



**PERAMALAN KAPASITAS BATERAI LEAD ACID PADA
MOBIL LISTRIK BERBASIS LEVENBERG
MARQUARDT NEURAL NETWORK**

SKRIPSI

Oleh

**Mambak Udin
NIM 121910201036**

**PROGRAM STUDI STRATA 1 TEKNIK
JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS JEMBER
2016**



**PERAMALAN KAPASITAS BATERAI LEAD ACID PADA
MOBIL LISTRIK BERBASIS LEVENBERG
MARQUARDT NEURAL NETWORK**

SKRIPSI

diajukan guna melengkapi tugas akhir dan memenuhi salah satu syarat
untuk menyelesaikan studi di Jurusan Teknik Elektro (S1)
dan mencapai gelar Sarjana Teknik

Oleh

**Mambak Udin
NIM 121910201036**

**PROGRAM STUDI STRATA 1 TEKNIK
JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS JEMBER
2016**

PERSEMBAHAN

Syukur Alhamdulillah segala puji bagi Allah SWT atas segala limpahan rahmat dan karunia-Nya kepada penulis sehingga dapat menyelesaikan skripsi yang berjudul “Peramalan Kapasitas Baterai *Lead Acid* pada Mobil Listrik Berbasis *Levenberg Marquardt Neural Network*” dan skripsi ini merupakan gerbang awal dalam mencapai kesuksesan yang lebih baik lagi. Oleh karena itu, saya ingin mempersembahkan karya ini kepada :

1. Allah SWT, Tuhan yang bergantung kepada-Nya segala sesuatu.
2. Nabi Muhammad SAW, utusan Allah SWT untuk menjadi pedoman manusia.
3. Ibu Musringah dan Bapak Mudjito tercinta, yang telah mendidik, memberi kasih sayang dan mendoakan.
4. Semua Dosen Jurusan Teknik Elektro Fakultas Teknik Universitas Jember yang telah membimbing dan memberikan ilmu. Terutama Dosen Pembimbing Bapak Dr. Bambang Sri Kaloko, S.T., M.T. dan Dr. Triwahju Hardianto, S.T., M.T. yang telah membimbing hingga terselesaikannya skripsi ini.
5. Saudaraku seperjuangan Teknik Elektro angkatan 2012 “SATE UJ”.
6. Almamater Fakultas Teknik Universitas Jember.
7. Semua teman – teman yang telah menemani selama tinggal di Jember. Keluarga Asy-Syabab, Al-Mumtaz, dan lain – lain.
8. Serta semua pihak yang belum tertulis dalam lembar persembahan ini, terima kasih atas segalanya.

MOTTO

“Hai orang-orang mukmin, jika kamu menolong (agama) Allah, niscaya Dia akan menolongmu dan meneguhkan kedudukanmu”

(Q.S. Muhammad : 07)

“Dan Allah telah berjanji kepada orang-orang yang beriman di antara kamu dan mengerjakan amal-amal yang saleh bahwa Dia sungguh-sungguh akan menjadikan mereka berkuasa dimuka bumi, sebagaimana Dia telah menjadikan orang-orang sebelum mereka berkuasa, dan sungguh Dia akan meneguhkan bagi mereka agama yang telah diridhai-Nya untuk mereka, dan Dia benar-benar akan menukar (keadaan) mereka, sesudah mereka dalam ketakutan menjadi aman sentausa. Mereka tetap menyembahku-Ku dengan tiada mempersekutukan sesuatu apapun dengan Aku. Dan barangsiapa yang (tetap) kafir sesudah (janji) itu, maka mereka itulah orang-orang yang fasik”

(Q.S. An - Nuur : 55)

“Sebaik - baik manusia adalah yang paling bermanfaat bagi manusia lain”

(HR. Ahmad, ath - Thabrani, ad - Daruqutni. Di dalam shahihul Jami' no. 3289)

“Barang siapa yang memudahkan kesulitan seorang mu'min dari berbagai kesulitan - kesulitan dunia, Allah akan memudahkan kesulitan - kesulitannya pada hari kiamat. Dan siapa yang memudahkan orang yang sedang dalam kesulitan niscaya akan Allah memudahkan baginya di dunia dan akhirat”

(HR. Muslim)

PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : Mambak Udin

NIM : 121910201035

menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul :
“Peramalan Kapasitas Baterai *Lead Acid* pada Mobil Listrik Berbasis *Levenberg Marquardt*” adalah benar-benar karya sendiri, kecuali kutipan yang sudah saya sebutkan sumbernya, belum pernah diajukan pada institusi manapun, dan bukan karya jiplakan. Saya bertanggung jawab atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

Demikian pernyataan ini saya buat dengan sebenarnya, tanpa adanya tekanan dan paksaan dari pihak mana pun serta bersedia mendapat sanksi akademik jika ternyata di kemudian hari pernyataan ini tidak benar.

Jember, 29 juni 2016

Yang menyatakan,

Mambak Udin

NIM 121910201036

SKRIPSI

**PERAMALAN KAPASITAS BATERAI LEAD ACID PADA
MOBIL LISTRIK BERBASIS LEVENBERG
MARQUARDT NEURAL NETWORK**

Oleh
Mambak Udin
NIM 121910201036

Pembimbing

Dosen Pembimbing Utama : Dr. Bambang Sri Kaloko, S.T., M.T.
Dosen Pembimbing Anggota : Dr. Triwahju Hardianto, S.T., M.T.

PENGESAHAN

Skripsi berjudul “Peramalan Kapasitas Baterai *Lead Acid* pada Mobil Listrik Berbasis *Levenberg Marquardt Neural Network*” telah diuji dan disahkan oleh Fakultas Teknik Universitas Jember pada:

Hari, tanggal : Kamis, 30 Juni 2016

Tempat : Ruang Sidang 1 Fakultas Teknik Universitas Jember

Tim Penguji :

Pembimbing Utama,

Pembimbing Anggota,

Dr. Bambang Sri Kaloko, S.T., M.T.
NIP 19710402 2003121001

Dr. Triwahju Hardianto, S.T., M.T.
NIP 197008261997021001

Penguji I,

Penguji II,

Dedy Kurnia Setiawan, S.T., M.T.
NIP 198006102005011003

RB. Moch. Gozali, S.T., M.T.
NIP 196906081999031002

Mengesahkan
Dekan,

Dr.Ir. Entin Hidayah M.U.M
NIP 196612151995032001

**PERAMALAN KAPASITAS BATERAI LEAD ACID PADA MOBIL
LISTRIK BERBASIS LEVENBERG MARQUARDT NEURAL NETWORK**

Mambak Udin

Jurusan Teknik Elektro, Fakultas Teknik, Universitas Jember

ABSTRAK

Mobil listrik pada saat ini menjadi pembicaraan banyak orang dan sangat populer. Namun, jumlah mobil listrik yang ada di dunia masih jauh dibandingkan dengan mobil konvensional. Salah satu pembahasan dalam penelitian mobil listrik adalah sumber energi atau baterai. Baterai pada mobil listrik mempunyai kapasitas yang terbatas sehingga harus dilakukan pengisian ulang agar mobil listrik tetap bisa berjalan jauh. Keadaan ini bila tidak segera diatasi bisa menyebabkan baterai tiba – tiba habis ditengah jalan. Tentu hal ini membuat cemas dan tidak nyaman bagi pengguna mobil listrik. Oleh karena itu diperlukan peramalan kapasitas baterai agar diketahui kapan waktunya untuk mengisi ulang baterai atau bahkan menggantinya. Penelitian ini membahas peramalan kapasitas baterai mobil listrik berbasis kecerdasan buatan *levenberg marquardt neural network*, serta membandingkan dengan sistem *recurrent neural network*. Kondisi mobil listrik menggunakan tiga kondisi beban yaitu kecepatan maksimal, akselerasi awal, dan tanjakan. Hasil penelitian menunjukkan peramalan sisa kapasitas baterai saat kecepatan maksimal setelah pemakaian selama 30 menit, sisa kapasitas baterai yang dapat digunakan diramalkan dengan backpropagation sebesar 61 Ah, dengan recurrent sebesar 60,5 Ah. Sedangkan target kapasitas sebesar 59,6 Ah. Sehingga metode terbaik dalam meramalkan kapasitas baterai adalah menggunakan recurrent neural network.

Kata kunci : Peramalan, baterai *lead acid*, mobil listrik, neural network, *levenberg marquardt*

**Prediction Capacity of Lead Acid Battery on Electric Vehicle Based Neural
Network Levenberg Marquardt**

Mambak Udin

Department of Electrical Engineering, Engineering Faculty, University of Jember

ABSTRACT

Electric vehicle currently the talk of many people and is very popular. However, the number of electric vehicle in the world is still less than a conventional car. One of the discussion in the research of electric vehicle is the energy source or battery. Batteries in electric vehicle have limited capacity so it must be done so that recharging of electric vehicle can still walk away. This situation if not resolved could cause the battery suddenly depleted middle of the road. Of course this makes anxious and uncomfortable for users of electric vehicle. Therefore, it is necessary forecasting capacity of the battery in order to know when it's time to recharge the battery or replace it. This paper discusses the forecasting capacity of electric car batteries Levenberg Marquardt based artificial intelligence neural network, and compares with a recurrent neural network system. Conditions electric vehicle using three load conditions, namely the maximum speed, the initial acceleration and gradeability. The results showed forecasting the remaining battery capacity when the maximum speed for 30 minutes after consumption, the remaining battery capacity that can be predicted by the propagation method is 61 Ah, with recurrent amounted to 60.5 Ah. While the target capacity is 59.6 Ah. So the best method of predicting the battery capacity is using recurrent neural network.

Keywords : *Forecasting, lead acid batteries, electric vehicle, neural network, Levenberg Marquardt*

RINGKASAN

“Peramalan Kapasitas Baterai Lead Acid pada Mobil Listrik Berbasis Levenberg Marquardt Neural Network”; Mambak Udin; 121910201036; 2016; 60 halaman; Program Studi Strata 1 (S1) Teknik, Jurusan Teknik Elektro, Fakultas Teknik Universitas Jember.

Mobil listrik mempunyai beberapa keuntungan seperti efisiensi yang tinggi, tingkat pencemaran lingkungan yang rendah, tingkat kebisingan yang rendah, energinya tersedia dari berbagai sumber energi alternatif, mudah perawatannya, dan regeneratif. Salah satu pembahasan dalam penelitian mobil listrik adalah sumber energi atau baterai. Baterai pada mobil listrik mempunyai kapasitas yang terbatas sehingga harus dilakukan pengisian ulang agar mobil listrik tetap bisa berjalan jauh. Keadaan ini bila tidak segera diselesaikan bisa menyebabkan baterai mendadak habis ditengah jalan. Tentu hal ini membuat cemas dan tidak nyaman bagi pengguna mobil listrik. Oleh karena itu diperlukan peramalan kapasitas baterai agar diketahui sisa kapasitas baterai setelah terpakai.

Penelitian ini membahas peramalan kapasitas baterai mobil listrik berbasis kecerdasan buatan *levenberg marquardt neural network*, serta membandingkan dengan sistem *recurrent neural network*. Kondisi mobil listrik menggunakan tiga kondisi beban yaitu kecepatan maksimal, akselerasi awal, dan tanjakan. Perancangan model sistem menggunakan 8 baterai *lead acid* yang dirangkai seri masing-masing 4 baterai kemudian diparalel. Kapasitas masing-masing baterai sebesar 70 Ah dan tegangan 12 V. Sedangkan beban dimodelkan dengan sebuah motor dengan daya 5 Hp tegangan 240 V dan 1750 rpm.

Metode terbaik dalam meramalkan sisa kapasitas baterai adalah dengan metode *recurrent neural network*. Hal ini ditunjukkan salah satunya, saat menggunakan kecepatan maksimum (16,67 m/s) hasil peramalan setelah 30 menit berjalan sisa kapasitas baterai dari *backpropagation* adalah 60,5 Ah dan *recurrent* adalah 61 Ah sedangkan targetnya adalah 59,6 Ah.

SUMMARY

“Prediction Capacity of Lead Acid Battery on Electric Vehicle Based Neural Network Levenberg Marquardt”; Mambak Udin; 2016: 60 pages; Bachelor degree (S1) Engineering, Department of Electrical Engineering, Faculty of Engineering University of Jember

Electric vehicle have several advantages such as high efficiency, low levels of environmental pollution, low noise levels, the energy available from a variety of alternative energy sources, easy to maintain, and regenerative. One of the discussion in the research of electric cars is the energy source or battery. Batteries in electric cars have limited capacity so it must be done so that recharging of electric cars can still walk away. This situation if not immediately resolved could cause the battery suddenly depleted middle of the road. Of course this makes anxious and uncomfortable for users of electric cars. Therefore, it is necessary forecasting capacity of the battery in order to know the remaining battery capacity after use.

This study discusses the forecasting capacity of electric car batteries Levenberg Marquardt-based artificial intelligence neural network, and compares with a recurrent neural network system. Conditions electric cars using three load conditions, namely the maximum speed, the initial acceleration and gradeability. The design of the system model uses 8 lead acid batteries were assembled each series 4 battery then paralel. Each capacity of 70 Ah battery and a voltage of 12 V. The load is modeled by a 5 hp motor with power voltage of 240 V and 1750 rpm.

The best method to predict remaining battery capacity is the recurrent neural network method. This is shown one of them, when using the maximum speed (16.67 m/s) forecasting results after 30 minutes of running remaining battery capacity of propagation is 60.5 Ah and 61 Ah is recurrent, while the target is 59.6 Ah

PRAKATA

Puji syukur kehadiran Allah SWT, atas segala rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul “Peramalan Kapasitas Baterai *Lead Acid* pada Mobil Listrik Berbasis *Levenberg Marquardt Neural Network*”. Skripsi ini disusun untuk memenuhi salah satu syarat menyelesaikan pendidikan Strata Satu (S1) pada Jurusan Teknik Elektro Fakultas Teknik Universitas Jember.

Penyusunan skripsi ini tidak lepas dari bantuan berbagai pihak. Oleh karena itu, disampaikan terima kasih kepada:

1. Allah Swt yang telah melimpahkan rahmat dan rizki-Nya serta memberi kelancaran dan kemudahan sehingga terselesaikannya proyek akhir ini.
2. Nabi Muhammad SAW yang telah membawa kita ke peradaban manusia yang lebih baik.
3. Ibu Dr. Ir. Entin Hidayah M.U.M selaku Dekan Fakultas Teknik Universitas Jember.
4. Dr. Bambang Sri Kaloko, S.T., M.T., selaku Ketua Jurusan Teknik Elektro Universitas Jember.
5. Bapak Dedy Setia Kurniawan, S.T., M.T. selaku Ketua Prodi S1 Fakultas Teknik Elektro Universitas Jember.
6. Bapak Dr. Bambang Sri Kaloko, S.T., M.T., selaku Dosen Pembimbing Utama dan Bapak Dr. Triwahju Hardianto, S.T., M.T., selaku Dosen Pembimbing Anggota yang telah membantu memberikan bimbingan dalam penyusunan penelitian ini.
7. Bapak Dedy Kurnia Setiawan, S.T., M.T., selaku dosen penguji 1 dan Bapak RB. Moch. Gozali, S.T., M.T., selaku dosen penguji 2.
8. Para dosen beserta seluruh karyawan Fakultas Teknik Universitas Jember, terima kasih atas segala dukungannya selama ini;
9. Ibunda Musringah dan Ayahanda Mudjito tercinta, yang telah mendidik, memberi kasih sayang dan mendoakan.

10. Saudaraku elektro angkatan 2012 “SATE UJ” terima kasih kalian adalah inspirasi dan penyemangatku.
11. Serta semua pihak yang turut membantu dalam penyelesaian penelitian ini.

Semoga skripsi ini dapat bermanfaat dalam mengembangkan ilmu pengetahuan khususnya untuk disiplin ilmu teknik elektro, kritik dan saran diharapkan terus mengalir untuk lebih menyempurnakan skripsi ini dan diharapkan dapat dikembangkan untuk penelitian-penelitian selanjutnya.

Jember, 30 Juni 2016

Penulis

DAFTAR ISI

	Halaman
HALAMAN JUDUL	i
HALAMAN PERSEMBAHAN	ii
HALAMAN MOTTO	iii
HALAMAN PERNYATAAN	iv
HALAMAN PEMBIMBING	v
HALAMAN PENGESAHAN	vi
ABSTRAK	vii
ABSTRACT	viii
RINGKASAN	ix
SUMMARY	x
PRAKATA	xi
DAFTAR ISI	xiii
DAFTAR GAMBAR	xv
DAFTAR TABEL	xvi
BAB 1. PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah	2
1.4 Tujuan	2
1.5 Manfaat	3
BAB 2. TINJAUAN PUSTAKA	4
2.1 Mobil Listrik	4
2.1.1 Permodelan Beban Mobil Listrik.....	5
2.2 Baterai	7
2.2.1 Kapasitas Baterai	8
2.2.2 Baterai Lead Acid	10
2.3 Neural Network (NN)	11
2.3.1 Sejarah Neural Network.....	13
2.3.2 Arsitektur Neural Network	13

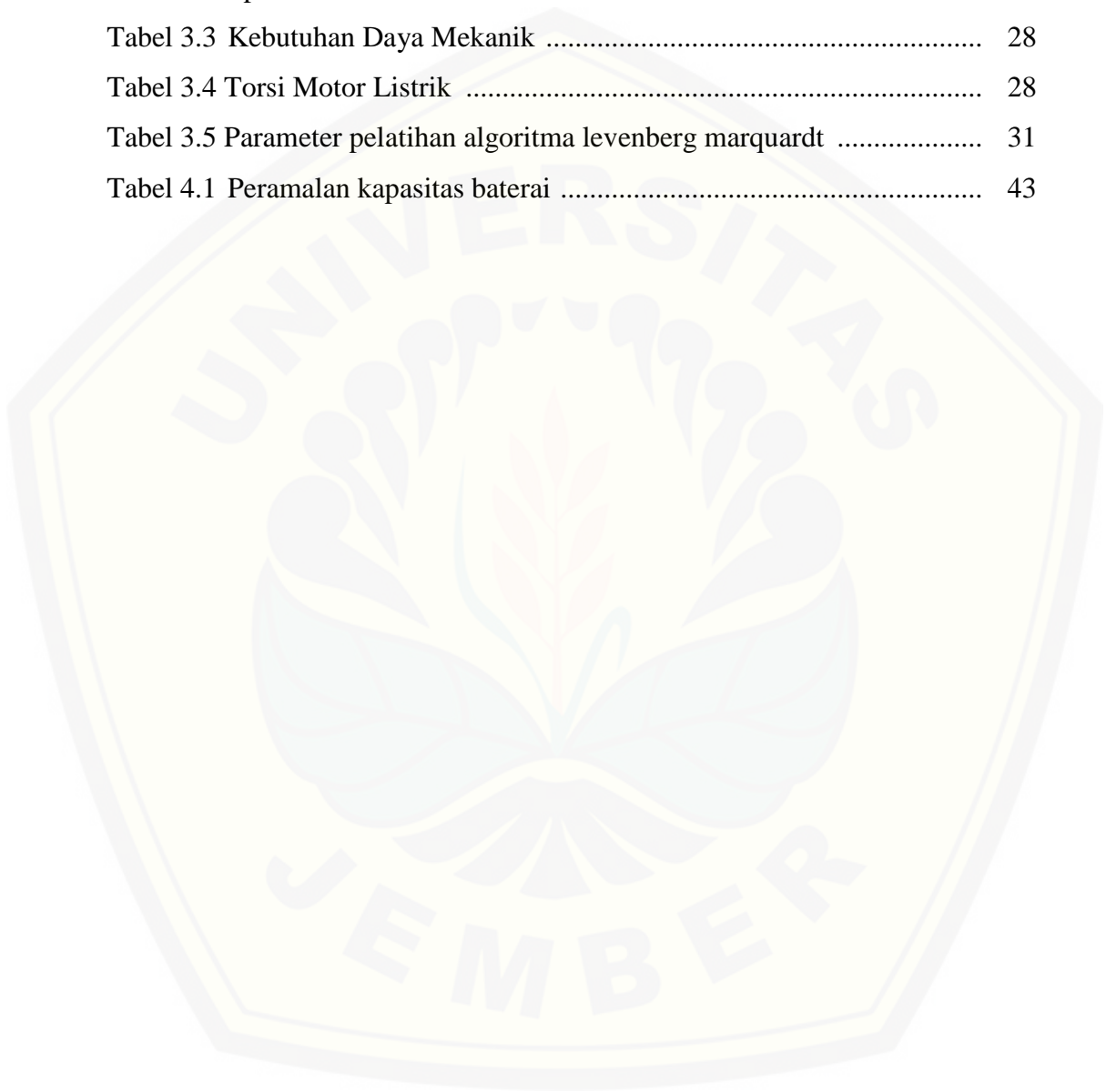
2.3.3 Fungsi Transfer	15
2.3.4 Aplikasi Neural Network	16
2.4 Backpropagation Neural Network	16
2.5 Recurrent Neural Network	20
2.6 Levenberg Marquardt.....	22
BAB 3. METODOLOGI PENELITIAN	24
3.1 Tahapan Penelitian	24
3.2 Matlab	25
3.3 Permodelan Baterai	25
3.4 Permodelan Bidirectional DC-DC Converter	26
3.5 Permodelan Beban Mobil Listrik.....	27
3.5 Backpropagation.....	28
3.6 Recurrent.....	29
3.7 Levenberg Marquardt.....	31
3.8 Permodelan Keseluruhan Mobil Listrik.....	32
3.9 Diagram Alur Penelitian	32
BAB 4. ANALISA DATA DAN PEMBAHASAN	34
4.1 Simulasi Menggunakan Backpropagation Neural Network.....	34
4.2 Simulasi Menggunakan Recurrent Neural Network.....	37
4.3 Peramalan Kapasitas Baterai	40
BAB 5. KESIMPULAN DAN SARAN	44
DAFTAR PUSTAKA	45
LAMPIRAN	47

DAFTAR GAMBAR

	Halaman
Gambar 2.1 Baterai pada kondisi discharge	8
Gambar 2.2 Baterai pada kondisi charge	8
Gambar 2.3 Salah satu baterai lead acid di pasaran	11
Gambar 2.4 Bentuk dasar neuron	12
Gambar 2.5 Arsitektur single layer network	14
Gambar 2.6 Multi layer network	15
Gambar 2.7 Struktur Recurrent neural network	15
Gambar 2.8 Struktur recurrent neural network	21
Gambar 3.1 Spesifikasi baterai lead acid	26
Gambar 3.2 Rangkaian Bidirectional DC-DC Converter.....	26
Gambar 3.3 Permodelan mobil listrik	27
Gambar 3.3 Arsitektur backpropagation, hidden layer dan bobot	29
Gambar 3.4 Arsitektur recurrent, hidden layer dan bobot.....	30
Gambar 3.5 Permodelan pengambilan data mobil listrik.....	32
Gambar 3.6 Diagram Alur keseluruhan sistem	33
Gambar 4.1 Model peramalan kapasitas baterai neural network	34
Gambar 4.2 Hasil training backpropagation	35
Gambar 4.3 Hasil performance backpropagation.....	35
Gambar 4.4 Tampilan plot training state backpropagation	36
Gambar 4.5 Tampilan plot regression backpropagation	37
Gambar 4.6 Tampilan training recurrent neural network	38
Gambar 4.7 Tampilan plot performance recurrent.....	39
Gambar 4.8 Tampilan plot training state recurrent	39
Gambar 4.9 Tampilan regression recurrent.....	40
Gambar 4.10 Grafik peramalan kapasitas baterai saat kecepatan akselerasi awal, kecepatan maksimum, dan kecepatan tanjakan	42

DAFTAR TABEL

	Halaman
Tabel 3.1 Spesifikasi Baterai	25
Tabel 3.2 Spesifikasi Mobil Listrik	27
Tabel 3.3 Kebutuhan Daya Mekanik	28
Tabel 3.4 Torsi Motor Listrik	28
Tabel 3.5 Parameter pelatihan algoritma levenberg marquardt	31
Tabel 4.1 Peramalan kapasitas baterai	43



BAB 1. PENDAHULUAN

1.1 Latar Belakang

Mobil listrik pada saat ini menjadi pembicaraan banyak orang dan sangat populer. Wai (2015:359) mengatakan, bahwa mobil listrik mempunyai beberapa keuntungan seperti efisiensi yang tinggi, tingkat pencemaran lingkungan yang rendah, tingkat kebisingan yang rendah, energinya tersedia dari berbagai sumber energi alternatif, mudah perawatannya, dan regeneratif. Mobil listrik lebih hemat energi dibandingkan dengan mobil berbahan bakar fosil konvensional.

Namun, jumlah mobil listrik yang ada masih jauh dibandingkan dengan mobil konvensional. Di Indonesia, mobil listrik masih dalam tahap pengembangan dan penelitian. Salah satu pembahasan dalam penelitian mobil listrik adalah sumber energi atau baterai. Baterai pada mobil listrik yang digunakan bermacam-macam, seperti baterai *lead acid*, *lithium ion* dan lain-lain. Jenis baterai sangat berpengaruh pada kapasitas energinya karena perbedaan karakteristik yang dimiliki oleh baterai. Peter Cubon (2015:156) mengatakan, bahwa kapasitas dari sumber energi merupakan parameter utama dari penyusun sebuah mobil listrik. Baterai pada mobil listrik mempunyai kapasitas yang terbatas sehingga harus dilakukan pengisian ulang agar mobil listrik tetap bisa berjalan jauh. Keadaan ini bila tidak segera diatasi bisa menyebabkan baterai tiba – tiba habis ditengah jalan. Tentu hal ini membuat cemas dan tidak nyaman bagi pengguna mobil listrik. Oleh karena itu diperlukan peramalan kapasitas baterai agar diketahui kapan waktunya untuk mengisi ulang baterai atau bahkan menggantinya.

Peramalan kapasitas baterai mobil listrik telah diteliti oleh Chew Kuew Wai dalam jurnalnya (Wai, 2015), namun dalam peramalannya hanya menggunakan simulasi simulink Matlab secara kisaran dinamis tanpa menggunakan kecerdasan buatan. Sedangkan penelitian yang dilakukan oleh Jie Liu dalam jurnalnya (Jie Liu, 2010), menyatakan bahwa peramalan menggunakan *neural network* dapat meningkatkan akurasi peramalan dengan baik dengan memanfaatkan perulangan

sistem. Penelitian Jie Liu menggunakan teknik optimasi *levenberg marquardt* yang digunakan untuk mengoptimasi bobot *neural network* untuk mengakomodasi kondisi sistem yang bervariasi terhadap waktu.

Penelitian ini membahas peramalan kapasitas baterai mobil listrik berbasis kecerdasan buatan *levenberg marquardt neural network*, serta membandingkan dengan sistem *recurrent neural network*.

1.2 Rumusan Masalah

Sebagaimana latar belakang yang telah dipaparkan, maka dalam penelitian ini dapat dirumuskan beberapa masalah yaitu :

1. Bagaimana merancang permodelan peramalan kapasitas baterai dan waktu habisnya baterai mobil listrik ?
2. Bagaimana perbandingan permodelan peramalan kapasitas baterai dan waktu habisnya baterai mobil listrik menggunakan *backpropagation neural network* dengan *recurrent neural network* berbasis *levenberg marquardt*?

1.3 Batasan Masalah

Berdasarkan rumusan masalah yang telah diuraikan di atas maka pembahasan dibatasi pada :

1. Baterai yang digunakan yaitu baterai *lead acid*.
2. Beban mobil listrik yang digunakan dimodelkan dengan motor DC.
3. Pengujian sistem menggunakan tiga kondisi beban, yaitu saat beban kecepatan awal, menanjak, dan kecepatan maksimum.

1.4 Tujuan

Tujuan dilakukan penelitian ini adalah sebagai berikut :

1. Mendapatkan permodelan peramalan kapasitas baterai dan waktu habisnya baterai.
2. Membandingkan sistem *backpropagation neural network* dengan *recurrent neural network* berbasis *levenberg marquardt*.

1.5 Manfaat

Manfaat dari penelitian ini adalah sebagai berikut :

1. Dapat digunakan untuk meramalkan kapasitas baterai dan waktu habisnya baterai dari sebuah mobil listrik.
2. Dapat menjadi pertimbangan dalam memilih sistem *neural network* untuk digunakan peramalan sistem.



BAB 2. TINJAUAN PUSTAKA

2.1 Mobil Listrik

Mobil listrik adalah mobil yang digerakkan dengan motor listrik, menggunakan energi listrik yang disimpan dalam baterai atau tempat penyimpanan energi lainnya. Mobil listrik sangat populer pada akhir abad ke-19 dan awal abad ke-20, tetapi kemudian popularitasnya meredup karena teknologi mesin pembakaran dalam yang semakin maju dan harga kendaraan berbahan bakar bensin yang semakin murah. Krisis energi pada tahun 1970-an dan 1980-an pernah membangkitkan sedikit minat pada mobil-mobil listrik, tetapi baru pada tahun 2000-an lah para produsen kendaraan baru menaruh perhatian yang serius pada kendaraan listrik. Hal ini disebabkan karena harga minyak yang melambung tinggi pada tahun 2000-an serta banyak masyarakat dunia yang sudah sadar akan buruknya dampak emisi gas rumah kaca. Sampai bulan November 2011, model-model listrik yang tersedia dan dijual di pasaran beberapa negara adalah Tesla Roadster, REVAi, Renault Fluence Z.E., Buddy, Mitsubishi i MiEV, Tazzari Zero, Nissan Leaf, Smart ED, Wheego Whip LiFe, Mia listrik, dan BYD e6. Nissan Leaf, dengan penjualan lebih dari 20.000 unit di seluruh dunia (sampai November 2011), dan Mitsubishi i-MiEV, dengan penjualan global lebih dari 17.000 unit (sampai Oktober 2011), adalah kedua mobil listrik paling laris di dunia (Al-Atas, 2015).

Mobil listrik memiliki beberapa kelebihan yang potensial jika dibandingkan dengan mobil bermesin pembakaran dalam biasa. Yang paling utama adalah mobil listrik tidak menghasilkan emisi kendaraan bermotor. Selain itu, mobil jenis ini juga mengurangi emisi gas rumah kaca karena tidak membutuhkan bahan bakar fosil sebagai penggerak utamanya. Pada akhirnya, ketergantungan minyak dari luar negeri pun berkurang, karena bagi beberapa negara maju seperti Amerika Serikat dan banyak negara Eropa, kenaikan harga minyak dapat memukul ekonomi mereka. Bagi negara berkembang, harga minyak yang tinggi semakin memberatkan neraca pembayaran mereka, sehingga menghambat pertumbuhan ekonomi mereka.

Mobil listrik terdiri dari tiga sub-sistem utama yaitu terdiri dari :

1. Sistem penggerak motor listrik : pengendali mobil, konverter elektronika daya, motor listrik dan transmisi.
2. Sistem baterai : baterai, Sistem manajemen baterai dan unit pengisian.
3. Sistem pembantu : pemanas/pendingin, pompa elektronika dan pembantu elektronika lainnya.

Prinsip kerja mobil listrik adalah sebagai berikut. *Input* pengendali diperoleh dari pedal akselerator dan rem. Pengendali ini menyediakan sinyal yang sesuai ke konverter daya elektronika yang mengatur aliran daya antara motor listrik dan baterai. Motor juga memainkan peran generator, yang mengkonversi energi pengereman menjadi elektron dan mengisi baterai. Unit manajemen energi, bekerja sama dengan pengendali mobil, mengendalikan pengereman regenerasi dan rekoveri energi. Motor listrik menghasilkan torsi yang besar dari keadaan mobil berhenti. Hal ini menghasilkan kinerja yang sangat bagus. Percepatan dan daya mobil listrik jauh melampaui mobil mesin pembakaran dalam (Al-Atas, 2015).

2.1.1 Permodelan Beban Mobil Listrik

Kinerja dari sebuah mobil listrik biasanya digambarkan menjadi 3 model kondisi yang dirumuskan sebagai berikut (Agarwal & Rajan, 2013).

1. *Maximum Cruising Speed* (Kecepatan maksimum menjelajah)

Kecepatan maksimum kendaraan didefinisikan sebagai kecepatan menjelajah konstan dimana kendaraan dapat menopang dengan beban penuh (daya penuh dari motor) di jalan datar.

Ketika kendaraan melaju pada kecepatan maksimum, permintaan daya mekanik yang dibutuhkan oleh mobil listrik dirumuskan sebagai berikut:

$$P = (f_{rr} m g + \rho A C_d V_{max}^2) V_{max} \dots\dots\dots (1)$$

- Keterangan :
- P = Daya mekanik (watt)
 - m = Massa kendaraan (kg)
 - g = Percepatan gravitasi (9,81 m/s²)
 - F_{rr} = Hambatan putaran ban (*rolling resistance coefficient*)
 - ρ = Kerapatan udara / *Air Density* (1,202 kg/m³)

- C_d = Koefisien aerodinamis (*Aerodynamics Drag Coefficient*)
 A = Luasan area depan kendaraan (*Front Area of the Vehicle*)
 V = Kecepatan kendaraan (m/s)

2. *Gradeability* (Tanjakan)

Permintaan daya mekanik yang dibutuhkan pada mobil listrik selama tanjakan diperoleh dari rumus sebagai berikut:

$$P = m \cdot g \cdot \beta \cdot V \dots\dots\dots (2)$$

- Keterangan :
- P = Daya mekanik (watt)
 - m = Massa kendaraan (kg)
 - β = Koefisien tanjakan (*gradeability coefficient*)
 - g = Percepatan gravitasi (9,81 m/s²)
 - V = Kecepatan kendaraan (m/s)

3. *Initial Acceleration* (Akselerasi awal)

Saat akselerasi awal, daya mekanik yang diminta oleh kendaraan dapat dirumuskan sebagai berikut:

$$P = (m g f_{rr} + \rho C_d A V^2 + m) V \dots\dots\dots (3)$$

- Keterangan :
- P = Daya mekanik (watt)
 - m = Massa kendaraan (kg)
 - g = Percepatan gravitasi (9,81 m/s²)
 - F_{rr} = Hambatan putaran ban (*rolling resistance coefficient*)
 - ρ = Kerapatan udara / *Air Density* (1,202 kg/m³)
 - C_d = Koefisien aerodinamis (*Aerodynamics Drag Coefficient*)
 - A = Luasan area depan kendaraan (*Front Area of the Vehicle*)
 - V = Kecepatan kendaraan (m/s)

Pada kendaraan bermotor, hubungan hubungan torsi terhadap daya dirumuskan dengan persamaan berikut:

$$P = \tau \times \omega \dots\dots\dots (4)$$

- Keterangan :
- P = Daya (watt)
 - τ = Torsi (Nm)
 - ω = Kecepatan sudut (radian per detik)

Sedangkan untuk motor listrik, rumusan untuk kecepatan sudut adalah :

$$\omega = 2 \pi n / 60 \dots\dots\dots(5)$$

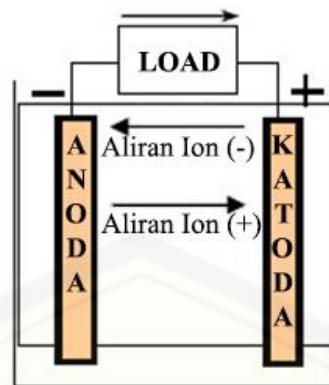
Keterangan : n = kecepatan putaran motor (rpm)

2.2 Baterai

Baterai adalah sebuah alat yang dapat menyimpan energi listrik untuk digunakan sesuai dengan keperluannya sewaktu-waktu serta dapat dengan mudah dipindahkan dari satu tempat ketempat lainnya. Berdasarkan jenisnya, baterai dapat dibagi menjadi baterai primer (*primary battery*) dan baterai sekunder (*secondary battery*). Baterai primer adalah baterai yang tidak dapat diisi muatan listrik kembali (*charge*) setelah habis digunakan. Beberapa contoh baterai primer adalah baterai Seng – Karbon (Baterai Kering), baterai Alkalin dan baterai Merkuri. Sedangkan baterai sekunder dapat diisi kembali oleh muatan listrik (*rechargeable*). Kemampuan diisi ulang baterai sekunder bervariasi antara 100-500 kali (Satu siklus adalah satu kali pengisian dan pengosongan). Beberapa contoh baterai sekunder adalah baterai asam timbal (*lead acid*), baterai Ni-Cd, baterai Ni-MH, dan baterai Litium-Ion (Hudaya, 2011).

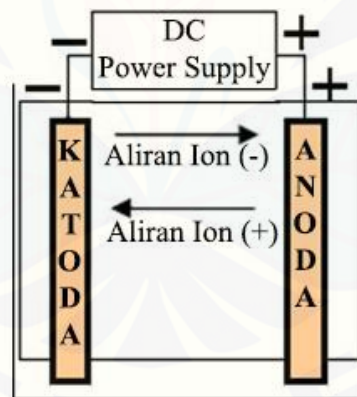
Listrik yang dihasilkan oleh sebuah baterai muncul akibat adanya perbedaan potensial energi listrik kedua buah elektrodanya. Perbedaan potensial ini dikenal dengan potensial sel atau gaya gerak listrik (ggl). Untuk melengkapi reaksi dalam sebuah baterai dibutuhkan media transfer muatan dan sirkuit luar sebagai jalur alir listrik. Prinsip kerja baterai yaitu:

1. Proses discharge pada sel berlangsung menurut skema Gambar 2.1. Bila sel dihubungkan dengan beban maka elektron mengalir dari anoda melalui beban ke katoda, kemudian ion-ion negatif mengalir ke anoda dan ion-ion positif mengalir ke katoda.



Gambar 2.1 Baterai pada kondisi *discharge* (Putra, 2015).

2. Pada proses pengisian menurut skema Gambar 2.2 adalah bila sel dihubungkan dengan power supply maka elektroda positif menjadi anoda dan elektroda negatif menjadi katoda dan proses kimia yang terjadi sebagai berikut:
 - a. Aliran elektron terbalik, mengalir dari anoda melalui power supply ke katoda.
 - b. Ion-ion negatif mengalir dari katoda ke anoda.
 - c. Ion-ion positif mengalir dari anoda ke katoda.



Gambar 2.2 Baterai pada kondisi *charge* (Putra, 2015).

2.2.1 Kapasitas Baterai

Kapasitas baterai adalah ukuran muatan yang disimpan suatu baterai yang ditentukan oleh masa aktif material didalamnya. Kapasitas menggambarkan sejumlah energi maksimum yang dapat dikeluarkan dari sebuah baterai dengan kondisi khusus tertentu. Tetapi kemampuan penyimpanan baterai dapat berbeda dari kapasitas nominalnya, diantaranya karena kapasitas baterai bergantung pada

umur dan keadaan baterai, parameter charging dan discharging, dan temperatur. Satuan dari kapasitas baterai ini sering dinyatakan dalam Ampere hours (walaupun kadang dalam Wh), ditentukan sebagai waktu dalam jam yang dibutuhkan baterai untuk secara kontinu mengalirkan arus atau nilai discharge pada tegangan nominal baterai. Satuan Ah sering digunakan ketika tegangan baterai bervariasi selama siklus charging atau discharging. Kapasitas Wh dapat diperkirakan dengan mengalikan kapasitas Ah dengan tegangan nominal baterai. Misalnya, baterai 12 V dengan kapasitas 70 Ah memberikan energi yang tersimpan sekitar $70 \text{ Ah} \times 12 \text{ V} = 840 \text{ Wh}$ atau 0,84 KWh.

Temperatur dari baterai berpengaruh terhadap energi yang dapat dikeluarkan dari baterai. Pada temperatur yang lebih tinggi akan memiliki kapasitas yang lebih besar dari pada temperatur yang rendah. Tetapi meningkatkan temperatur dengan disengaja memiliki dampak negatif, karena akan mengurangi lifetime dari baterai. Umur dan keadaan baterai juga berpengaruh terhadap kapasitas baterai. Meskipun baterai dipergunakan secara benar sesuai aturan manufaktur, semakin lama kapasitas baterai tersebut dapat berkurang. Keadaan dari baterai juga berpengaruh terhadap kapasitas baterai. Misalnya, jika baterai pernah discharge dibawah maksimum DOD, maka kapasitas baterai dapat berkurang (S. Anda, 2014).

Xiyun Yang dalam jurnalnya (Yang, 2013) menjelaskan ada beberapa metode yang dapat digunakan untuk mengetahui kondisi kapasitas baterai, diantaranya adalah :

1. Metode densitas cairan listrik, tetapi metode ini tidak cocok untuk *Valve Regulated Lead Acid Battery* (VRLA). Metode ini menggunakan pengukuran berat jenis cairan pada baterai untuk mengetahui sisa kapasitas suatu baterai.
2. Metode *Open circuit voltage* (OCV). Metode ini cocok untuk baterai baru, tetapi ketika baterai digunakan setelah waktu yang lama, dan kapasitas baterai turun, perubahan tegangan rangkaian terbuka tidak dapat mencerminkan keadaan sebenarnya dari kapasitas.
3. Metode *discharge*, kurva baterai diperoleh dengan eksperimen *discharge* dengan keakuratan dapat menggambarkan kinerja baterai. Namun, tes

discharge tidak dapat sering dilakukan karena akan mempengaruhi kehidupan pelayanan baterai.

4. Metode resistensi internal, kurva resistansi – kapasitas baterai harus diukur dalam metode ini. Proses pengukuran rumit, sehingga keumuman dari metode ini adalah lemah.

2.2.2 Baterai Lead Acid

Ada berbagai jenis baterai sekunder yang saat ini secara komersial telah diproduksi. Masing – masing jenis baterai tersebut memiliki karakteristik yang berbeda-beda. Parameter-parameter utama untuk membedakan baterai satu dengan yang lainnya diantaranya yaitu kerapatan energi (Wh/dm^3), kerapatan energi jenis (Wh/kg), kerapatan daya (W/kg), kapasitas (Ah/g), dan banyaknya siklus (*cycles*). Jenis-jenis baterai sekunder yang sedang dikembangkan dan yang telah ada dipasaran adalah baterai *lead acid*, baterai nikel kadmium, baterai nikel metal hidrida dan baterai lithium (Hudaya, 2011).

Baterai asam timbal adalah baterai sekunder yang paling banyak dikembangkan di dunia. Baterai jenis ini pertama kali ditemukan oleh Gaston Planté pada tahun 1859. Baterai asam timbal banyak digunakan untuk aplikasi otomotif, sehingga dinamakan juga sebagai baterai SLI (Starting, Lightning and Ignition). Penggunaan masal baterai jenis SLI dikarenakan material untuk membuat baterai tersebut cukup murah namun baterai memiliki performa cukup baik. Akan tetapi, untuk aplikasi yang membutuhkan daya yang lebih tinggi dengan waktu yang relatif lama, baterai SLI tidak dapat digunakan. Hal ini karena baterai asam timbal hanya memiliki kedalaman pelepasan muatan listrik (Depth of Discharge – DOD) sebesar 50 % saja.

Baterai jenis asam timbal tersusun atas timbal dioksida sebagai katoda, sepon logam timbal sebagai anoda dan asam sulfat sebagai elektrolitnya. Setiap sel memiliki tegangan sebesar 2 Volt. Keuntungan penggunaan baterai jenis asam timbal diantaranya adalah kuat, murah, handal, toleran terhadap kelebihan pengisian, impedansi internal yang rendah, dan banyaknya perusahaan pembuat baterai jenis ini di berbagai belahan dunia. Sedangkan kekurangan dari baterai jenis

SLI ini diantaranya adalah sangat berat, memiliki efisiensi energi yang rendah (sekitar 70%), berbahaya jika kelebihan panas pada saat pengisian, memiliki waktu siklus yang rendah (300-500 siklus), dan materialnya berbahaya bagi lingkungan.

Dengan memodifikasi elektroda dan susunan internal baterai, jenis-jenis baterai asam timbal dapat dibedakan menjadi baterai timbal-kalsium, baterai timbal-antimoni dan baterai asam timbal yang elektrolitnya terpisah (*sealed lead acid* – SLA).



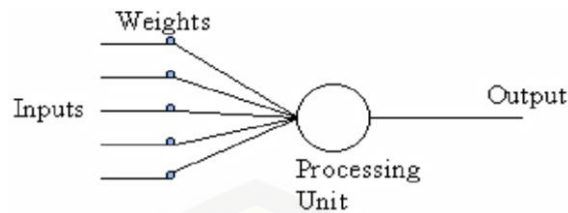
Gambar 2.3 Salah satu baterai *lead acid* di pasaran (Hudaya, 2011).

2.3 Neural Network (NN)

Secara umum *Neural Network* (NN) adalah jaringan dari sekelompok unit pemroses kecil yang dimodelkan berdasarkan jaringan syaraf manusia. NN ini merupakan sistem adaptif yang dapat merubah strukturnya untuk memecahkan masalah berdasarkan informasi eksternal maupun internal yang mengalir melalui jaringan tersebut. Secara sederhana NN adalah sebuah alat pemodelan data statistik non-linear. NN dapat digunakan untuk memodelkan hubungan yang kompleks antara *input* dan *output* untuk menemukan pola-pola pada data.

Secara mendasar, sistem pembelajaran merupakan proses penambahan pengetahuan pada NN yang sifatnya kontinuitas sehingga pada saat digunakan pengetahuan tersebut akan dieksploitasikan secara maksimal dalam mengenali suatu objek. *Neuron* adalah bagian dasar dari pemrosesan suatu *Neural Network* (Noviana, 2008).

Dibawah ini merupakan bentuk dasar dari suatu *neuron*.



Gambar 2.4 Bentuk dasar *neuron* (Noviana, 2008)

Keterangan Gambar 2.4 di atas adalah sebagai berikut:

1. *Input* = merupakan masukan yang digunakan baik saat pembelajaran maupun dalam mengenali suatu objek.
2. *Weight* = beban yang selalu berubah setiap kali diberikan *input* sebagai proses pembelajaran.
3. *Processing Unit* = merupakan tempat berlangsungnya proses pengenalan suatu objek berdasarkan pembebanan yang diberikan.
4. *Output* = keluaran dari hasil pengenalan suatu objek.

Beberapa keuntungan penggunaan *neural network* dalam suatu sistem adalah:

1. Perangkat mampu untuk mengenali suatu objek secara non-linier.
2. Mempermudah pemetaan *input* menjadi suatu hasil tanpa mengetahui proses sebenarnya.
3. Mampu melakukan pengadaptasian terhadap pengenalan suatu objek.
4. Perangkat yang memiliki toleransi terhadap suatu kesalahan dalam pengenalan suatu objek.
5. *Neural network* mampu diimplementasikan pada suatu *hardware* atau perangkat keras.

Suatu sistem *neural network* mempunyai beberapa parameter yang menentukan keberhasilan sistem, yaitu:

1. Pola hubungan antar *neuron* (disebut arsitektur jaringan).
2. Metode untuk menentukan bobot penghubung (disebut metode *training/learning/algorithm*).
3. Fungsi aktivasi.

2.3.1 Sejarah Neural Network

Neural network sederhana pertama kali diperkenalkan oleh McCulloch dan Pitts di tahun 1943. McCulloch dan Pitts menyimpulkan bahwa kombinasi beberapa *neuron* sederhana menjadi sebuah sistem neural akan meningkatkan kemampuan komputasinya. Bobot dalam jaringan yang diusulkan oleh McCulloch dan Pitts diatur untuk melakukan fungsi logika sederhana. Fungsi aktivasi yang dipakai adalah fungsi *threshold*.

Tahun 1958, Rosenblatt memperkenalkan dan mulai mengembangkan model jaringan yang disebut *perceptron*. Metode pelatihan diperkenalkan untuk mengoptimalkan hasil *iterasi*-nya.

Widrow dan Hoff (1960) mengembangkan *perceptron* dengan memperkenalkan aturan pelatihan jaringan, yang dikenal sebagai aturan *delta* (atau sering disebut kuadrat rata-rata terkecil). Aturan ini akan mengubah bobot *perceptron* apabila keluaran yang dihasilkan tidak sesuai dengan target yang diinginkan.

Apa yang dilakukan peneliti terdahulu hanya menggunakan jaringan dengan layer tunggal (*single layer*). Rumelhart (1986) mengembangkan *perceptron* menjadi *backpropagation*, yang memungkinkan jaringan diproses melalui beberapa *layer*.

Selain itu, beberapa model *neural network* lain juga dikembangkan oleh Kohonen (1972), Hopfield (1982), dll.

Pengembangan yang ramai dibicarakan sejak tahun 1990an adalah aplikasi model-model *neural network* untuk menyelesaikan berbagai masalah di dunia. (Siregar, 2013)

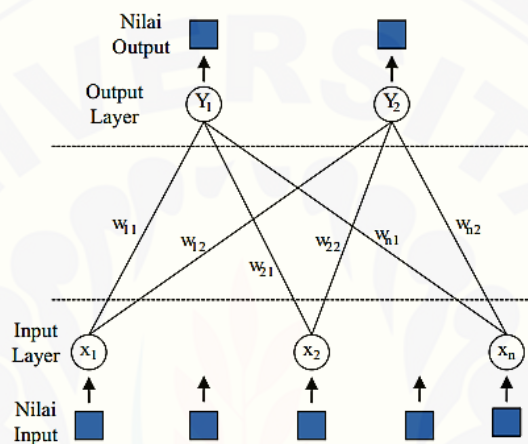
2.3.2 Arsitektur Neural Network

Beberapa arsitektur jaringan yang sering dipakai dalam neural network antara lain:

1. Jaringan Layer Tunggal (*single layer network*)

Jaringan dengan lapisan tunggal terdiri dari 1 *layer input* dan 1 *layer output*. Setiap *neuron* yang terdapat di dalam *layer input* selalu terhubung dengan setiap

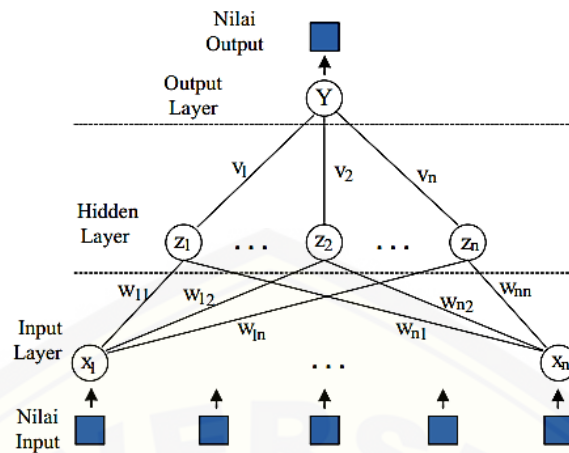
neuron yang terdapat pada *layer output*. Jaringan ini hanya menerima *input* kemudian secara langsung akan mengolahnya menjadi *output* tanpa harus melalui *hidden layer*. Contoh algoritma JST yang menggunakan metode ini yaitu *adaline* dan *perceptron*. Pada Gambar 2.5 diperlihatkan arsitektur *single layer network* dengan n buah *input layer* (X_1, X_2, \dots, X_n) dan dua buah *output layer* (Y_1, Y_2) dengan bobot masing-masing yang menghubungkan antara *input layer* dengan *output layer* ($W_{11}, W_{12}, W_{21}, W_{22}, W_{n1}, W_{n2}$).



Gambar 2.5 Arsitektur *single layer network* (Siregar, 2013)

2. Jaringan Layar Jamak (*Multi Layer Network*)

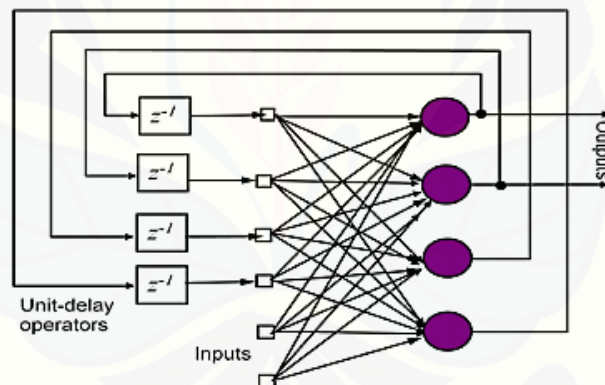
Jaringan dengan lapisan jamak memiliki ciri khas tertentu yaitu memiliki 3 jenis *layer* yaitu *layer input*, *layer output*, dan *layer* tersembunyi. Jaringan dengan banyak lapisan ini dapat menyelesaikan permasalahan yang kompleks dibandingkan jaringan dengan lapisan tunggal. Namun, proses pelatihan sering membutuhkan waktu cenderung lama. Contoh algoritma *neural network* yang menggunakan metode ini yaitu *backpropagation*. Pada Gambar 2.6 diperlihatkan arsitektur *multi layer network* dengan n buah *input layer* (X_1, \dots, X_n), n buah *hidden layer* (Z_1, Z_2, \dots, Z_n) dan sebuah *output layer* (Y) dengan bobot yang menghubungkan *input layer* dengan *hidden layer* ($W_{11}, W_{12}, W_{1n}, W_{n1}, W_{n2}, W_{nn}$) dan bobot yang menghubungkan *hidden layer* dengan *output layer* (V_1, V_2, \dots, V_n).



Gambar 2.6 *Multi layer network* (Siregar, 2013)

3. *Recurrent Network*

Recurrent network memiliki minimal paling sedikit ada satu *feedback loop* didalamnya. *Feedback loop* mempengaruhi kemampuan belajar dan kinerja jaringan.



Gambar 2.7 Struktur *recurrent network* (Siregar, 2013)

2.3.3 Fungsi Transfer

Karakter dari *neural network* tergantung atas bobot dan fungsi *input – output* (fungsi transfer) yang mempunyai ciri tertentu untuk setiap unit. Fungsi ini terdiri dari 3 kategori yaitu :

1. *Linear Units*, Aktifitas *output* adalah sebanding dengan jumlah bobot *output*.
2. *Threshold Units*, *Output* diatur satu dari dua tingkatan tergantung dari apakah jumlah *input* adalah lebih besar atau lebih kecil dari nilai ambang.

3. *Sigmoid Units, Output* terus menerus berubah-ubah tetapi tidak berbentuk linear. Unit ini mengandung kesamaan yang lebih besar dari sel syaraf sebenarnya dibandingkan dengan linear dan threshold unit, namun ketiganya harus dipertimbangkan dengan perkiraan kasar.

Untuk membuat *neural network* agar dapat melakukan beberapa kerja khusus, harus dipilih unit-unit yang akan dihubungkan antara satu dengan yang lain dan harus bisa mengatur bobot dari hubungan tersebut secara tepat. Hubungan tersebut akan menentukan apakah mungkin suatu unit mempengaruhi unit yang lain. Bobot menentukan kekuatan dari pengaruh tersebut (Nasution, 2009).

2.3.4 Aplikasi Neural Network

Saat ini *neural network* banyak dipakai dalam berbagai sistem. Beberapa aplikasi neural network adalah sebagai berikut:

1. Pengenalan Pola (*Pattern Recognition*)

Neural network dapat dipakai untuk mengenali pola (misal huruf, angka, suara atau tanda tangan) yang sudah sedikit berubah. Hal ini mirip dengan otak manusia yang masih mampu mengenali orang yang sudah beberapa waktu tidak dijumpainya (mungkin wajah atau bentuk tubuhnya yang sudah sedikit berubah).

2. *Signal Processing*

Neural network (model *Adaline*) dapat dipakai untuk menekan *noise* dalam saluran telpon.

3. Peramalan

Neural network juga dapat dipakai untuk meramalkan apa yang akan terjadi di masa yang akan datang berdasarkan pola kejadian yang ada di masa yang lampau. Peramalan ini dapat dilakukan mengingat kemampuan *neural network* dapat membuat generalisasi dari apa yang sudah ada sebelumnya. *Neural network* juga dapat menyelesaikan masalah dalam bidang kontrol dan kedokteran. (Siregar, 2013)

2.4 Backpropagation Neural Network

Backpropagation merupakan salah satu dari beberapa metode yang digunakan dalam *neural network* dan yang paling sering digunakan dalam berbagai

bidang aplikasi, seperti pengenalan pola, peramalan dan optimisasi. Hal ini dikarenakan metode ini menggunakan pembelajaran yang terbimbing. Pola masukan dan target diberikan sebagai pasangan data. Bobot-bobot awal dilatih dengan melalui tahap maju untuk mendapatkan *error* keluaran yang selanjutnya *error* ini digunakan dengan tahap mundur untuk memperoleh nilai bobot yang sesuai agar dapat memperkecil nilai *error* sehingga target keluaran yang dikehendaki tercapai.

Tujuan dari model ini adalah untuk mendapatkan keseimbangan antara kemampuan jaringan mengenali pola yang digunakan selama proses pelatihan berlangsung serta kemampuan jaringan memberikan respon yang benar terhadap pola masukan yang berbeda dengan pola masukan selama pelatihan.

Algoritma *backpropagation* terdiri dari dua bagian antara lain sebagai berikut: (Oktaorora, 2011)

1. Algoritma Pelatihan *Backpropagation* (*Training Proccess*)

Di dalam proses pelatihan *backpropagation* terdapat tiga tahap. Tahap pertama ialah tahap maju (feed-forward). Pada tahap ini seluruh proses awal inialisasi bobot-bobot *input* dilakukan. Pada tahap ini juga ditentukan angka pembelajaran (α), nilai toleransi *error* dan jumlah *epoch* (siklus setiap pola pelatihan) yang diperlukan selama proses komputasi berlangsung.

Setelah semua proses inialisasi dilakukan, maka langkah selanjutnya ialah proses maju. Setiap unit masukan x_i akan mengirimkan sinyal masukan ke lapisan tersembunyi.

Setelah dihitung dengan menggunakan fungsi aktivasi maka keluarannya akan dikirimkan ke lapisan di atasnya, yaitu lapisan *output*. Setelah nilai keluaran (y_k) diperoleh, maka dibandingkan dengan target keluaran sebenarnya (tk). Selisih $y_k - tk$ disebut dengan *error* (δk). Jika nilai *error* lebih kecil atau sama dengan dari nilai ambang maka proses iterasi dihentikan, tetapi jika tidak maka nilai *error* tersebut digunakan untuk memodifikasi bobot-bobot untuk mengoreksi kesalahan yang terjadi.

Tahap kedua adalah tahap mundur atau *backpropagation*. Pada tahap ini, nilai *error* (δk) yang diperoleh pada di lapisan *output* digunakan untuk mengoreksi

bobot-bobot yang ada pada lapisan tersembunyi yang berhubungan langsung dengan lapisan *output*. Setelah itu nilai *error* (δ_j) di setiap unit pada lapisan tersembunyi juga dihitung untuk mengoreksi bobot-bobot yang menghubungkan lapisan *input* dengan lapisan tersembunyi.

Tahap ketiga adalah tahap pengoreksian bobot. Setelah seluruh bobot pada lapisan *input* dan lapisan tersembunyi dimodifikasi sesuai dengan besar faktor *error*nya, maka ketiga fase ini diulang secara terus menerus sampai kondisi berhenti dipenuhi. Kondisi berhenti yang dimaksud adalah jika jumlah *epoch* yang ditetapkan tercapai atau jika nilai *error* jaringan telah sama dengan atau lebih kecil dari nilai toleransi *error* yang ditetapkan sebelumnya. Pada tahap pelatihan, jaringan diharapkan dapat melatih seluruh data pelatihan yang diberikan untuk mendapatkan bobot akhir jaringan yang akan digunakan pada tahap pengujian. Struktur algoritma pelatihan *Backpropagation* antara lain:

1. Inisialisasi bobot-bobot

Mentukan angka pembelajaran (α). Tentukan pula nilai toleransi error yang diinginkan dan set maksimal *epoch* jika ingin membatasi jumlah *epoch* yang digunakan.

2. Selama kondisi berhenti tidak terpenuhi, lakukan langkah ke-2 sampai ke-9.

3. Setiap pasangan pola pelatihan, lakukan langkah ke-3 sampai ke-8.

Tahap maju (Feedforward)

4. Tiap-tiap unit *input* ($x_i, i = 1, 2, 3, \dots, o$) menerima sinyal *input* dan meneruskan sinyal tersebut ke tiap-tiap unit pada lapisan tersembunyi.

5. Tiap-tiap unit di lapisan tersembunyi ($z_j, j = 1, 2, 3, \dots, p$) menjumlahkan sinyal-sinyal *input* yang berbobot, yaitu:

$$z_{net\ j} = v_{j0} + \sum_{i=1}^n x_i v_{ji} \dots\dots\dots (6)$$

Keterangan : x : input

z : lapisan tersembunyi

Fungsi aktivasi untuk menghitung sinyal *output*, yaitu:

$$z_j = v_{j0} + \sum_{i=1}^n x_i v_{ji} \dots\dots\dots (7)$$

dan mengirimkan sinyal tersebut ke semua unit pada lapisan di atasnya (lapisan *output*).

6. Tiap-tiap unit di lapisan *output* ($y_k, k = 1, 2, 3, \dots, m$) menjumlahkan sinyal *input* yang berbobot, yaitu:

$$y_{net\ k} = w_{k0} + \sum_{j=1}^p z_j w_{kj} \dots\dots\dots (8)$$

Fungsi aktivasi untuk menghitung sinyal *output*, yaitu:

$$y_k = w_{k0} + \sum_{j=1}^p z_j w_{kj} \dots\dots\dots (9)$$

- Keterangan :
- x : input
 - y : output
 - z : lapisan tersembunyi
 - w : bobot

Tahap mundur (Backpropagation)

7. Tiap-tiap unit *output* y_k menerima pola target t_k untuk menghitung error (δ_k), yaitu:

$$\delta_k = (t_k - y_k) f'(y_{netk}) = (t_k - y_k) y_k (1 - y_k) \dots\dots\dots (10)$$

Kemudian menghitung nilai koreksi bobot yang nantinya digunakan untuk memperbaiki nilai bobot antara lapisan tersembunyi dan lapisan *output* (w_{jk}), yaitu:

$$\Delta w_{jk} = \alpha \delta_k z_j \dots\dots\dots (11)$$

menghitung juga koreksi bias yang digunakan untuk memperbaiki nilai bias antara lapisan tersembunyi dan lapisan *output* (w_{k0}), yaitu:

$$\Delta w_{k0} = \alpha \delta_k \dots\dots\dots (12)$$

8. Tiap-tiap unit pada lapisan tersembunyi ($z_j, j = 1, 2, 3, \dots, p$) menjumlahkan sinyal-sinyal *input* dari lapisan *output*, yaitu:

$$\delta_{net\ j} = \sum_{k=1}^n \delta_k w_{jk} \dots\dots\dots (13)$$

mengalikan nilai ini dengan fungsi aktivasi untuk menghitung *error* pada lapisan tersembunyi (δ_j), yaitu:

$$\delta_j = \delta_{net\ j} f'(z_{net\ j}) = \delta_{net\ j} z_j (1 - z_j) \dots\dots\dots (14)$$

Kemudian hitung koreksi bobot untuk memperbaiki nilai bobot antara lapisan *input* dan lapisan tersembunyi (v_{ji}), yaitu:

$$\Delta v_{ji} = \alpha \delta_j x_i \dots\dots\dots (15)$$

Kemudian menghitung koreksi bias untuk memperbaiki nilai bobot antara lapisan *input* dan lapisan tersembunyi (v_{j0}), yaitu:

$$\Delta v_{j0} = \alpha \delta_j \dots\dots\dots (16)$$

Tahap pengoreksian bobot

9. Tiap-tiap unit keluaran ($y_k, k = 1, 2, 3, \dots, m$) memperbaiki bobot dan bias, yaitu.

$$w_{kj}(\text{baru}) = w_{kj}(\text{lama}) + \Delta w_{kj}, (k = 1, 2, \dots, m; j = 0, 1, \dots, p) \dots\dots (17)$$

Tiap-tiap unit tersembunyi memperbaiki bobot dan bias, yaitu.

$$v_{ji}(\text{baru}) = v_{ji}(\text{lama}) + \Delta v_{ji}, (j = 1, 2, \dots, p; i = 0, 1, \dots, n) \dots\dots (18)$$

10. Tes kondisi berhenti

b). Algoritma Pengujian *Backpropagation* (*Testing Process*)

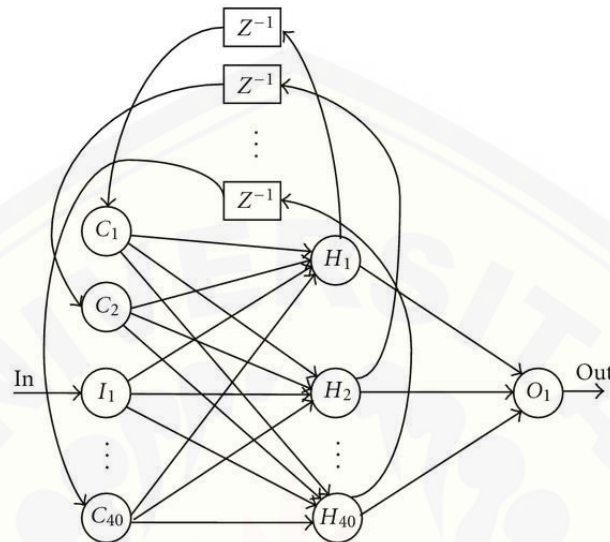
Setelah proses pelatihan, *backpropagation* dapat digunakan untuk proses pengujian jaringan. Pada proses pengujian, tahap yang dilakukan hanya sampai tahap maju saja, tidak ada tahap mundur apalagi tahap modifikasi bobot.

Seluruh bobot *input* diambil dari nilai bobot terakhir yang diperoleh dari proses pelatihan. Pada tahap pengujian ini, jaringan diharapkan dapat mengenali pola berdasarkan data baru yang diberikan (generalisasi).

2.5 Recurrent Neural Network (RNN)

Recurrent Neural Network adalah *neural network* dengan fasilitas umpan balik menuju neuron itu sendiri maupun neuron yang lain, sehingga aliran informasi dari masukan mempunyai arah jamak (Aribowo dalam Ammar, 2014). RNN adalah jaringan yang mempunyai minimal satu *feedback loop*. *Feedback loop* mempengaruhi kemampuan belajar dan kinerja jaringan. RNN memiliki kemampuan penggambaran yang sangat bagus dan dapat mengatasi kelemahan *feedforward* (Soelaiman & Rifa'i dalam Ammar, 2014). Keluaran RNN tidak hanya bergantung pada masukan saat itu saja, tetapi juga tergantung pada kondisi masukan *neural network* untuk waktu lampau. Kondisi ini dimaksudkan untuk menampung kejadian lampau diikutkan pada proses komputasi. Hal ini penting untuk masalah yang cukup rumit, dan tanggapan keluaran *neural network* berkaitan dengan variasi

waktu (*time-varying*), sehingga *neural network* memiliki kepekaan terhadap waktu dengan memori kondisi lampau (Ammar, 2014). Struktur *recurrent neural network* dapat dilihat pada Gambar 2.8 dibawah ini.



Gambar 2.8 Struktur *recurrent neural network* (Jaroslaw S., 2011).

Keluaran *Recurrent Neural Network* dapat dihitung dengan persamaan sebagai berikut:

$$y(k) = x(k + 1) - Ax(k) - V_1^0 \sigma[W_1^0 x(k)] - V_2^0 \phi[W_2^0 x(k)]u(k) \dots (19)$$

dimana :

- $x(k) \in n$ adalah internal state vector
- $A \in \mathfrak{R}^{n \times n}$ adalah matrik tetap
- $x(k) \in \mathfrak{R}^n$ adalah state vector
- $u(k) \in \mathfrak{R}^m$ adalah vektor masukan, Nilai $u(k)$ dan $x(k)$ diketahui
- $W_1^0 \in \mathfrak{R}^{n \times n}$ adalah bobot pada lapisan hidden
- $V_1^0 \in \mathfrak{R}^{n \times n}$ adalah bobot pada lapisan *output*
- $\hat{x}(k+1)$ adalah nilai dari state vektor hasil perkiraan pada iterasi ke $k+1$

Fungsi aktivasi yang digunakan pada jaringan adalah

$$T_m[\dots] \sigma = \sigma, \sigma \in \mathfrak{R}^m \text{ dan } \phi \in \mathfrak{R}^{m \times m} \dots (20)$$

Nilai σ dan ϕ dihitung dengan menggunakan fungsi sigmoid (Soelaiman & Rifa'i dalam Ammar, 2014).

2.6 Levenberg Marquardt

Algoritma *levenberg marquardt* merupakan pengembangan dari algoritma *error back propagation*. Algoritma ini dibangun untuk mengatasi beberapa kekurangan yang ada pada algoritma *error back propagation* dengan memanfaatkan teknik optimisasi numerik standar yaitu menggunakan pendekatan matriks Jacobian. Tujuan dari *levenberg marquardt* adalah meminimalkan total error. (Oktaorora, 2011)

Algoritme *levenberg marquardt* adalah salah satu solusi untuk meningkatkan kinerja algoritme *backpropagation* karena algoritme *levenberg marquardt* dirancang secara khusus untuk meminimalisasi galat jumlah kuadrat. Algoritme ini akan memperbarui bobot dengan cara memperkecil nilai galat dari selisih bobot baru dengan bobot lama. Namun, pada saat yang bersamaan, algoritme ini akan mempertahankan *step size* agar tidak menjadi besar. (Kurniawan, 2012)

Persamaan matematika untuk menentukan bobot baru pada algoritma *levenberg marquardt* ialah sebagai berikut:

$$W_{\text{new}} = W_{\text{old}} - (Z^T Z)^{-1} Z^T \epsilon (W_{\text{old}}) \dots\dots\dots (21)$$

dengan

W_{new} : bobot baru

W_{old} : bobot lama

ϵ : vektor galat

Z : matriks yang anggotanya adalah hasil diferensiasi galat terhadap bobotnya.

Algoritme Levenberg Marquardt memiliki jumlah iterasi yang lebih baik dibandingkan dengan algoritme optimasi *backpropagation* yang lain. Namun, algoritme ini membutuhkan memori yang lebih banyak dibandingkan algoritme sejenisnya (Kurniawan, 2012).

Beberapa struktur algoritma *levenberg marquardt* menurut Oktaorora (2011) :

1. Inisialisasi bobot dan bias dengan bilangan acak, *epoch* maksimum, dan minimal goal (performance yang dihitung dengan MSE).
2. Menentukan parameter yang dibutuhkan, antara lain :
 - a. Parameter *levenberg marquardt* yang nilainya harus lebih besar dari nol.

- b. Parameter faktor Beta (β) yang digunakan sebagai parameter yang dikalikan atau dibagi dengan parameter *levenberg marquardt*.
3. Menghitung maju (feedforward) pada hidden dan output layer seperti langkah-langkah pada algoritma *error back propagation*.
4. Menghitung nilai MSE.
5. Menghitung error dan total error jaringan.

Rumus untuk error :

$$e_r = t_r - y_r \dots\dots\dots(22)$$

$r = \text{input ke-}r$

Rumus untuk menghitung total error :

$$e = [e_1 \ e_2 \ e_3 \ \dots \ e_N]^T \dots\dots\dots(23)$$

$e =$ vektor kesalahan berukuran $N \times 1$ yang terdiri dari $e_r \ r = 1,2,3 \dots N$

6. Menghitung matriks Jacobian $J(x)$

x merupakan matriks yang berisi nilai bobot dan bias dari keseluruhan jaringan.

$$X = [\ v_{11}, v_{12}, \dots, v_{ij}; \ v_{01}, v_{02}, \dots, v_{0j}; \ w_{11}, w_{12}, \dots, w_{jk}; \ w_{01}, w_{02}, \dots, w_{0k}] \dots (24)$$

Matriks Jacobian berisi turunan pertama error jaringan terhadap bobot dan bias jaringan. Rumus untuk mencari Jacobian Matriks antara lain:

$$J = [\frac{\partial e_r}{\partial w}] \dots\dots\dots(25)$$

7. Setelah didapatkan nilai $J(x)$ maka dapat dihitung perubahan koreksi bobot dan biasnya dengan rumus berikut :

$$\Delta x = [J(x)^T J(x) + \mu I]^{-1} * \text{Gradient} \dots\dots\dots(26)$$

Gradient : $J(x)^T e$

8. Setelah didapatkan nilai Δx tahap selanjutnya adalah pengoreksian bobot dengan rumus yang sama seperti pada algoritma *error back propagation*.
9. Menghitung maju (feedforward) dengan bobot dan bias yang baru.
10. Menghitung MSE jaringan dengan bobot dan bias yang baru. Kemudian tes kondisi berhenti.
 - Jika *epoch* atau iterasi masih berlanjut maka akan terdapat 2 kemungkinan kondisi berikut : Jika MSE naik $\mu \times \beta$ dan Jika MSE turun $\frac{\mu}{\beta}$
11. Kemudian melakukan kembali langkah 5 sampai langkah 8.

BAB 3. METODOLOGI PENELITIAN

3.1 Tahapan Penelitian

Dalam metodologi penelitian ini, dibuat langkah-langkah dalam melaksanakan penelitian sebagai berikut :

1. Persiapan

Mempersiapkan referensi yang digunakan dalam penelitian. Serta mempersiapkan peralatan pendukung penelitian seperti PC atau laptop yang telah terpasang aplikasi Matlab versi 7.8.0.

2. Merancang permodelan mobil listrik.

Perancangan model mobil listrik menggunakan 8 baterai *lead acid* yang dirangkai seri masing-masing 4 baterai kemudian diparalel. Kapasitas masing-masing baterai sebesar 70 Ah dan tegangan 12 V. Sedangkan beban dimodelkan dengan sebuah motor dengan daya 5 Hp tegangan 240 V dan 1750 rpm.

3. Pengambilan data *input* dan target.

Data input berupa tegangan dan arus dari baterai. Sedangkan kapasitas baterai didapatkan dari batas nominal tegangan yang dapat digunakan yaitu masing-masing baterai 10,5 V yang menandakan kapasitas baterai habis. Secara keseluruhan batas tegangan pemakaian adalah 42 V.

4. Pembuatan sistem peramalan *neural network* berbasis *levenberg marquardt*.

Pembuatan sistem peramalan kapasitas dan waktu habis baterai menggunakan *neural network* yang ada pada software Matlab dengan arsitektur *backpropagation* dan *recurrent* sebagai pembandingnya. Pembuatan masing-masing *neural network* baik *backpropagation* dan *recurrent* menggunakan metode pelatihan *levenberg marquardt*. sehingga nanti akan diketahui sistem peramalan manakah yang lebih baik jika sama-sama menggunakan metode *levenberg marquardt*.

5. Pengujian sistem *neural network*

Setelah sistem peramalan menggunakan *neural network* selesai, maka blok *simulink* dari masing-masing *neural network* yaitu *backpropagation* dan *recurrent*

diujicobakan ke dalam permodelan mobil listrik. dari pengujian tersebut akan muncul grafik sebagai hasil peramalan tersebut.

6. Analisis dan pembahasan

Hasil peramalan menggunakan kedua metode yaitu *backpropagation* dan *recurrent* berbasis *levenberg marquardt* dianalisis dan dibandingkan manakah sistem yang paling bagus.

3.2 MATLAB

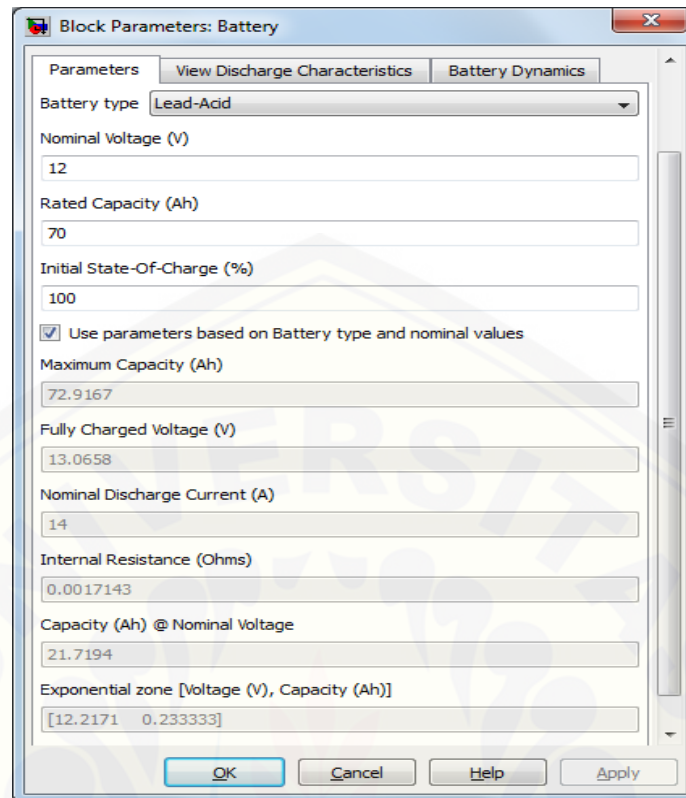
MATLAB adalah singkatan dari *matemathics laboratory* atau *matrix laboratory*. Dalam ilmu komputer, MATLAB didefinisikan sebagai bahasa pemrograman yang digunakan untuk mengerjakan operasi matematika atau operasi aljabar matriks. MATLAB merupakan sistem interaktif yang menggunakan data dasar matriks. Selain itu di dalam MATLAB tersedia perangkat lunak Simulink. Manfaat penggunaan simulink adalah dapat digunakan untuk analisis maupun permodelan suatu sistem. Pada penelitian ini menggunakan simulink yang ada pada MATLAB versi 7.8.0.

3.3 Permodelan Baterai

Pada penelitian ini menggunakan 8 buah baterai yang dirangkai masing-masing 4 baterai diseri kemudian diparalel. Baterai masing-masing mempunyai spesifikasi tegangan nominal 12 V dan kapasitas 70 Ah. Sehingga saat dirangkai seri kapasitas baterai menjadi empat kali lipat, yaitu tegangan 48 V dan kapasitas 140 Ah. Spesifikasi baterai yang digunakan diatur sebagaimana Gambar 3.1 berikut ini.

Tabel 3.1 Spesifikasi Baterai

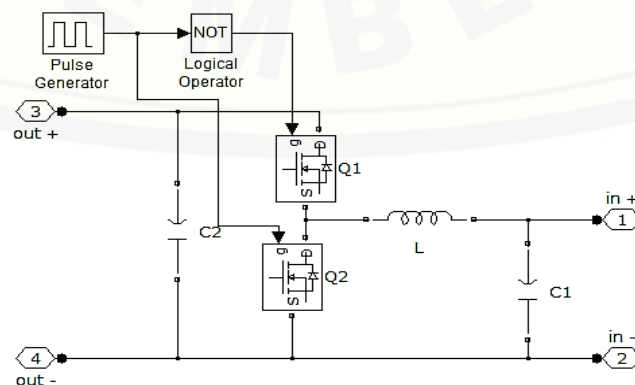
Jenis	<i>Lead acid</i>
Tegangan Nominal	12 V
Kapasitas	70 Ah



Gambar 3.1 Spesifikasi masing-masing baterai *lead acid*

3.4 Permodelan Bidirectional DC-DC Converter

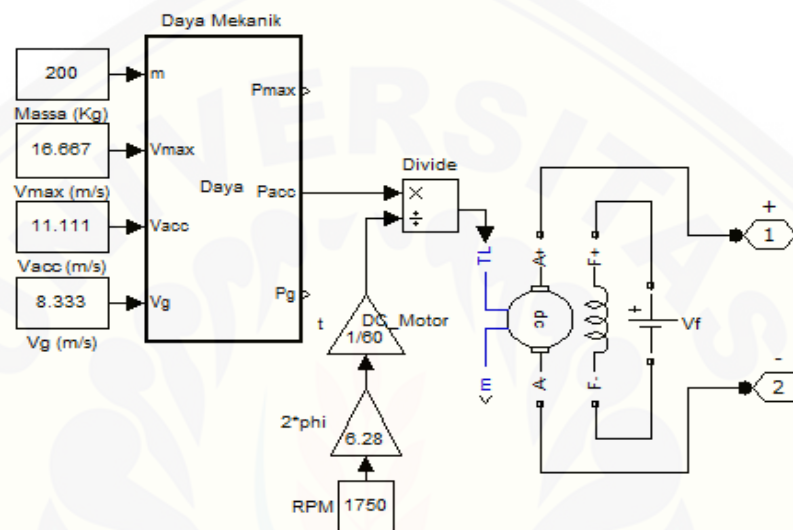
Bidirectional DC-DC converter digunakan untuk menaikkan tegangan DC baterai yaitu 48 V menjadi tegangan nominal motor yaitu 240 V. Spesifikasi dari komponen *bidirectional DC-DC converter* adalah $L = 1600 \mu\text{H}$, $C_H = 470 \mu\text{F}$, $C_L = 470 \mu\text{F}$, $f_{sw} = 20 \text{ KHz}$. Data tersebut mengacu pada referensi (Pany, 2011). Gambar 3.2 berikut ini adalah rangkaian dari *bidirectional DC-DC converter*.



Gambar 3.2 Rangkaian Bidirectional DC-DC Converter (Pany, 2011)

3.5 Permodelan Beban Mobil Listrik

Mobil listrik yang digunakan dimodelkan dengan sebuah motor DC dengan daya 5 hp, tegangan 240 V, putaran motor 1750 rpm, dan tegangan medan 150 V. Masukan torsi motor (TL) ialah hasil perhitungan antara daya mekanik dibagi dengan kecepatan sudut motor listrik. Motor DC yang digunakan dimodelkan seperti pada Gambar 3.2 berikut ini.



Gambar 3.2 Permodelan mobil listrik

Mobil listrik yang digunakan dimodelkan didapatkan dengan spesifikasi pada tabel 3.2 berikut:

Tabel 3.2 Spesifikasi Mobil Listrik

Massa Kendaraan (Kg)	200
Kecepatan maksimum (m/s)	16,667
Kecepatan tanjakan (m/s)	7
Kecepatan akselerasi (m/s)	11,111
Gravitasi (m/s^2)	9,81
Rolling Resistance Coefficient (fr)	0,013
Air Density (Kg/m^3)	1,202
Aerodynamics Drag Coefficient (Cd)	0,29
Front Area of the Vehicle (m^2)	2,13
Koefisien tanjakan sudut 30^0	$\text{Sin } 30^0$

Dari data spesifikasi kendaraan diatas, maka didapatkan tiga nilai kondisi daya mekanik mobil listrik yang akan digunakan dalam pengujian sistem. Hasil perhitungan dari rumus daya mekanik pada tiga kondisi mobil listrik ditampilkan oleh tabel 3.3 berikut :

Tabel 3.3 Kebutuhan daya mekanik

Kondisi	Daya (watt)
Kecepatan maksimum menjelajah	3863
Tanjakan	6867
Akselerasi	3524

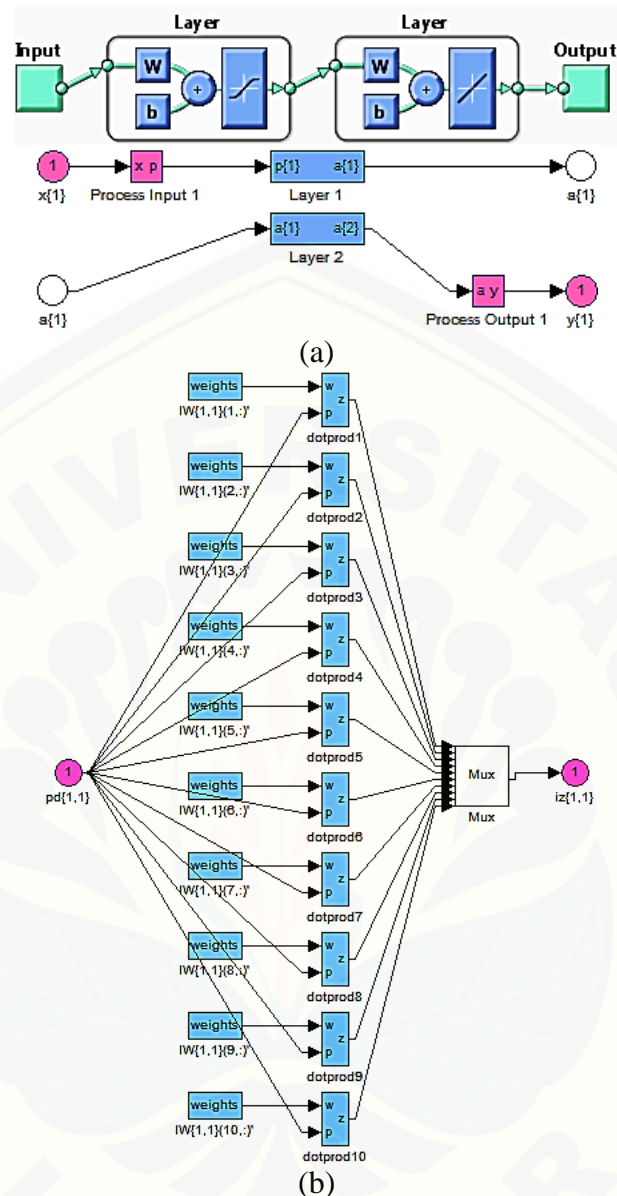
Hasil daya mekanik tersebut digunakan untuk menghitung torsi motor listrik. Hasil perhitungan torsi motor listrik seperti pada Tabel 3.4 sebagai berikut:

Tabel 3.4 Torsi motor listrik

Kondisi	Torsi (Nm)
Kecepatan maksimum menjelajah	21,09
Tanjakan	37,49
Akselerasi	19,23

3.6 Backpropagation

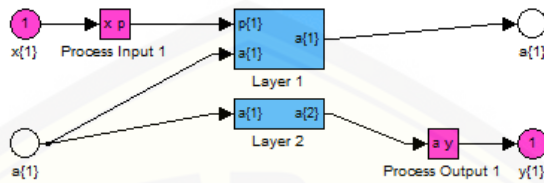
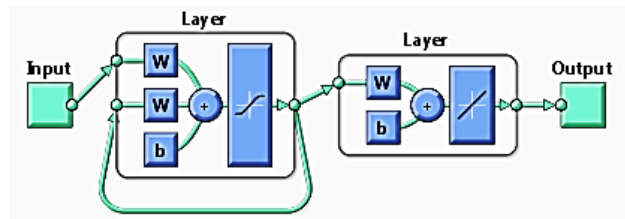
Arsitektur yang digunakan pada *backpropagation* adalah *multilayer network* atau jaringan layar jamak, seperti pada Gambar 3.3 dibawah ini. Parameter *input* berisi dua nilai data yaitu tegangan dan arus dan *output* yaitu kapasitas baterai. Fungsi transfer yang digunakan dalam *backpropagation* yaitu ‘tansig’ untuk layer *input* dan ‘purelin’ untuk layer *output*. Sedangkan hidden layer dicari hasil terbaik dari percobaan menggunakan 1 hidden layer sampai 10 hidden layer, dan hasil terbaik yaitu 10 hidden layer.



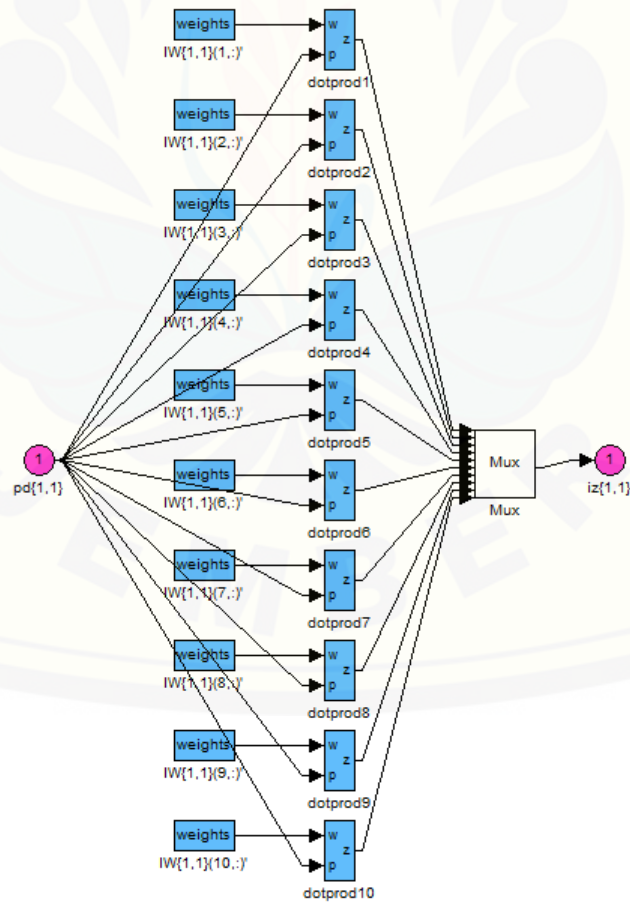
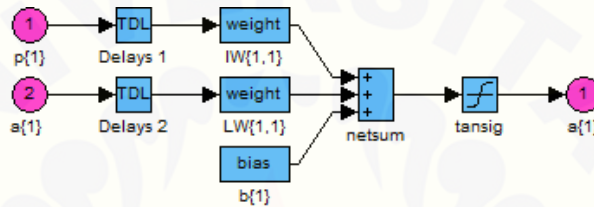
Gambar 3.3 (a) Arsitektur *backpropagation* (b) *Hidden layer* dan bobot.

3.7 Recurrent

Arsitektur yang digunakan pada *recurrent neural network* adalah *recurrent network* atau jaringan berulang, seperti pada Gambar 3.4 dibawah ini. Parameter *input* berisi dua nilai data yaitu tegangan dan arus dan *output* yaitu kapasitas baterai. Fungsi transfer yang digunakan dalam *backpropagation* yaitu ‘tansig’ untuk layer *input* dan ‘purelin’ untuk layer *output*. Sedangkan hidden layer yang digunakan dicari hasil terbaik dari percobaan menggunakan 1 hidden layer sampai 10 hidden layer, dan hasil terbaik yaitu 10 hidden layer.



(a)



(b)

Gambar 3.4 (a) Arsitektur recurrent (b) Hidden layer dan bobot.

3.8 Levenberg Marquardt

Dasar pelatihan yang digunakan adalah menggunakan algoritma *levenberg marquardt*. Parameter dari pelatihan algoritma *levenberg marquardt* dapat diketahui dalam Tabel 3.5 berikut ini.

Tabel 3.5 Parameter pelatihan algoritma *levenberg marquardt*

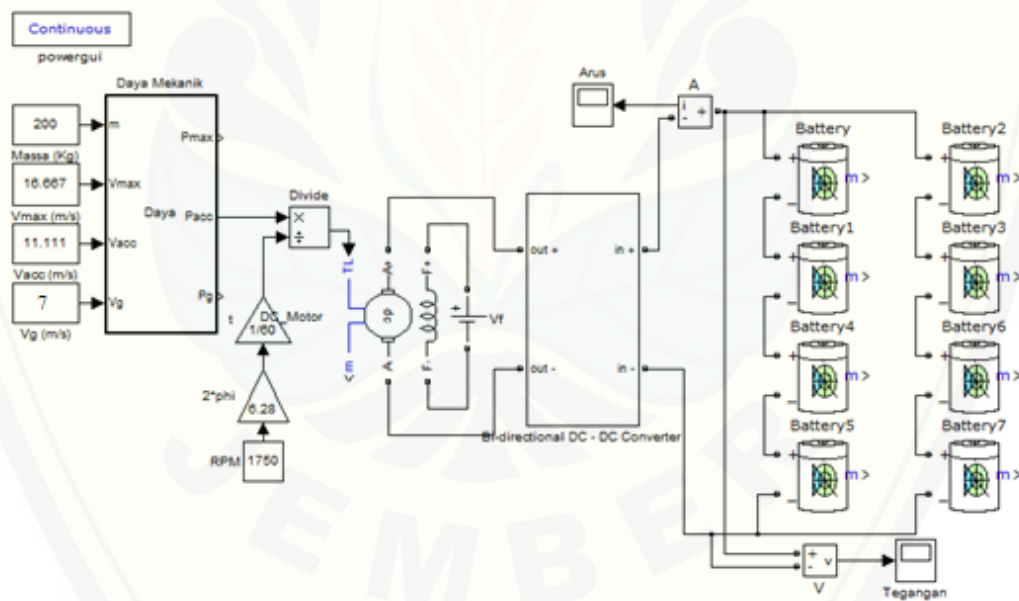
net.trainParam.epochs	1000	Maximum number of epochs to train
net.trainParam.goal	0	Performance goal
net.trainParam.max_fail	6	Maximum validation failures
net.trainParam.min_grad	1e-10	Minimum performance gradient
net.trainParam.mu	0.001	Initial mu
net.trainParam.mu_dec	0.1	mu decrease factor
net.trainParam.mu_inc	10	mu increase factor
net.trainParam.mu_max	1e10	Maximum mu
net.trainParam.show	25	Epochs between displays (NaN for no displays)
net.trainParam.showCommandLine	0	Generate command-line output
net.trainParam.showWindow	1	Show training GUI
net.trainParam.time	inf	Maximum time to train in seconds

Epoch adalah jumlah iterasi maksimum yang boleh dilakukan selama proses pelatihan. *Iterasi* akan dihentikan apabila nilai epoch melebihi maksimum epoch. *Goal* atau kinerja tujuan adalah target nilai fungsi kinerja. *Iterasi* akan dihentikan apabila nilai fungsi kinerja kurang dari atau sama dengan kinerja tujuan. *Show* merupakan jumlah epoch yang akan ditunjukkan kemajuannya. Menunjukkan berapa jumlah epoch berselang yang akan ditunjukkan kemajuannya. Pada sistem ini jumlah epoch yang akan ditunjukkan kemajuannya sebesar 25. *Maximum fail* atau maksimum kegagalan diperlukan apabila pada algoritma disertai dengan validitas (optional). Maksimum kegagalan ketidakvalitan terbesar yang diperbolehkan. Apabila gradien pada *iterasi* ke- k lebih besar daripada gradien iterasi ke- $(k-1)$,

maka kegagalannya akan bertambah 1. Iterasi akan dihentikan apabila jumlah kegagalan lebih dari maksimum kegagalan. Pada sistem ini maksimum kegagalan yang digunakan sebesar 6. Gradien minimum adalah akar dari jumlah kuadrat semua gradien (bobot input, bobot lapisan, bobot bias) terkecil yang diperbolehkan. Iterasi akan dihentikan apabila nilai akar jumlah kuadrat semua gradien ini kurang dari gradien minimum. Pada sistem ini gradien yang digunakan adalah $1e-7$.

3.9 Pemodelan Keseluruhan Mobil Listrik

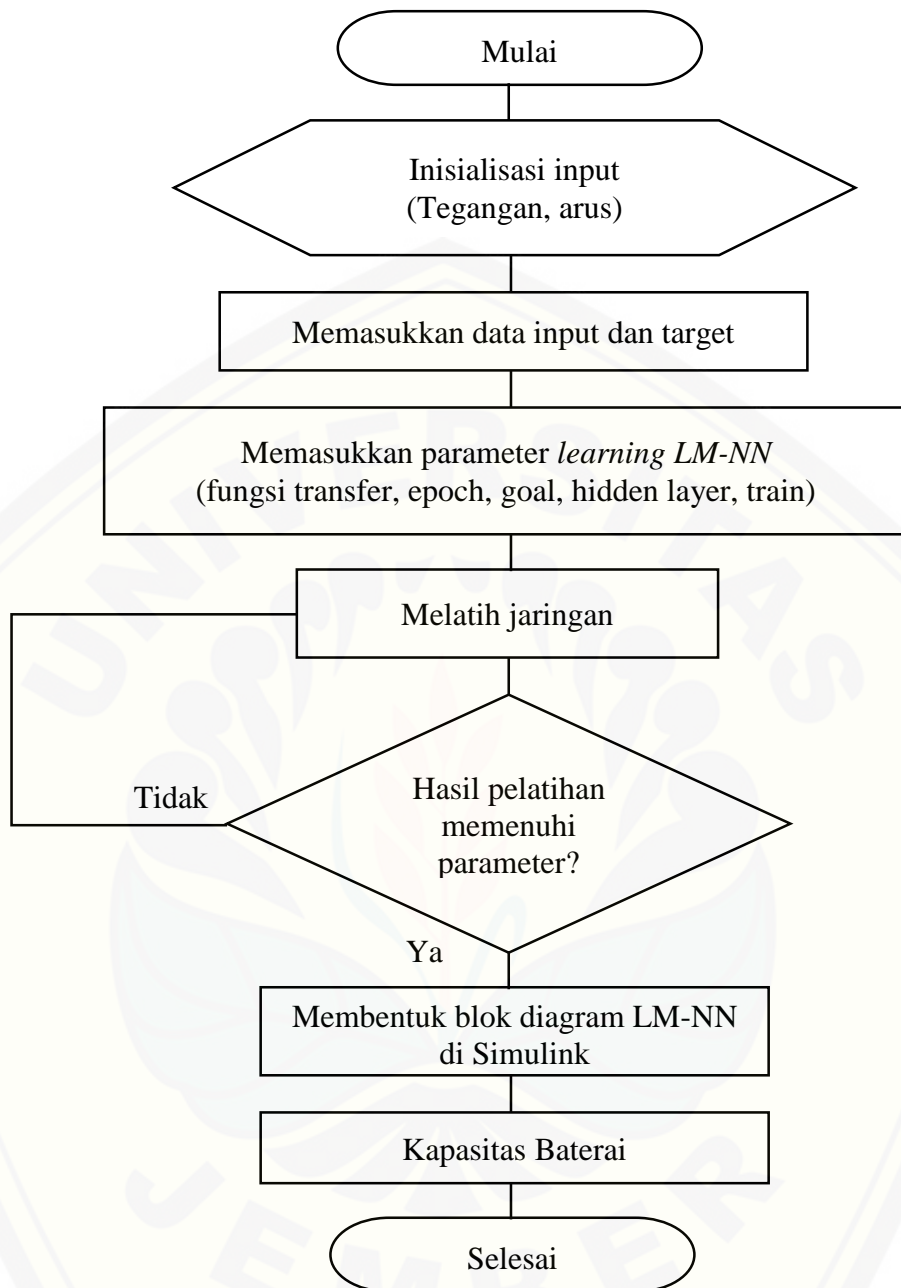
Pengambilan data dilakukan dengan membuat permodelan secara keseluruhan. Kemudian data *input* yang diperlukan adalah tegangan dan arus baterai yang menyuplai motor. Sedangkan data target yang digunakan untuk pembuatan *neural network* adalah kapasitas baterai dan waktu habisnya baterai. Pengambilan data dilakukan dengan permodelan pada Gambar 3.5 berikut ini.



Gambar 3.5 Permodelan keseluruhan mobil listrik

3.10 Alur Penelitian

Pada pembuatan simulasi peramalan kapasitas baterai mobil listrik menggunakan *levenberg marquardt neural network*, proses pembuatan simulasinya dirancang sesuai dengan diagram alur yang ada pada Gambar 3.6 dibawah ini.



Gambar 3.6 Diagram Alur keseluruhan sistem

BAB 5. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil dan analisa data pada bab sebelumnya maka dalam penelitian ini dapat disimpulkan bahwa :

1. Perancangan peramalan kapasitas baterai *lead acid* pada mobil listrik berbasis *levenberg marquardt* ini dapat menggunakan dua buah *neural network* yaitu *backpropagation* dan *recurrent*.
2. Hasil peramalan kapasitas baterai yaitu 140 Ah saat kecepatan akselerasi maka akan habis dalam waktu 1,09 jam. Sedangkan pada saat kecepatan maksimum maka akan habis dalam 0,88 jam dan saat beban kecepatan tanjakan maka akan habis dalam waktu 0,64 jam.
3. Metode terbaik dalam meramalkan sisa kapasitas baterai adalah dengan metode *recurrent neural network*. Hal ini ditunjukkan salah satunya, saat menggunakan kecepatan maksimum (16,67 m/s) hasil peramalan setelah 30 menit berjalan sisa kapasitas baterai dari *backpropagation* adalah 60,5 Ah dan *recurrent* adalah 61 Ah sedangkan targetnya adalah 59,6 Ah.

5.2 Saran

Saran yang dianjurkan untuk pengembangan penelitian lebih lanjut adalah sebagai berikut:

1. Data yang digunakan harus memperhatikan lebih banyak kondisi supaya hasil peramalan lebih optimal.
2. Metode yang digunakan dibandingkan dengan metode-metode peramalan yang lain agar didapatkan metode peramalan yang paling optimal.

DAFTAR PUSTAKA

- Adianshar, Ammar. 2014. *Penerapan Recurrent Neural Network Dalam Identifikasi Tulisan Tangan Huruf Jepang Jenis Katakana*. Universitas Sumatra Utara.
- Agarwal, Vipul & Rajan, Ritu. 2013. Modeling of Electric Vehicle : State of Art. *MIT International Journal of Electrical and Instrumentation Engineering*. Vol. 3(1) : 24–28
- Al-Atas, Husin Mustafa. 2015. *Pengembangan Model Baterai Timbal Asam Berbasis RBFNN (Radial Basic Function Neural Network)*. Universitas Jember.
- Cubon, P., Radvan, R. & Vavrus, V., 2015. Optimization Source Structure of Electric Vehicle Using Calculation of Energy Consumption in Matlab GUI. *Mechatronics*. Vol. 13(2) : 156 - 161.
- Hudaya, Chairul. 2011. *Peranan Riset Baterai Sekunder dalam Mendukung Penyediaan Energi Bersih Di Indonesia 2025*. Proceeding Olimpiade Karya Tulis Inovatif (OKTI)
- Kurniawan, Alif. 2012. *Peramalan Awal Musim Hujan Menggunakan Jaringan Syaraf Tiruan Backpropagation Levenberg – Marquardt*. Institut Pertanian Bogor.
- Liu, J., Saxena, A., Goebel, K., Saha, B., Wang, W., 2010. An Adaptive Recurrent Neural Network for Remaining Useful Life Prediction of Lithium-ion Batteries. *Annual Conference of the Prognostics and Health Management Society*.
- Nasution, Jamil Fahmi. 2009. *Implementasi Jaringan Syaraf Tiruan untuk Prediksi Keputusan Medis Pada Penyakit Demam Berdarah Dengue*. Universitas Sumatra Utara.
- Noviana, Agus. 2008. *Analisa Pengaruh Parameter – Parameter Neural Network Pada Kasus Pemodelan*. Universitas Komputer Indonesia.
- Oktaorora, A., Amaliah, B. & Saikhu, A. 2011. *Penentuan Jenis Produk Kosmetik Pilihan Berdasarkan Faktor Usia Dan Warna Kulit Menggunakan Metode Jaringan Syaraf Tiruan*. Institut Teknologi Sepuluh November.
- Pany, Premananda., R.K. Singh., R.K. Tripathi. 2011. Bidirectional DC-DC converter fed drive for electric vehicle system. *Intenational Journal of Engineering, science and Technology*. Vol. 3(3) : 101 – 110.

- Putra, Bayu Segara. 2015. *Desain dan Implementasi Sistem Monitoring dan Manajemen Baterai Mobil Listrik*. Universitas Telkom
- S., Anda Andycka & Brahmana, K., 2014. *Pembuatan Sumber Tenaga Listrik Cadangan Menggunakan Solar Cell, Baterai dan Inverter Untuk Keperluan Rumah Tangga*. Universitas Sumatra Utara.
- Siregar, Tika Romauli. 2013. *Implementasi Jaringan Syaraf Metode Bidirectional Associative Memory Untuk Pengenalan Pola Wajah*. Universitas Sumatra Utara.
- Wai, C. K., Rong, Y. Y. & Morris, S., 2015. Simulation of a Distance Estimator for Battery Electric Vehicle. *Alexandria Engineering Journal*. Vol. 54 : 359 – 371.
- Yang, Xiyun., Jiang, Feifei., & Wu, Xiaoning., 2013. Prediction of Lead Acid Storage Battery's Remaining Capacity Based on LM-BP Neural Network. *Chinese Control and Decision Conference (CCDC)*. Vol 25 : 1908 – 1912.

LAMPIRAN**A. Perhitungan Daya Mekanik**

1. Beban kecepatan akselerasi awal (11,11 m/s)

$$P = (m g f_{rr} + \rho C_d A V^2 + m) V$$

$$P = (200 \times 9,81 \times 0,013 + 1,202 \times 0,29 \times 2,13 + 200) \times 11,11$$

$$P = 3524 \text{ watt}$$

2. Beban kecepatan tanjakan (7 m/s)

$$P = m \cdot g \cdot \beta \cdot V$$

$$P = 200 \times 9,81 \times \sin 30 \times 7$$

$$P = 6867 \text{ watt}$$

3. Beban kecepatan maksimum (16,67 m/s)

$$P = (f_{rr} m g + \rho A C_d V_{\max}^2) V_{\max}$$

$$P = (0,013 \times 200 \times 9,81 + 1,202 \times 2,13 \times 0,29 \times 16,67^2) \times 16,67$$

$$P = 3863 \text{ watt}$$

B. Perhitungan Torsi Motor Listrik

$$\text{Rumus : } P = \tau \times \omega$$

$$\omega = 2 \pi n / 60$$

$$\omega = 2 \times 3,14 \times 1750 / 60 = 183,167$$

$$\text{maka: } \tau = P / \omega$$

1. Torsi saat kecepatan akselerasi awal

$$\tau = 3524 / 183,167 = 19,23 \text{ Nm}$$

2. Torsi saat kecepatan tanjakan

$$\tau = 6867 / 183,167 = 37,49 \text{ Nm}$$

3. Torsi saat kecepatan maksimum

$$\tau = 3863 / 183,167 = 21,09 \text{ Nm}$$

C. Program Backpropagation

```
load data baterai.mat
P=databaterai(:,1:2)';
T=databaterai(:,3:4)';
net=newff(minmax(P),[10 2],{'tansig','purelin'},'trainlm');
net.trainFcn = 'trainlm';
net.performFcn = 'mse';
[net,tr]=train(net,P,T);
y=sim(net,P);
view(net);
gensim (net);
```

D. Program Recurrent

```
load data baterai.mat
p1=databaterai(:,1:2)';
t1=databaterai(:,3:4)';
p=con2seq(p1);
t=con2seq(t1);
net=newlrm(p1,t1,[10],{'tansig','purelin'},'trainlm');
net.trainFcn = 'trainlm';
net.performFcn = 'mse';
net.plotFcns = {'plotperform','plottrainstate','plotregression'};
net=train(net,p,t);
y=sim(net,p);
plotregression(t1,y);
view(net);
gensim(net);
```

E. Data Input Dan Target

Tegangan (V)	Arus (A)	Kapasitas (Ah)	Time (s)
50,52791	106,8359	140	2
47,82306	106,978	139,2922	22
47,67507	106,9736	138,5844	42
47,65129	106,9711	137,8766	62
47,63937	106,9692	137,1689	82
47,62889	106,9673	136,4611	102
47,61852	106,9655	135,7533	122
47,60809	106,9636	135,0455	142
47,59758	106,9618	134,3377	162
47,58698	106,9599	133,6299	182
47,5763	106,958	132,9222	202
47,56552	106,9561	132,2144	222
47,55465	106,9541	131,5066	242
47,54369	106,9522	130,7988	262
47,53263	106,9502	130,091	282
47,52148	106,9483	129,3832	302

47,51023	106,9463	128,6755	322
47,49888	106,9443	127,9677	342
47,48744	106,9423	127,2599	362
47,47589	106,9403	126,5521	382
47,46425	106,9383	125,8443	402
47,4525	106,9363	125,1365	422
47,44064	106,9342	124,4288	442
47,42868	106,9321	123,721	462
47,41661	106,9302	123,0132	482
47,40443	106,9281	122,3054	502
47,39215	106,926	121,5976	522
47,37975	106,9239	120,8898	542
47,36724	106,9217	120,182	562
47,35461	106,9196	119,4743	582
47,34187	106,9174	118,7665	602
47,32901	106,9153	118,0587	622
47,31603	106,913	117,3509	642
47,30293	106,9108	116,6431	662
47,2897	106,9086	115,9353	682
47,27635	106,9064	115,2276	702
47,26288	106,9042	114,5198	722
47,24927	106,9019	113,812	742
47,23554	106,8997	113,1042	762
47,22167	106,8975	112,3964	782
47,20767	106,8952	111,6886	802
47,19354	106,8929	110,9809	822
47,17927	106,8907	110,2731	842
47,16485	106,8884	109,5653	862
47,1503	106,8861	108,8575	882
47,1356	106,8838	108,1497	902
47,12075	106,8815	107,4419	922
47,10576	106,8792	106,7342	942
47,09061	106,8769	106,0264	962
47,07532	106,8745	105,3186	982
47,05986	106,8722	104,6108	1002
47,04425	106,8699	103,903	1022
47,02849	106,8676	103,1952	1042
47,01255	106,8653	102,4874	1062
46,99646	106,8629	101,7797	1082
46,98019	106,8606	101,0719	1102
46,96376	106,8583	100,3641	1122

46,94715	106,856	99,65631	1142
46,93037	106,8538	98,94853	1162
46,9134	106,8515	98,24074	1182
46,89626	106,8492	97,53296	1202
46,87893	106,847	96,82518	1222
46,86142	106,8448	96,11739	1242
46,84372	106,8427	95,40961	1262
46,82582	106,8405	94,70182	1282
46,80772	106,8385	93,99404	1302
46,78943	106,8365	93,28626	1322
46,77093	106,8345	92,57847	1342
46,75223	106,8327	91,87069	1362
46,73332	106,8309	91,1629	1382
46,71419	106,8292	90,45512	1402
46,69484	106,8277	89,74734	1422
46,67528	106,8264	89,03955	1442
46,65549	106,8254	88,33177	1462
46,63547	106,8246	87,62398	1482
46,61521	106,8242	86,9162	1502
46,59472	106,8245	86,20842	1522
46,57398	106,8257	85,50063	1542
46,553	106,8288	84,79285	1562
46,53173	106,8402	84,08506	1582
46,51135	106,5192	83,37728	1602
46,48803	106,9829	82,6695	1622
46,46744	106,5714	81,96171	1642
46,44458	106,7437	81,25393	1662
46,4211	107,0195	80,54614	1682
46,40029	106,4321	79,83836	1702
46,37743	106,3619	79,13058	1722
46,35406	106,3581	78,42279	1742
46,33039	106,3543	77,71501	1762
46,30643	106,3505	77,00722	1782
46,28216	106,3467	76,29944	1802
46,25758	106,3428	75,59166	1822
46,23268	106,3389	74,88387	1842
46,20746	106,3349	74,17609	1862
46,18192	106,331	73,4683	1882
46,15604	106,327	72,76052	1902
46,12981	106,3229	72,05274	1922
46,10324	106,3189	71,34495	1942

46,07632	106,3148	70,63717	1962
46,04903	106,3107	69,92938	1982
46,02137	106,3065	69,2216	2002
45,99334	106,3023	68,51382	2022
45,96476	106,3437	67,80603	2042
45,93595	106,3393	67,09825	2062
45,90675	106,3294	66,39046	2082
45,87646	106,5216	65,68268	2102
45,84624	106,5679	64,9749	2122
45,81606	106,4799	64,26711	2142
45,78478	106,5855	63,55933	2162
45,75262	106,8167	62,85154	2182
45,7208	106,8175	62,14376	2202
45,69032	106,2919	61,43598	2222
45,65767	106,2588	60,72819	2242
45,62448	106,2408	60,02041	2262
45,59072	106,2469	59,31262	2282
45,55635	106,2811	58,60484	2302
45,52175	106,2299	57,89706	2322
45,48639	106,2464	57,18927	2342
45,45061	106,2285	56,48149	2362
45,41421	106,2282	55,7737	2382
45,3772	106,2395	55,06592	2402
45,33971	106,2181	54,35814	2422
45,30158	106,2145	53,65035	2442
45,26285	106,2044	52,94257	2462
45,2235	106,1938	52,23478	2482
45,18351	106,1825	51,527	2502
45,14287	106,1685	50,81922	2522
45,10155	106,1584	50,11143	2542
45,05953	106,1507	49,40365	2562
45,0168	106,1428	48,69586	2582
44,97335	106,135	47,98808	2602
44,92943	106,2388	47,2803	2622
44,88558	106,232	46,57251	2642
44,84172	106,2252	45,86473	2662
44,79786	106,2183	45,15694	2682
44,754	106,2115	44,44916	2702
44,71015	106,2047	43,74138	2722
44,66629	106,1979	43,03359	2742
44,62243	106,191	42,32581	2762

44,57857	106,1842	41,61802	2782
44,53472	106,1774	40,91024	2802
44,49086	106,1705	40,20246	2822
44,447	106,1637	39,49467	2842
44,40314	106,1569	38,78689	2862
44,35929	106,1501	38,0791	2882
44,31543	106,1432	37,37132	2902
44,27157	106,1364	36,66354	2922
44,22771	106,1296	35,95575	2942
44,18386	106,1228	35,24797	2962
44,14	106,1159	34,54018	2982
44,09614	106,1091	33,8324	3002
44,05228	106,1023	33,12462	3022
44,00843	106,0955	32,41683	3042
43,96457	106,0886	31,70905	3062
43,92071	106,0818	31,00126	3082
43,87685	106,075	30,29348	3102
43,833	106,0681	29,5857	3122
43,78914	106,0613	28,87791	3142
43,74528	106,0545	28,17013	3162
43,70143	106,0477	27,46234	3182
43,65757	106,0408	26,75456	3202
43,61371	106,034	26,04678	3222
43,56985	106,0272	25,33899	3242
43,526	106,0204	24,63121	3262
43,48214	106,0135	23,92342	3282
43,43828	106,0067	23,21564	3302
43,39442	105,9999	22,50786	3322
43,35057	105,993	21,80007	3342
43,30671	105,9862	21,09229	3362
43,26285	105,9794	20,3845	3382
43,21899	105,9726	19,67672	3402
43,17514	105,9657	18,96894	3422
43,13128	105,9589	18,26115	3442
43,08742	105,9521	17,55337	3462
43,04356	105,9453	16,84558	3482
42,99971	105,9384	16,1378	3502
42,95585	105,9316	15,43002	3522
42,91199	105,9248	14,72223	3542
42,86813	105,918	14,01445	3562
42,82428	105,9111	13,30666	3582

42,78042	105,9043	12,59888	3602
42,73656	105,8975	11,8911	3622
42,6927	105,8906	11,18331	3642
42,64885	105,8838	10,47553	3662
42,60499	105,877	9,767744	3682
42,56113	105,8702	9,05996	3702
42,51728	105,8633	8,352176	3722
42,47342	105,8565	7,644392	3742
42,42956	105,8497	6,936608	3762
42,3857	105,8429	6,228824	3782
42,34185	105,836	5,52104	3802
42,29799	105,8292	4,813256	3822
42,25413	105,8224	4,105472	3842
42,21027	105,8155	3,397688	3862
42,16642	105,8087	2,689904	3882
42,12256	105,8019	1,98212	3902
42,0787	105,7951	1,274336	3922
42,03484	105,7882	0,566552	3942
49,01197	116,0988	139,8214	6
47,64005	116,1692	138,9286	26
47,54638	116,179	138,0357	46
47,52545	116,1816	137,1429	66
47,51269	116,1834	136,25	86
47,50092	116,1861	135,3571	106
47,48919	116,1913	134,4643	126
47,47788	116,0452	133,5714	146
47,46627	115,9609	132,6786	166
47,45357	116,1606	131,7857	186
47,43818	117,1115	130,8929	206
47,42966	116,0267	130	226
47,41671	116,2014	129,1071	246
47,40475	116,0527	128,2143	266
47,39184	116,1484	127,3214	286
47,37954	116,0308	126,4286	306
47,36578	116,3034	125,5357	326
47,35107	116,8189	124,6429	346
47,34071	116,0277	123,75	366
47,3281	115,8581	122,8571	386
47,31478	115,8578	121,9643	406
47,30139	115,8408	121,0714	426
47,28636	116,2634	120,1786	446

47,27408	115,8456	119,2857	466
47,26051	115,7645	118,3929	486
47,24636	115,8118	117,5	506
47,23164	115,9872	116,6071	526
47,21809	115,7774	115,7143	546
47,20301	115,9741	114,8214	566
47,1892	115,7582	113,9286	586
47,17261	116,3095	113,0357	606
47,15784	116,286	112,1429	626
47,14241	116,4089	111,25	646
47,12912	115,8645	110,3571	666
47,11047	116,8373	109,4643	686
47,09803	115,9552	108,5714	706
47,08267	115,8753	107,6786	726
47,06528	116,3398	106,7857	746
47,05068	115,9417	105,8929	766
47,03444	115,9739	105	786
47,0183	115,9228	104,1071	806
47,00173	115,9507	103,2143	826
46,98569	115,771	102,3214	846
46,96693	116,3299	101,4286	866
46,95171	115,8044	100,5357	886
46,93398	115,9572	99,64286	906
46,91723	115,7699	98,75001	926
46,89899	115,9575	97,85715	946
46,87989	116,3421	96,96429	966
46,86326	115,9469	96,07144	986
46,84367	116,3546	95,17858	1006
46,82741	115,7325	94,28572	1026
46,808	115,9681	93,39286	1046
46,78718	116,5551	92,50001	1066
46,76993	116,0342	91,60715	1086
46,75141	115,8201	90,71429	1106
46,73163	115,9072	89,82144	1126
46,71033	116,373	88,92858	1146
46,69242	115,7826	88,03572	1166
46,67201	115,8496	87,14287	1186
46,6505	116,1707	86,25001	1206
46,63054	115,9664	85,35715	1226
46,60799	116,4454	84,46429	1246
46,58829	116,0186	83,57144	1266

46,56809	115,6636	82,67858	1286
46,54314	116,6154	81,78572	1306
46,52303	116,0789	80,89287	1326
46,50186	115,7709	80,00001	1346
46,4798	115,6409	79,10715	1366
46,4556	116,0532	78,2143	1386
46,43341	115,7932	77,32144	1406
46,40915	116,054	76,42858	1426
46,3855	116,049	75,53572	1446
46,36302	115,6109	74,64287	1466
46,33821	115,7639	73,75001	1486
46,31074	116,6002	72,85715	1506
46,28906	115,6515	71,9643	1526
46,26263	115,9923	71,07144	1546
46,23652	116,1427	70,17858	1566
46,20794	116,9134	69,28573	1586
46,18473	116,0145	68,39287	1606
46,15874	115,8205	67,50001	1626
46,1313	115,9447	66,60715	1646
46,10469	115,7191	65,7143	1666
46,07728	115,6143	64,82144	1686
46,04583	116,5741	63,92858	1706
46,02074	115,5656	63,03573	1726
45,98778	116,7328	62,14287	1746
45,96113	115,9378	61,25001	1766
45,93264	115,5565	60,35716	1786
45,90069	116,0594	59,4643	1806
45,8717	115,5694	58,57144	1826
45,8395	115,8859	57,67858	1846
45,8091	115,5407	56,78573	1866
45,77381	116,4848	55,89287	1886
45,74543	115,2735	55,00001	1906
45,71025	115,9004	54,10716	1926
45,6785	115,382	53,2143	1946
45,64008	116,6588	52,32144	1966
45,60727	116,1438	51,42859	1986
45,5727	115,985	50,53573	2006
45,53872	115,494	49,64287	2026
45,50092	115,9494	48,75001	2046
45,46449	115,8393	47,85716	2066
45,42779	115,6336	46,9643	2086

45,39011	115,5354	46,07144	2106
45,34936	116,153	45,17859	2126
45,31217	115,5456	44,28573	2146
45,27057	116,0332	43,39287	2166
45,23116	115,6871	42,50002	2186
45,18958	115,7737	41,60716	2206
45,148	115,6569	40,7143	2226
45,10608	115,4268	39,82144	2246
45,06228	115,528	38,92859	2266
45,01888	115,2888	38,03573	2286
44,97384	115,3028	37,14287	2306
44,92796	115,3283	36,25002	2326
44,87654	116,7269	35,35716	2346
44,83394	115,3107	34,4643	2366
44,78575	115,2665	33,57145	2386
44,73642	115,2984	32,67859	2406
44,68537	115,5625	31,78573	2426
44,63304	115,9255	30,89287	2446
44,58164	115,7328	30,00002	2466
44,52962	115,4302	29,10716	2486
44,47669	115,0957	28,2143	2506
44,42013	115,5091	27,32145	2526
44,36018	116,5957	26,42859	2546
44,30775	115,1631	25,53573	2566
44,24868	115,3275	24,64288	2586
44,18982	115,0848	23,75002	2606
44,12644	115,8057	22,85716	2626
44,06598	115,305	21,9643	2646
44,00083	115,7897	21,07145	2666
43,93605	115,7782	20,17859	2686
43,86988	115,7664	19,28573	2706
43,80227	115,7542	18,39288	2726
43,7332	115,7414	17,50002	2746
43,66258	115,7325	16,60716	2766
43,59273	115,0382	15,71431	2786
43,51803	115,2819	14,82145	2806
43,44356	114,9683	13,92859	2826
43,36627	114,9703	13,03573	2846
43,28611	115,2796	12,14288	2866
43,20564	115,1335	11,25002	2886
43,12095	115,6539	10,35716	2906

43,03859	114,9098	9,464307	2926
42,95166	114,892	8,57145	2946
42,86275	114,8222	7,678593	2966
42,77059	115,0489	6,785736	2986
42,67495	115,6133	5,892879	3006
42,58155	114,8216	5,000022	3026
42,48031	115,5824	4,107165	3046
42,38021	115,2555	3,214308	3066
42,2781	114,7204	2,321451	3086
42,16882	115,4576	1,428594	3106
42,06185	114,6657	0,535737	3126
47,27143	197,1515	139,514	10
46,56338	197,0224	138,299	30
46,45853	196,6911	137,084	50
46,42211	196,6052	135,869	70
46,39363	196,846	134,654	90
46,36761	196,6254	133,439	110
46,34105	196,4908	132,224	130
46,30913	197,8028	131,009	150
46,2851	196,69	129,794	170
46,25641	196,8038	128,579	190
46,22881	196,466	127,364	210
46,19832	196,8356	126,149	230
46,16793	197,0362	124,934	250
46,13863	196,7734	123,719	270
46,10788	196,7879	122,504	290
46,07664	196,7935	121,289	310
46,04576	196,5404	120,074	330
46,0126	196,7953	118,859	350
45,9799	196,749	117,644	370
45,94665	196,7004	116,429	390
45,91301	196,5908	115,214	410
45,87801	196,7032	113,999	430
45,84331	196,5503	112,784	450
45,80745	196,5454	111,569	470
45,77092	196,5478	110,354	490
45,73374	196,5453	109,139	510
45,69513	196,756	107,924	530
45,65692	196,6434	106,709	550
45,61796	196,5377	105,494	570
45,57759	196,6253	104,279	590

45,53691	196,5777	103,064	610
45,49488	196,6904	101,849	630
45,45275	196,5962	100,634	650
45,40824	196,9485	99,419	670
45,36592	196,4102	98,204	690
45,32075	196,4431	96,989	710
45,27438	196,5548	95,774	730
45,22791	196,4223	94,559	750
45,17918	196,6615	93,344	770
45,13035	196,6352	92,129	790
45,08068	196,5496	90,914	810
45,02967	196,5398	89,699	830
44,97794	196,4185	88,484	850
44,92477	196,3803	87,269	870
44,86976	196,5365	86,054	890
44,8155	196,1121	84,839	910
44,75785	196,3108	83,624	930
44,69825	196,6918	82,409	950
44,63844	196,7425	81,194	970
44,57471	197,5231	79,979	990
44,65823	196,5677	78,764	1010
44,61726	196,5626	77,549	1030
44,57629	196,5575	76,334	1050
44,53532	196,5524	75,119	1070
44,49436	196,5472	73,904	1090
44,45339	196,5421	72,689	1110
44,41242	196,537	71,474	1130
44,37145	196,5319	70,259	1150
44,33048	196,5267	69,044	1170
44,28952	196,5216	67,829	1190
44,24855	196,5165	66,614	1210
44,20758	196,5114	65,399	1230
44,16661	196,5062	64,184	1250
44,12564	196,5011	62,969	1270
44,08468	196,496	61,754	1290
44,04371	196,4909	60,539	1310
44,00274	196,4857	59,324	1330
43,96177	196,4806	58,109	1350
43,9208	196,4755	56,894	1370
43,87984	196,4704	55,679	1390
43,83887	196,4652	54,464	1410

43,7979	196,4601	53,249	1430
43,75693	196,455	52,034	1450
43,71596	196,4499	50,819	1470
43,675	196,4447	49,604	1490
43,63403	196,4396	48,389	1510
43,59306	196,4345	47,174	1530
43,55209	196,4294	45,959	1550
43,51112	196,4242	44,744	1570
43,47016	196,4191	43,529	1590
43,42919	196,414	42,314	1610
43,38822	196,4089	41,099	1630
43,34725	196,4037	39,884	1650
43,30628	196,3986	38,669	1670
43,26532	196,3935	37,454	1690
43,22435	196,3883	36,239	1710
43,18338	196,3832	35,024	1730
43,14241	196,3781	33,809	1750
43,10144	196,373	32,594	1770
43,06048	196,3678	31,379	1790
43,01951	196,3627	30,164	1810
42,97854	196,3576	28,949	1830
42,93757	196,3525	27,734	1850
42,8966	196,3473	26,519	1870
42,85564	196,3422	25,304	1890
42,81467	196,3371	24,089	1910
42,7737	196,332	22,874	1930
42,73273	196,3268	21,659	1950
42,69176	196,3217	20,444	1970
42,6508	196,3166	19,229	1990
42,60983	196,3115	18,014	2010
42,56886	196,3063	16,799	2030
42,52789	196,3012	15,584	2050
42,48692	196,2961	14,369	2070
42,44596	196,291	13,154	2090
42,40499	196,2858	11,939	2110
42,36402	196,2807	10,724	2130
42,32305	196,2756	9,509	2150
42,28208	196,2705	8,294	2170
42,24112	196,2653	7,079	2190
42,20015	196,2602	5,864	2210
42,15918	196,2551	4,649	2230

42,11821	196,25	3,434	2250
42,07724	196,2448	2,219	2270
42,03628	196,2397	1,004	2290

