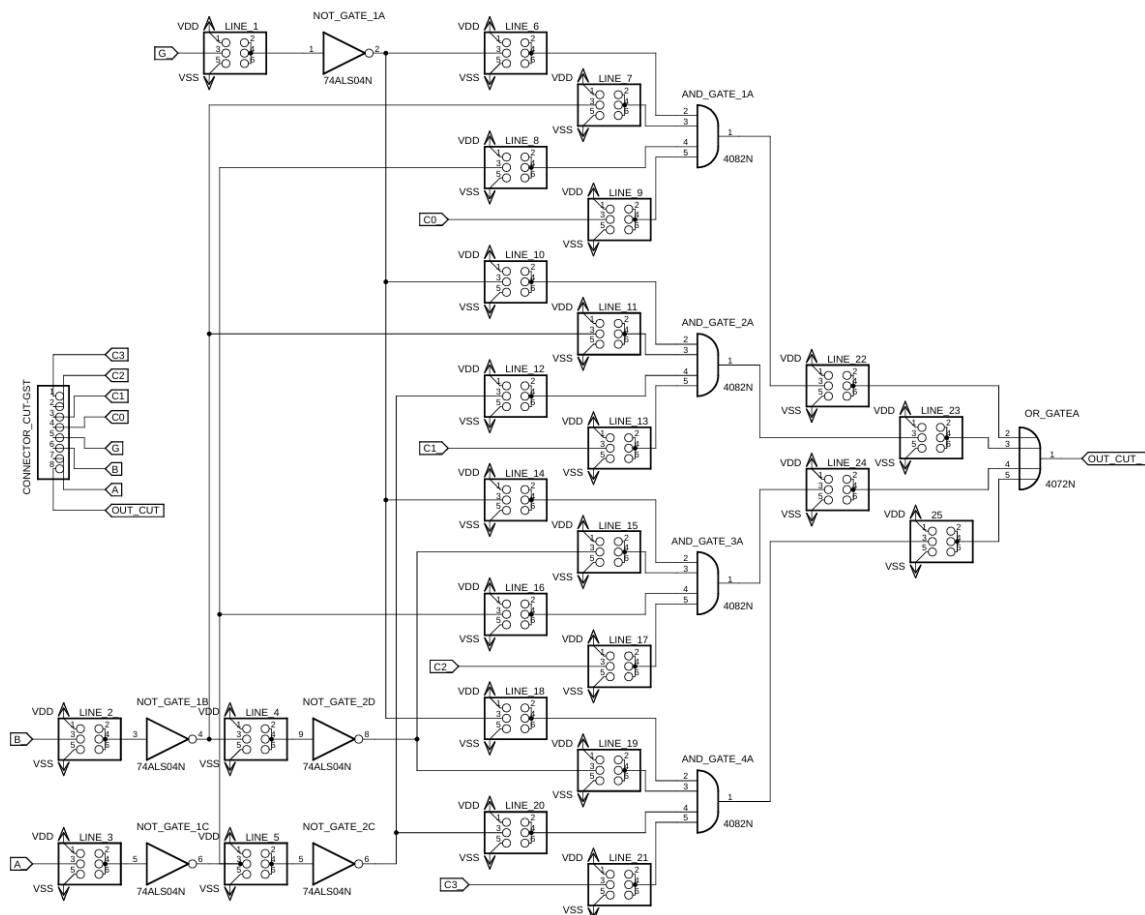


Gambar 3. 6. Kondisi *Sa1* dan *Sa0*.

Kondisi *Sa0* merupakan kondisi gangguan seolah jalur pada gerbang logika mengalami status *low* meskipun *input* mengalami *high* atau *low*. Kondisi *low* diidentifikasi dengan terhubung pin *output* kondisi terhubung pada *ground (GND)*. Kondisi *Sa1* merupakan kondisi gangguan seolah jalur pada gerbang logika mengalami status *high* meskipun *input* bernilai *high* atau *low*. Kondisi *high* diidentifikasi dengan terhubungnya pin *output* kondisi terhubung dengan pada sumber tegangan *voltage commond connector (VCC)*. Pada kondisi normal, seharusnya jalur *output* akan dipengaruhi langsung pada masukkan yang berikan.

Lokasi kesalahan dibagi atas 3 kelompok berdasarkan gerbang penyusun Dimana kelompok lokasi kesalahan pertama mencakup pada *line 1*, *line 2*, *line 3*, *line 4* dan *line 5* dengan gerbang penyusun gerbang *NOT*. Kelompok lokasi kesalahan kedua mencakup pada *line 6*, *line 7*, *line 8*, *line 9*, *line 10*, *line 11*, *line*

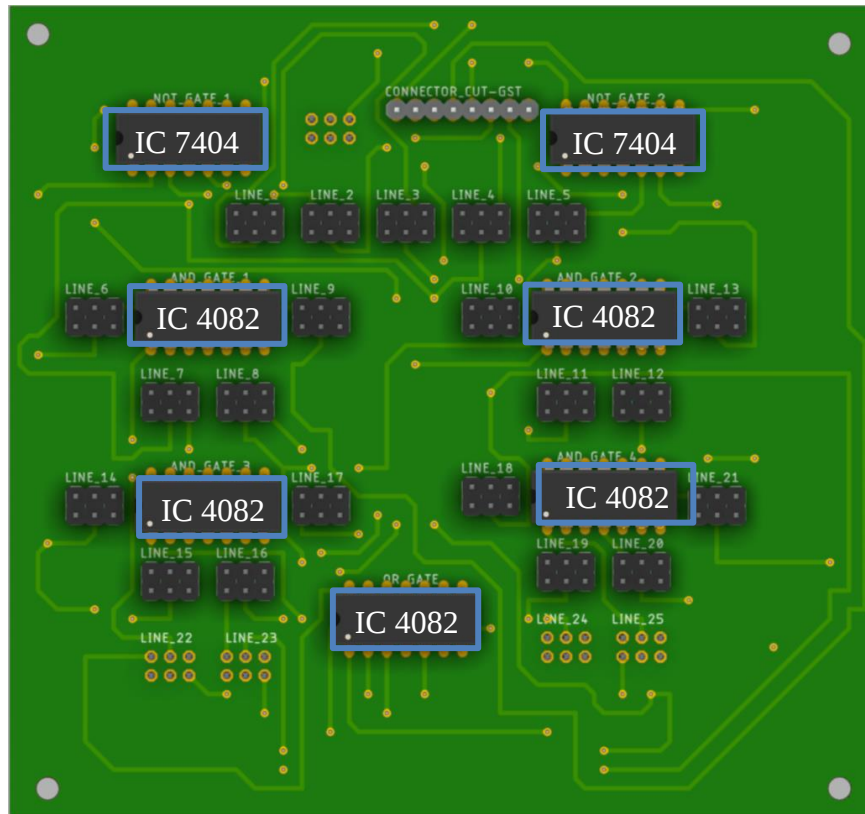
12, line 13, line 14, line 15, line 16, line 17, line 18, line 19, line 20 dan line 21 dengan gerbang penyusun gerbang *AND*. Kelompok lokasi kesalahan ketiga mencakup pada line 22, line 23, line 24 dan line 25 dengan gerbang penyusun gerbang *OR*. Masing-masing kelompok kesalahan dibuat untuk memudahkan proses sampling pengujian supaya tercakup seluruh bagian.



Gambar 3. 7. Diagram Pengkabelan CUT Multiplexer Dual 4 Line to 1 Line IC 74153.

Rangkaian pada Gambar 3. 7 merupakan hasil representasi dari rangkaian IC 74153 potongan 1 yang disusun menjadi kesatuan didalam sebuah *board* yang dinamakan rangkaian CUT IC 74153 yang disambungkan pada generator set uji menggunakan sebuah konektor 8 pin. Keterangan jalur di identifikasikan dengan nama jalur yang berpotensi gangguan *Sa0* dan *Sa1*. Implementasi kondisi *Sa0* ketika sebuah jalur terhubung dengan kutub negatif sumber dan kondisi *Sa1* ketika sebuah jalur terhubung dengan kutub positif baterai. Teridentifikasi potensi kerusakan terdapat total 25 jalur dan setiap jalur berpotensi memiliki kemungkinan kerusakan *Sa0* dan *Sa1*. Desain *printed circuit board* (PCB) dibuat menggunakan

double layer dengan susunan yang menyerupai diagram pengkabelan untuk memudahkan pencarian titik gangguan. Berikut Gambar 3. 8 merupakan tampilan desain *PCB* dari *circuit under test* IC 74153.



Gambar 3. 8. *PCB CUT Multiplexer Dual 4 Line to 1 Line IC 74153.*

Gambar 3. 8 menunjukkan desain yang dibuat berdasarkan diagram pengkabelan pada Gambar 3. 7. Melalui desain *circuit under test* yang dibuat, *printed circuit board PCB* merepresentasikan secara utuh terhadap prinsip kerja dari *circuit under test* hingga dapat melakukan pemetaan terhadap seluruh kesalahan pada masing-masing jalur.

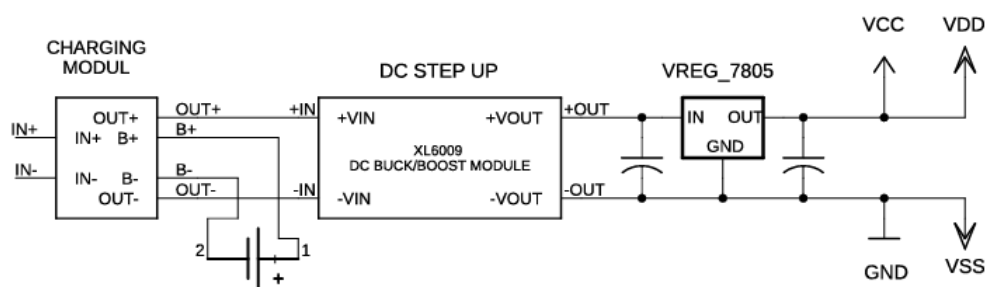
3. 8. Perancangan *Generator Set Test*

Generator set test (GST) merupakan sebuah rangkaian yang akan menguji *CUT* dan akan mendiagnosis kesalahan yang terjadi pada rangkaian *IC* yang digunakan. Generator set test terdiri atas mikrokontroler, modul catu daya, serta tambahan rangkaian *clock* peubah *state* yang dibangun menggunakan *switch* dan *potensio* sebagai pengatur jeda pengujian. Mikrokontroler yang digunakan yaitu *Arduino Mega Pro Mini 2560*. *Arduino Mega 2560* merupakan mikrokontroler

dengan ukuran kecil yang menggunakan *chipset* yang sama dengan *Arduino Mega 1280* dengan pengembangan sebelumnya yang ukurannya lebih besar. *Arduino Mega 2560* ini dilakukan penulisan *code* menggunakan *Integrated Development Environment (IDE) Arduino*[21]. Catu daya merupakan pusat pengelolaan daya yang digunakan dalam sistem untuk bekerja secara optimal. Rangkaian catu daya terdiri atas modul *charging*, modul *step-up* dan *voltage regulator*.

A. Subsistem Catu Daya

Subsistem catu daya merupakan subsistem yang menyuplai daya kedalam sistem. Subsistem ini terdiri dari *modul charging*, *baterai*, *step up voltage*, dan *voltage regulator*.



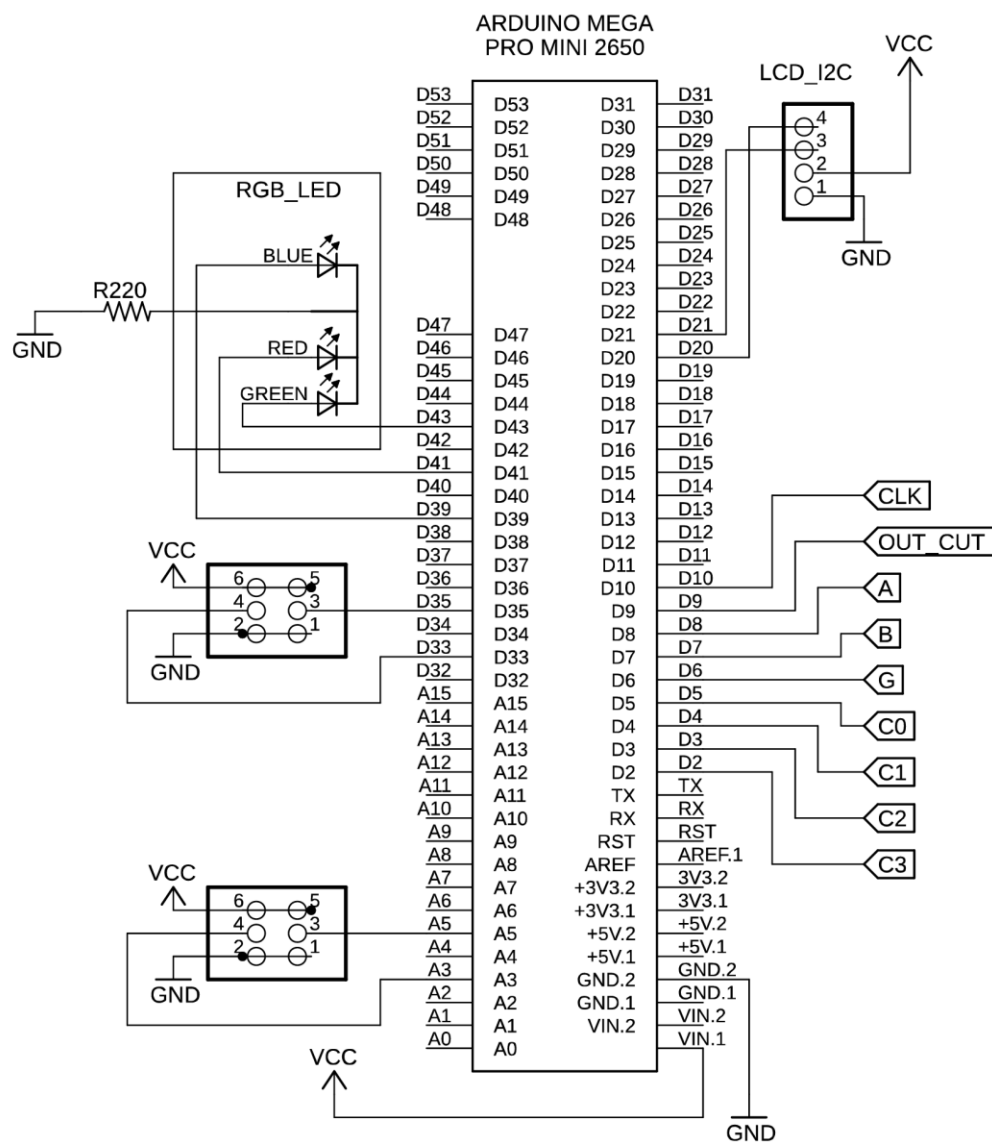
Gambar 3. 9. Diagram Pengkabelan Catu Daya.

Gambar 3. 9 merupakan diagram pengkabelan catu daya yang digunakan pada sistem generator set uji. *Modul charging* yang digunakan adalah *modul charging* tipe TP4056. *Modul charging* ini merupakan modul pengisian daya linier dengan kapasitas arus yang dialirkan untuk melakukan *charging* sebesar 1 *Ampere* dengan baterai *lithium ion* dengan port input *USB mini*[22]. Tegangan keluaran dari modul ini sebesar 4,2 V [23]. Perangkat selanjutnya pada subsistem catu daya adalah *DC step up XL6009* yang digunakan untuk menaikkan tegangan kerja secara langsung yang terdapat *postensio* untuk mengatur tegangan keluaran pada modul. Modul ini bekerja langsung tanpa perlu memberikan tegangan eksternal sehingga memudahkan untuk aplikasi yang membutuhkan proses menaikkan daya [24]. Perangkat terakhir pada modul ini adalah susunan *voltage regulator* yang menggunakan IC 7805 yang memiliki *output* sebesar 5 V yang sesuai dengan kebutuhan *device* modul. Modul IC 7805 merupakan sebuah modul yang

mempertahankan *output* pada 5 V dengan *input* maksimal 35 V [25]. Modul ini memiliki 3 pin dengan identifikasi *input*, *ground* dan *output*.

B. Subsistem Mikrokontroler

Subsistem inti dalam penelitian ini terletak pada subsistem mikrokontroler. Subsistem yang mengatur jalannya perintah, membaca *input* dan *output*, serta menampilkan informasi. Subsistem ini terdiri atas beberapa komponen yaitu mikrokontroler *Arduino Mega Pro Mini 2560*, lampu indikator, potensiometer, *switch*, penampil dan konektor yang menghubungkan antar komponen di luar modul. Berikut pada Gambar 3. 10 disajikan koneksi antara *Arduino* dengan komponen lain pada subsistem mikrokontroler.



Gambar 3. 10. Diagram Pengkabelan Mikrokontroler *Arduino Mega*.

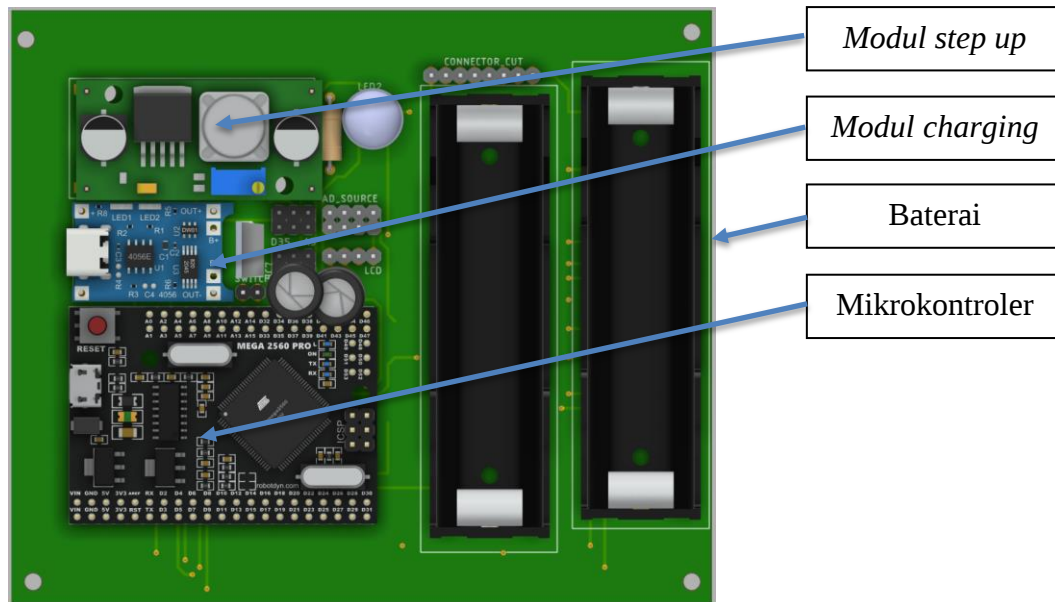
Koneksi pin yang digunakan pada mikrokontroler *Arduino Mega* merupakan koneksi satu kesatuan *pin* yang digunakan untuk menjalankan perintah, untuk menampilkan notifikasi, dan menerima informasi. Pin *D2, D3, D4, D5, D6, D7 dan D8* merupakan pin yang digunakan untuk mengirimkan sebuah nilai menuju *circuit under test*. Pin dengan inisiasi *D* merupakan pin digital dari mikrokontroler. Koneksi antara *generator set test* dan *circuit under test* menggunakan kabel jumper data dan daya. Kemudian terdapat notifikasi yang ditampilkan pada *OLED* menggunakan koneksi pin *Serial Data Line (SDA)* dan *Serial Clock Line (SCL)* berupa informasi pengujian dan informasi kode *error*. Selain *OLED*, indikator lampu juga digunakan sebagai notifikasi. Apabila lampu berkedip biru maka proses sedang melakukan pengujian, menyala merah merujuk pada sebuah lokasi kesalahan dan menyala hijau merujuk pada tidak adanya lokasi kesalahan pada sistem. Selanjutnya pada pin *switch* sebagai peubah kondisi waktu *delay*. Waktu *delay* digunakan untuk mempercepat atau memperlambat waktu pengujian sistem yang diatur menggunakan potensiometer yang terhubung dengan pin analog. *Delay* waktu diatur mulai dari 0 hingga 15 detik. Berikut pada Tabel 3. 4 merupakan koneksi pin mikrokontroler.

Tabel 3. 4. Koneksi Pin Mikrokontroler

<i>Pin Arduino</i>	Keterangan
D 2	<i>Pin connector C3 (CUT)</i>
D 3	<i>Pin connector C2 (CUT)</i>
D 4	<i>Pin connector C1 (CUT)</i>
D 5	<i>Pin connector C0 (CUT)</i>
D 6	<i>Pin connector G (CUT)</i>
D 7	<i>Pin connector B (CUT)</i>
D 8	<i>Pin connector A (CUT)</i>
D 9	<i>Pin connector OUT CUT (CUT)</i>
D 20	<i>Pin display SDA</i>
D 21	<i>Pin display SCL</i>
D 33	<i>Pin switch</i>
D 35	<i>Pin add</i>
D 39	<i>Pin LED biru</i>
D 41	<i>Pin LED hijau</i>
D 43	<i>Pin LED merah</i>
A 3	<i>Pin potensiometer pengatur delay</i>
A 5	<i>Pin add</i>

C. Generator Set Uji

Setelah masing masing subsistem dihubungkan menjadi satu kesatuan, maka proses terakhir pada generator set uji adalah proses penggabungan dilakukan pada sebuah papan *PCB*. Proses penggabungan dilakukan guna memudahkan dalam proses implementasi dan pengujian. Berikut pada Gambar 3. 11 merupakan *layout PCB* generator set uji.

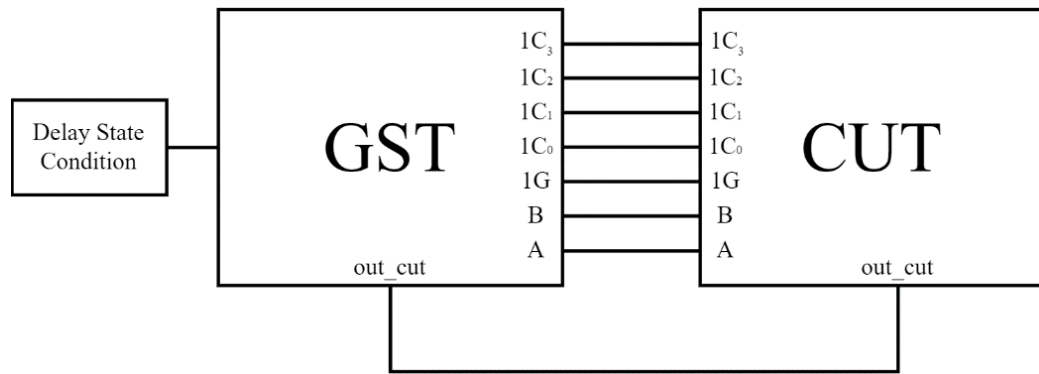


Gambar 3. 11 Tampak Atas Generator Set Uji.

Sistem terdiri atas 2 subsistem yaitu subsistem catu daya dan subsistem mikrokontroler digabungkan menjadi satu kesatuan yang dinamakan generator set uji.

3. 9. Experimental Setup

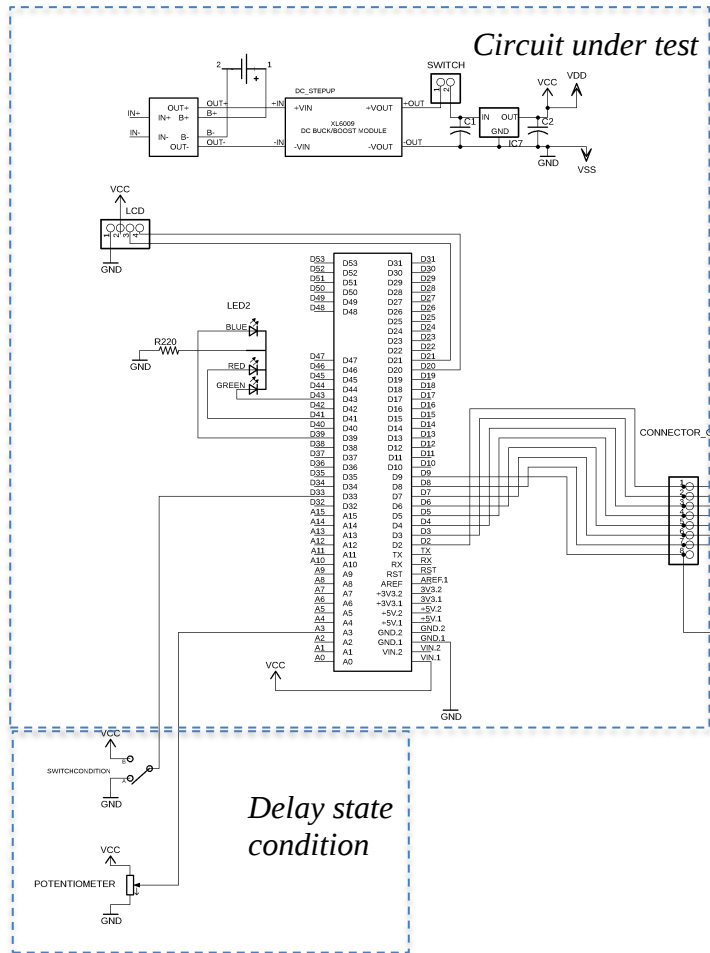
Experiment setup merupakan sebuah gambaran rangkaian yang digunakan pada proses pengujian. Secara umum, rangkaian pengujian dibagi atas dua keadaan yaitu generator set uji dan *circuit under test* dengan tambahan *delay state condition* yang digabungkan menggunakan *jumper*. *Delay state condition* merupakan sebuah pengatur waktu jeda antar pengujian satu ke pengujian selanjutnya dengan satuan detik. Manfaat penggunaan *delay state condition* adalah dapat diaturnya waktu jeda pengujian yang disesuaikan dengan kondisi. Berikut Gambar 3. 12 merupakan diagram *experimental circuit* yang digunakan.



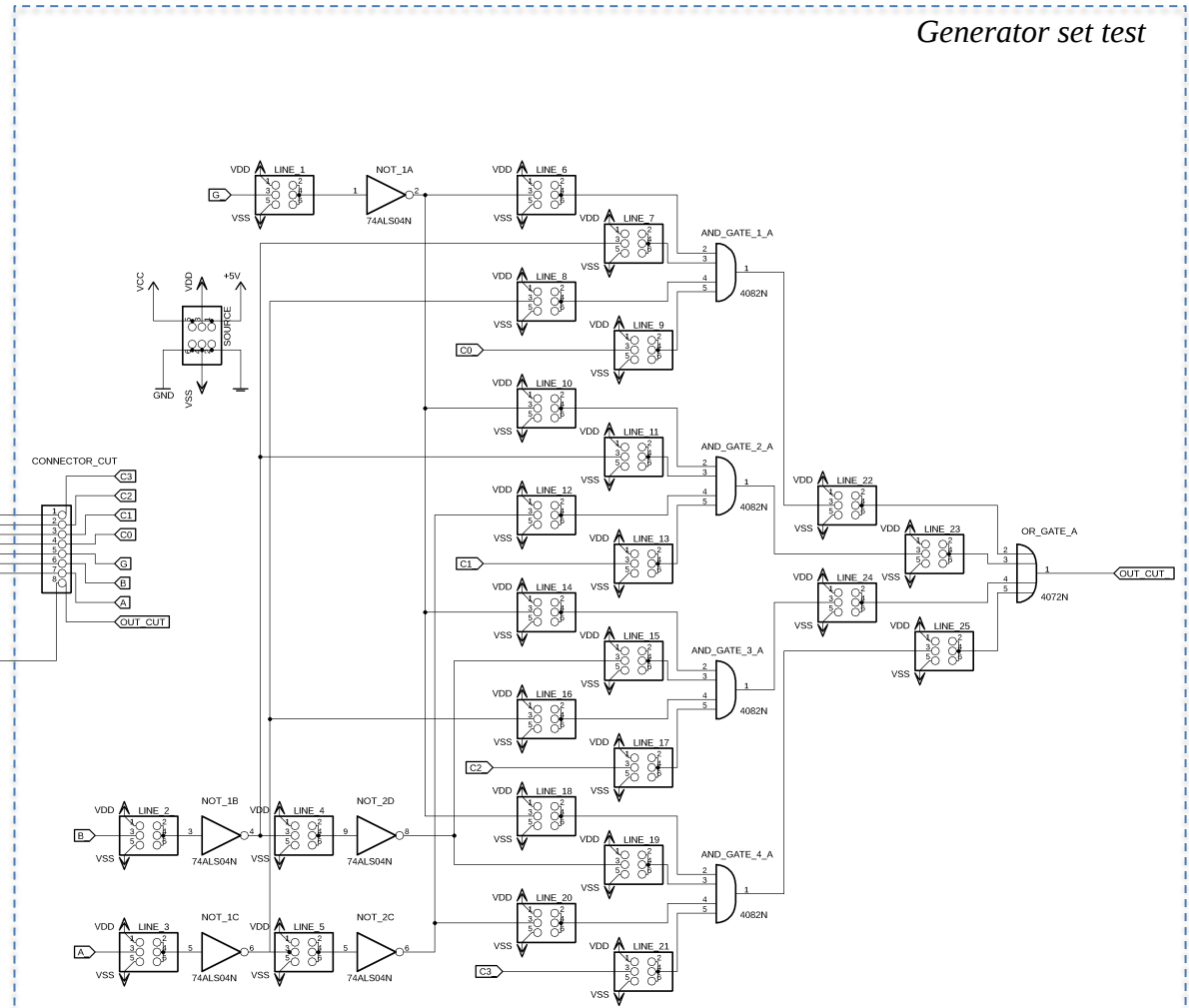
Gambar 3. 12. Diagram *Experimental Setup*.

Sisi generator set uji akan mengirimkan bit binner pada *CUT* melalui pin $1C_3$, $1C_2$, $1C_1$, $1C_0$, $1G$, B , A kemudian *CUT* akan merespon melalui pin *out_cut*. Pada percobaan akan dilakukan eksperimen berupa perubahan kondisi untuk dilakukan proses diagnosis kesalahan untuk menguji tingkat kemampuan *GST* dalam melakukan pendeteksian lokasi kesalahan. Secara lengkap, akan disajikan pada Gambar 3. 13.

Experimental Setup



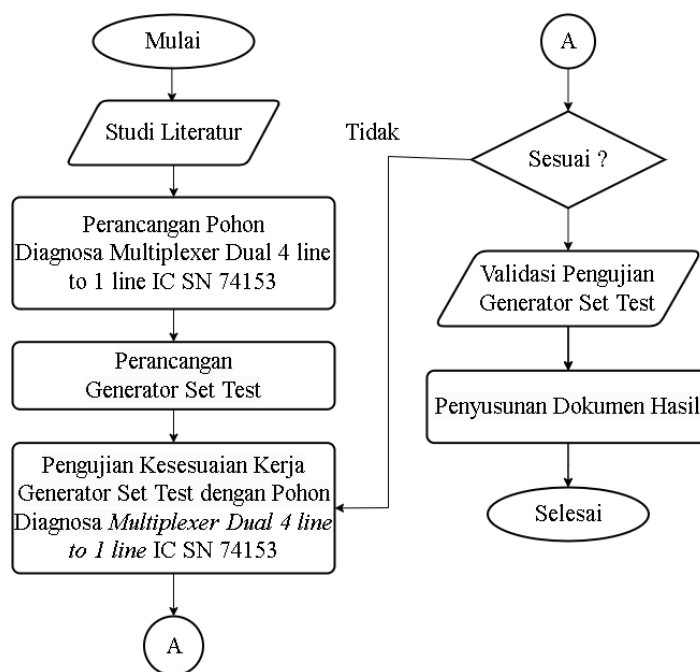
Generator set test



Gambar 3. 13. Experiment Setup Lengkap

3. 10. Diagram Alir Penelitian

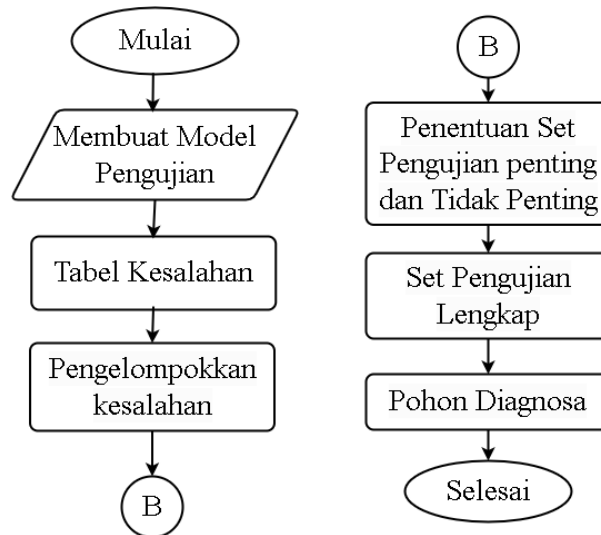
Diagram alir penelitian merupakan keseluruhan prosedur yang digunakan pada penelitian ini. Tahapan awal dilakukukan pencarian referensi terkait penelitian yang pernah dilakukan meliputi deteksi kesalahan pada rangkaian digital kombinasional dan perancangan sistem *neural network*. Kemudian dilakukan perancangan pohon diagnosa pada IC yang digunakan yaitu IC 74153. Proses dilanjutkan dengan membuat generator set uji menggunakan *neural network* yang dibangun pada mikrokontroler. Setelah generator set uji jadi, dilakukan pengujian kesesuaian apakah melalui struktur yang dibangun, model dapat bekerja sesuai dengan target capaian atau tidak. Jika model tidak dapat mencapai kesesuaian maka struktur dirubah kembali hingga mencapai target capaian. Apabila sudah sesuai maka proses dilanjutkan dengan melakukan validasi terhadap pengujian yang dilakukan. Terakhir dilakukan peroses pembuatan dokumen akhir. Gambar 3. 14 merupakan diagram alir utama yang digunakan dalam melakukan penelitian.



Gambar 3. 14. Diagram Alir Penelitian

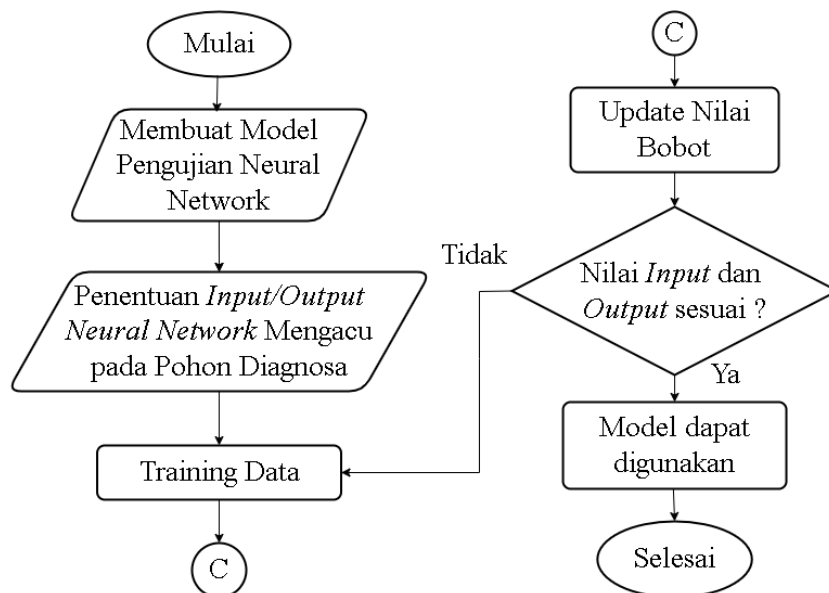
Tahapan yang dilakukan oleh generator set uji dalam melakukan diagnosa dari kesalahan dimulai dari penentuan model pengujian menggunakan rangkaian digital kombinasional yang dibuat pula ekspresi *boolean*. Selanjutnya akan

dimodelkan tabel kesalahan hingga membuat kelompok kesalahan, kemudian dilakukan pembuatan pohon diagnosa yang disajikan pada Gambar 3.15.



Gambar 3. 15. Diagram Alir Pengujian Pohon Diagnosa.

Setelah pohon diagnosa selesai dibuat, maka dijadikan sebuah persamaan *input output* dengan memberikan sebuah karakter tambahan sebagai tanda untuk proses berjalan dan berhenti. Selanjutnya rangkaian dapat melakukan pengujian pada sebuah *circuit under test* serta melakukan diagnosis kesalahan.



Gambar 3. 16. Diagram Alir Pengujian *Neural Network*.

BAB V

KESIMPULAN DAN SARAN

5. 1. Kesimpulan

Berdasarkan hasil perancangan, implementasi dan pembahasan maka didapat kesimpulan pada penelitian ini meliputi

1. Telah terbangun generator set uji rangkaian digital kombinasional yang bekerja sesuai dengan pohon diagnosa. Level pengujian terpendek yaitu 3 level pengujian dengan kode kesalahan f_{31} berlokasi di jalur 22 mengalami *Sa1* atau jalur 23 mengalami *Sa1* atau jalur 24 mengalami *Sa1* atau jalur 25 mengalami *Sa*. Level pengujian terpanjang yaitu pada f_0 dan f_{22} , f_0 merupakan tidak ada kesalahan dan f_{22} berlokasi di jalur 14 mengalami *Sa1*.
2. Telah dibangun sebuah model jaringan saraf tiruan yang mengacu pada implementasi pohon diagnosa rangkaian digital kombinasional *IC 74153* yang mampu melakukan prediksi dengan tingkat akurasi 99,95%. Dengan struktur menggunakan perulangan jumlah *dataset input output* sebanyak 4 kali, menggunakan 5 *hidden layer* dengan masing-masing *neuron* 30, 41, 49, 50, 51 dan jumlah *epoch* sebanyak 500 kali.
3. Generator set uji yang dibangun mampu melakukan pendeteksian kesalahan tunggal dengan rata-rata tingkat efisiensi pengujian sebesar 95,2 %. Efisiensi maksimal pada pengujian f_{31} sebesar 97,7% dan efisiensi minimal pada pengujian f_0 dan f_{22} sebesar 90,6%. Selain kesalahan tunggal dilakukan uji *sampling* pada kesalahan jamak dua dan tiga lokasi kesalahan. Hasilnya pada kesalahan jamak yang dipilih generator set uji mampu mendeteksi lokasi kesalahan dengan tahapan pencarian satu per satu lokasi kesalahan, apabila seluruh lokasi kesalahan telah diperbaiki maka akan menunjukkan hasil akhir f_0 atau tidak ada terdeteksi lokasi kesalahan.

5. 2. Saran

Berdasarkan penelitian yang telah dilakukan, terdapat saran untuk penelitian selanjutnya guna melakukan perbaikan ke depan yaitu diharapkan, terdapat suatu

model yang dapat menjalankan beberapa fungsi pohon diagnosa berbasis *neural network*. Artinya selama cakupan *input output generator set test* tersedia, maka rangkaian kombinasional dapat dilakukan pendeteksian kesalahan. Sehingga tingkat efisiensi dapat ditingkatkan.

DAFTAR PUSTAKA

DAFTAR PUSTAKA

- [1] S. Heo and J. H. Lee, "Fault Detection and Classification Using Artificial Neural Networks," *Int. Symp. Adv. Control Chem. Process.*, vol. 1, pp. 464–469, 2018.
- [2] A. Yadav and Y. Dash, "An Overview of Transmission Line Protection by Artificial Neural Network: Fault Detection, Fault Classification, Fault Location, and Fault Direction Discrimination," *Adv. Artif. Neural Syst.*, vol. 2014, pp. 1–20, Dec. 2014.
- [3] R. Isermann, "Model-Based Fault-Detection and Diagnosis - Status and Applications," *Annu. Rev. Control*, vol. 29, no. 1, pp. 71–85, 2005.
- [4] D. Miljković, "Fault Detection Methods: A Literature Survey," *MIPRO 2011/CTS*, pp. 110–115, 2011, [Online]. Available: <https://www.researchgate.net/publication/221412815>
- [5] R Isermann, *Fault-Diagnosis System*. 2006.
- [6] G. R. Yang and X. J. Wang, "Artificial Neural Networks for Neuroscientists: A Primer," *Neuron*, vol. 107, no. 6, pp. 1048–1070, Sep. 2020.
- [7] I. A. Basheer and M. Hajmeer, "Artificial Neural Networks: Fundamentals, Computing, Design, and Application," *J. Methods Microbiol. J. Microbiol. Methods*, vol. 43, pp. 3–31, 2000, [Online]. Available: www.elsevier.com/locate/jmicmeth
- [8] Sm Thamarai, K. Kuppusamy, and T. Meyyappan, "Fault Detection and Test Minimization Methods for Combinational Circuits-A Survey," *Int. J. Comput. Trends Technol.*, vol. 2, no. 2, pp. 140–146, 2011, [Online]. Available: <http://www.internationaljournals.org>
- [9] A. Smith, A. Veneris, M. Fahim Ali, and A. Viglas, "Fault diagnosis and Logic Debugging using Boolean Satisfiability," *IEEE Trans. Comput. Des. Integr. Circuits Syst.*, vol. 24, no. 10, pp. 1606–1620, Oct. 2005.
- [10] A.S. Repelianto, "Penggunaan Metoda Tabel Kesalahan untuk Diagnosis Kesalahan Rangkaian Digital Kombinasional," Universitas Sriwijaya, 1996.

- [11] R. . Setiawan, “Rancang Bangun Generator Set Uji untuk Diagnosis Kesalahan Rangkaian Multiplexer Quad 2 Line To 1 Line IC 74157 Menggunakan Metode Tabel Kesalahan Berbasis Arduino Nano Atmega328,” Universitas Lampung, 2023.
- [12] S. Heo and J. H. Lee, “Fault Detection and Classification using Artificial Neural Networks,” 2018.
- [13] M. Uzair and N. Jamil, “Effects of Hidden Layers on the Efficiency of Neural networks,” in *2020 IEEE 23rd International Multitopic Conference (INMIC)*, Pakistan: IEEE, Nov. 2020, pp. 1–6. [Online]. Available: <https://ieeexplore.ieee.org/document/9318195/>
- [14] O. Isaac Abiodun, A. Jantan, A. Esther Omolara, K. Victoria Dada, N. AbdElatif Mohamed, and H. Arshad, “State-of-the-art in Artificial Neural Network Applications: A Survey,” *Heliyon*, vol. 4, p. 938, 2018, [Online]. Available: <https://doi.org/10.1016/j.heliyon.2018.e00938>
- [15] Y. Andrian and M. R. Wayahdi, “Analisis Perbandingan Metode Perceptron dan Backpropagation dalam Mengenali Gerbang Logika,” *Konf. Nas. Pengemb. Teknol. Inf. dan Komun.*, pp. 195–200, 2014.
- [16] Hoiriyah, “Simulasi Gerbang Dasar Logika dalam Aplikasi,” *J. JURTIE*, vol. 2, no. 2, pp. 1–08, 2020, [Online]. Available: <https://jurnal.ugp.ac.id/index.php/jurtie>
- [17] A. B. Pulungan and S. R. Putri, “Data Logger berbasis IC Multiplexer untuk Pengukuran Arus dan Tegangan pada Panel Surya Skala Besar,” *JTEV (Jurnal Tek. Elektro dan Vokasional)*, vol. 8, no. 1, p. 151, 2022.
- [18] A. Mentaruk, X. Najoran, and A. Lumenta, “Implentasi Sistem Keamanan Toko Berbasis Internet of Things,” *J. Tek. Inform.*, vol. 15, no. 4, pp. 325–332, 2020.
- [19] O. I. Abiodun *et al.*, “Comprehensive Review of Artificial Neural Network Applications to Pattern Recognition,” *IEEE Access*, vol. 7, pp. 158820–158846, 2019, [Online]. Available: <https://ieeexplore.ieee.org/document/8859190/>
- [20] Ekojono, Faisal Rahutomo, and D. N. Sari, “Implementasi Library Deep Learning Keras pada Sistem Ujian Essay Online,” *J. Inform. Polinema*, vol.

- 6, no. 2, pp. 73–79, 2020.
- [21] W. Wahyudi, W. Purnamasari S, A. Hidayat, and M. M. Fakhri, “Penerapan Machine Learning Pada Mikrokontroler Arduino Mega PRO MINI ATmega2560-16AU,” *J. Embed. Syst. Secur. Intell. Syst.*, vol. 3, no. 1, p. 30, 2022.
 - [22] R. Ananda and M. Amin, “Utilization of Booster Circuit Joule Thief for Garden Lighting Source of Voltage From the Sun,” *JURTEKSI (Jurnal Teknol. dan Sist. Informasi)*, vol. 8, no. 1, pp. 103–110, 2021.
 - [23] R. Ananda and W. Handoko, “Penggunaan Rangkaian Booster Converter Dan Ic-Tp4056 Untuk Lampu Jalan Murah,” *JURTEKSI (Jurnal Teknol. dan Sist. Informasi)*, vol. 7, no. 1, pp. 9–14, 2020.
 - [24] R. Kumar Tripathi and J. Prasad, “Self-Starting Power Management System for Sediment Microbial Fuel Cell To Power Electronic Devices,” *Int. Conf. Power Electron. Sustain. Dev.*, vol. 3, no. March, pp. 206–210, 2024.
 - [25] K. Selvaraj and A. Chakrapani, “An Efficient and Smart Fan Using IR Communication,” *Int. J. Adv. Res. Basic Eng. Sci. Technol. An*, vol. 6, no. may, pp. 1–10, 2020.
 - [26] M. S. Wibawa, “Pengaruh Fungsi Aktivasi, Optimisasi dan Jumlah Epoch Terhadap Performa Jaringan Saraf Tiruan,” *J. Sist. dan Inform.*, vol. 11, no. December, pp. 167–174, 2017, [Online]. Available: <http://archive.ics.uci.edu/ml/datasets/Wine>
 - [27] M. Hafizh, F. Hamami, and T. F. Kusumasari, “Prediksi Network Capacity Planning PT XYZ Menggunakan Algoritma Recurrent Neural Network (RNN),” vol. 10, no. 5, p. 4880, 2023.
 - [28] D. Bashir, G. D. Montañez, S. Sehra, P. S. Segura, and J. Lauw, “An Information-Theoretic Perspective on Overfitting and Underfitting,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 12576 LNAI, pp. 347–358, 2020.
 - [29] J. E. R. S. Michael L. Commons, Stephen Grossberg, *Neural Network Models of Conditioning and Action*, 6th ed. Francis: Routledge, 2017.