

# **IDENTIFIKASI MULTI ELEMEN PADA SPEKTRUM EMISI LIBS KOMPLEKS DARI TANAH VULKANIK SEULAWAH AGAM MENGGUNAKAN ALGORITMA *LONG SHORT-TERM MEMORY (LSTM)***

## **PROPOSAL PENELITIAN**

Diajukan untuk melengkapi tugas-tugas dan  
memenuhi syarat-syarat guna pelaksanaan penelitian Tugas Akhir

Oleh:

**BIRRUL WALIDAIN**  
**2008102010010**



**PROGRAM STUDI SARJANA FISIKA DEPARTEMEN FISIKA  
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS SYIAH KUALA, BANDA ACEH  
DESEMBER, 2024**

## **PENGESAHAN PROPOSAL**

### **IDENTIFIKASI MULTI ELEMEN PADA SPEKTRUM EMISI LIBS KOMPLEKS DARI TANAH VULKANIK SEULAWAH AGAM MENGGUNAKAN ALGORITMA *LONG SHORT-TERM MEMORY* (LSTM)**

Oleh:

Nama : Birrul Walidain  
NPM : 2008102010010  
Departemen : Fisika

Menyetujui:

Pembimbing I,

Pembimbing II,

Prof. Dr. Eng. Nasrullah, S.Si., M.T.  
NIP. 197607031995121001

Dr. Khairun Saddami, S.T.  
NIP. 199103182022031008

Mengetahui:

Koordinator Program Studi S1 Fisika FMIPA  
Universitas Syiah Kuala,

Dr. Saumi Syahreza, S.Si, M.Si.  
NIP. 197609172005011002

## KATA PENGANTAR

Segala puji dan syukur kehadiran Allah SWT yang telah melimpahkan rahmat dan hidayah-Nya kepada kita semua, sehingga penulis dapat menyelesaikan penulisan Tugas Akhir yang berjudul **IDENTIFIKASI MULTI ELEMEN PADA SPEKTRUM EMISI LIBS KOMPLEKS DARI TANAH VULKANIK SEULAWAH AGAM MENGGUNAKAN ALGORITMA *LONG SHORT-TERM MEMORY* (LSTM)** yang telah dapat diselesaikan sesuai rencana. Penulis banyak mendapatkan berbagai pengarahan, bimbingan, dan bantuan dari berbagai pihak. Oleh karena itu, melalui tulisan ini penulis mengucapkan rasa terima kasih kepada:

1. Bapak Dr. Saumi Syahreza, S.Si., M.Si. selaku Ketua Departemen Fisika Fakultas MIPA Universitas Syiah Kuala.
2. Bapak Prof. Dr. Eng. Nasrullah, S.Si., M.T. selaku Dosen Pembimbing I yang telah banyak memberikan bimbingan dan arahan kepada penulis, sehingga penulis dapat menyelesaikan Tugas Akhir ini.
3. Bapak Dr. Khairun Saddami, S.T. selaku Dosen Pembimbing II yang telah banyak memberikan bimbingan dan arahan kepada penulis, sehingga penulis dapat menyelesaikan Tugas Akhir ini.
4. Bapak Dr. Kurnia Lahna, M.T. selaku Dosen Wali yang telah membimbing dan memberikan motivasi kepada penulis selama masa perkuliahan.
5. Ayah dan Ibu sebagai kedua orang tua penulis yang senantiasa selalu mendukung aktivitas dan kegiatan yang penulis lakukan baik secara moral maupun material serta menjadi motivasi terbesar bagi penulis untuk menyelesaikan Tugas Akhir ini.
6. Seluruh Dosen di Departemen Fisika Fakultas MIPA atas ilmu dan didikannya selama perkuliahan.
7. Sahabat dan teman-teman seperjuangan Departemen Fisika USK 2020 lainnya.

Penulis juga menyadari segala ketidaksempurnaan yang terdapat didalamnya baik dari segi materi, cara, ataupun bahasa yang disajikan. Seiring dengan ini penulis mengharapkan kritik dan saran dari pembaca yang sifatnya dapat berguna untuk kesempurnaan Tugas Akhir ini. Harapan penulis semoga tulisan ini dapat bermanfaat bagi banyak pihak dan untuk perkembangan ilmu pengetahuan.

Banda Aceh, 14 Agustus 2024

Birrul Walidain  
NPM. 2008102010010

## **DAFTAR ISI**

## **DAFTAR TABEL**

## **DAFTAR GAMBAR**

# **BAB I**

## **PENDAHULUAN**

### **1.1 LATAR BELAKANG**

Spektroskopi secara sederhana merupakan interaksi antara materi dan radiasi elektromagnetik yang dapat memberi informasi bagaimana materi menyerap, memancarkan, atau menghamburkan cahaya. Kemudian interaksi ini digunakan untuk mengidentifikasi dan mengukur komposisi, struktur, dan sifat materi tersebut (?). Joseph von Fraunhofer (1778 - 1826) adalah ilmuwan yang menemukan garis spektral hitam pada spektrum matahari dari penelitiannya dalam mencari indeks bias dari cahaya monokromatik. Hingga sekitar 50 tahun kemudian, Gustav Robert Kirchhoff (1824-1887) dan Robert Bunsen (1811-1899) menemukan bahwa setiap zat kimia memiliki karakteristiknya masing-masing berupa garis spektrum dan menjadi dasar untuk analisis spektrokimia. Upaya berlanjut hingga pada tahun 1885, Johann Jakob Balmer (1825-1898) memprediksi panjang gelombang spektrum Hidrogen. Perkembangan terus berlanjut hingga dasar teori laser diusulkan oleh Schawlow dan Townes pada tahun 1958 dan tercatat eksperimen pertama laser menggunakan kristal ruby pada tahun 1960 oleh Maiman. Dua tahun kemudian laser ruby digunakan untuk menghasilkan plasma pada sampel dan mengamati spektrum emisi oleh Brech dan Cross (?).

*Laser-Induced Breakdown Spectroscopy (LIBS)* adalah teknik analisis yang melibatkan penggunaan laser berenergi tinggi yang kemudian difokuskan ke permukaan sampel untuk menciptakan plasma. LIBS menunjukkan potensinya sebagai teknik analisis yang cepat dan serbaguna untuk berbagai aplikasi, termasuk analisis material, pemantauan lingkungan, dan analisis makanan. LIBS telah menjadi teknik yang menarik dan populer di bidang analisis kimia karena keunggulannya yang unik, seperti aplikasinya pada cairan, gas, dan padatan, tidak ada perlakuan awal sampel, deteksi simultan beberapa elemen, dan deteksi jarak jauh tanpa kontak di berbagai bidang, termasuk pembersihan laser, perlindungan lingkungan, eksplorasi ruang angkasa, dan pelestarian warisan budaya. Namun kompleksitas spektrum yang dihasilkan dan pengaruh faktor-faktor seperti efek matriks dan fluktuasi plasma menjadi tantangan dalam analisis kualitatif maupun kuantitatif pada LIBS (?). *Artificial Neural Network (ANN)* diusulkan untuk memberikan metode yang layak, cepat, dan kuat pada analisis LIBS. Hasil penelitian D'Andrea berhasil memprediksi konsentrasi unsur-unsur dalam sampel perunggu dengan akurasi yang baik (?).

Tanah vulkanis, yang dikenal sebagai Andosol, memiliki karakteristik unik yang menjadikannya sangat subur. Di Indonesia, tanah ini umumnya terbentuk dari

material vulkanik yang kaya akan mineral dan memiliki kapasitas retensi air yang tinggi. Penelitian menunjukkan bahwa penggunaan lahan yang intensif, seperti pertanian sayuran, dapat menyebabkan penurunan pH dan kehilangan bahan organik, yang berdampak negatif pada kesuburan tanah. Namun, praktik pertanian organik dapat membantu memulihkan kesuburan tanah dengan meningkatkan kandungan karbon organik dan nutrisi lainnya (?). Kompleksitas dari elemen dalam tanah vulkanis menjadi tantangan untuk dianalisa. Oleh karenanya dibutuhkan algoritma ANN yang mampu memprediksi multi elemen secara *real-time*. Pada penelitian ini, model *Deep Learning Long Shor-Term Memory* (LSTM) diusulkan untuk mempelajari spektrum berbagai elemen. Agar model LSTM dapat mempelajari spektrum, dibutuhkan spektrum elemen murni untuk menjadi *dataset* yang cukup besar. Untuk menjawab permasalahan ini, simulasi spektrum murni dilakukan dengan menghitung intensitas relatif dari tiap elemen. Profil garis *Gaussian* digunakan untuk melengkapi simulasi spektrum murni. Spektrum Elemen murni akan disimulasikan untuk berbagai keadaan suhu plasma dan memberikan rasio intensitas yang cukup untuk dipelajari model LSTM.

## **1.2 RUMUSAN MASALAH**

? menyatakan bahwa karakteristik tanah vulkanis unik dan menjadikannya sangat subu. Namun, analisa multi elemen pada tanah yang memiliki kompleksitas tinggi sulit untuk di analisa dalam waktu singkat dan akurat. Oleh karena itu muncul beberapa pertanyaan terkait metode alternatif untuk mempercepat analisa LIBS secara *real-ti* dan efisien; bagaimana algoritma LSTM dapat memprediksi multi elemen pada tanah vulkanis dan keakuratannya; apakah ada spektrum elemen yang bisa diidentifikasi oleh LSTM berbeda dengan identifikasi manual.

## **1.3 TUJUAN PENELITIAN**

Berdasarkan latar belakang dan rumusan masalah diatas, tujuan dari penelitian ini adalah untuk menguji metode LIBS berbasis LSTM dalam memprediksi multi elemen pada tanah vulkanik.

## **1.4 MANFAAT PENELITIAN**

Dengan dibuktikan nya algoritma LSTM dalam memprediksi multi elemen pada spektrum LIBS akan membawa pendekatan baru pada deteksi elemen mayor dan minor pada metode LIBS. Kemudian juga akan mendapatkan validasi dan hipotesa awal pada spektrum LIBS apa saja.



## BAB II

### TINJAUAN PUSTAKA

Dalam bab ini, berbagai teori yang relevan dan literatur sebelumnya yang menunjang penelitian ini dibahas secara komprehensif. Pembahasan mencakup teknik spektroskopi yang diaplikasikan, metode analisis spektrum, serta penerapan teknik statistik dalam pemrosesan data spektroskopi dengan penekanan pada penerapan *Deep Learning* di dalamnya. Arsitektur *Long Short-Term Memory* (LSTM) diterapkan untuk memprediksi elemen dalam sampel tersebut.

#### 2.1 SPEKTROSKOPI EMISI ATOM

##### 2.1.1 Prinsip Dasar

Spektroskopi emisi atom bertumpu pada teori atom Bohr yang menjelaskan kuantisasi energi elektron melalui persamaan:

$$E_n = -\frac{13.6 \text{ eV}}{n^2} \quad (n \in \mathbb{Z}^+), \quad (2.1)$$

di mana transisi elektron antar tingkat energi memenuhi  $\Delta E = h\nu$  (?). Fenomena spektrum garis hidrogen mengikuti persamaan umum Rydberg:

$$\frac{1}{\lambda} = R \left( \frac{1}{n_f^2} - \frac{1}{n_i^2} \right) \quad (n_i > n_f), \quad (2.2)$$

dengan  $R = 1.097 \times 10^7 \text{ m}^{-1}$  sebagai konstanta Rydberg (?). Contoh spesifik seperti deret Balmer ( $n_f = 2$ ) dan Lyman ( $n_f = 1$ ) menunjukkan konsistensi persamaan ini (?).

Mekanisme eksitasi terjadi ketika atom menyerap energi dari sumber eksternal (e.g., plasma), menyebabkan elektron berpindah ke tingkat energi lebih tinggi. De-eksitasi menghasilkan emisi foton dengan panjang gelombang:

$$\lambda = \frac{hc}{\Delta E}, \quad (2.3)$$

di mana  $h$  adalah konstanta Planck dan  $c$  kecepatan cahaya. Aturan seleksi  $\Delta l = \pm 1$  (?) membatasi transisi yang diperbolehkan, menghasilkan pola spektrum unik untuk setiap unsur.

Aplikasi praktis meliputi:

- Pemantauan polutan udara via spektroskopi laser resolusi tinggi (?),
- Analisis komposisi bintang menggunakan spektrograf astronomi (?),

- Deteksi logam berat dengan teknik LIBS (*Laser-Induced Breakdown Spectroscopy*).

Intensitas garis spektrum bergantung pada probabilitas transisi dan populasi elektron yang dapat dimodelkan dengan distribusi Boltzmann (?).

**Mekanisme eksitasi dan de-eksitasi atom.**

**Panjang gelombang emisi dan hubungannya dengan identifikasi unsur.**

### **2.1.2 Hukum Beer-Lambert**

**Hukum Beer-Lambert dan aplikasinya dalam analisis kuantitatif.**

**Absorbansi, transmitansi, dan konsentrasi analit.**

**Keterbatasan Hukum Beer-Lambert.**

### **2.1.3 Instrumentasi Spektroskopi Emisi**

**Sumber eksitasi (misalnya, plasma, api, arc).**

**Sistem dispersi cahaya (misalnya, prisma, kisi difraksi).**

**Detektor (misalnya, *photomultiplier tube* (PMT), *charge-coupled device* (CCD)).**

**Jenis-jenis spektrometer (misalnya, *Czerny-Turner*, *Echelle*) dan prinsip kerjanya.**

## **2.2 LASER-INDUCED BREAKDOWN SPECTROSCOPY (LIBS)**

### **2.2.1 Prinsip Dasar LIBS**

**Skema LIBS dan tahapan-tahapan dalam analisis LIBS.**

**Interaksi laser-materi: proses absorpsi energi dan peningkatan suhu.**

**Mekanisme ablasi laser: *thermal vaporization*, *photochemical decomposition*, *photophysical sputtering*.**

**Formasi plasma: ionisasi atom dan molekul, pembentukan spesies tereksitasi.**

**Emisi atomik: de-eksitasi radiatif dan emisi foton pada panjang gelombang karakteristik.**

**Simulasi Spektrum Emisi:** Penjelasan tentang bagaimana spektrum emisi dihasilkan dari plasma.

Persamaan intensitas emisi:  $I_{ij} = A_{ij} \cdot N_i \cdot h \cdot \nu_{ij}$ , dimana  $I_{ij}$  adalah intensitas emisi,  $A_{ij}$  adalah probabilitas transisi,  $N_i$  adalah populasi atom pada tingkat energi  $i$ ,  $h$  adalah konstanta Planck, dan  $\nu_{ij}$  adalah frekuensi transisi.

Contoh simulasi spektrum emisi dengan Manim:

**Akuisisi dan analisis spektrum: identifikasi unsur dan kuantifikasi konsentrasi.**

### 2.2.2 Karakteristik Plasma LIBS

**Parameter plasma: suhu, densitas elektron, komposisi spesies.**

**Kesetimbangan Termodinamika Lokal (LTE): definisi dan kriteria pemenuhan.**

**Efek Stark: pergeseran dan pelebaran garis spektral akibat medan listrik dalam plasma.**

**Efek matriks: pengaruh komposisi sampel terhadap proses ablasi dan emisi.**

### 2.2.3 Laser Nd:YAG dalam LIBS

**Prinsip kerja laser Nd:YAG: media aktif, rongga resonator, proses *pumping*.**

**Karakteristik laser Nd:YAG: panjang gelombang (1064 nm, 532 nm, 355 nm, 266 nm), energi pulsa, durasi pulsa (nanosecond, picosecond, femtosecond), mode operasi (*Q-switched, mode-locked*).**

**Diagram Grotrian Nd:YAG:** Tingkat energi ion  $\text{Nd}^{3+}$  dalam kristal YAG ( $4F_{3/2}$ ,  $4I_{11/2}$ , dll.).

Transisi elektronik yang menghasilkan emisi laser pada 1064 nm ( $4F_{3/2} \rightarrow 4I_{11/2}$ ).

Mekanisme *pumping* dan proses relaksasi non-radiatif.

**Keuntungan menggunakan laser Nd:YAG dalam LIBS: ketersediaan, keandalan, fleksibilitas panjang gelombang.**

**Pertimbangan dalam menggunakan laser Nd:YAG untuk CF-LIBS: validitas asumsi LTE.**

#### **2.2.4 Calibration-Free LIBS (CF-LIBS)**

**Motivasi CF-LIBS:** mengatasi keterbatasan kalibrasi standar (efek matriks, kesulitan mendapatkan standar yang sesuai).

**Prinsip dasar CF-LIBS:** penggunaan persamaan Boltzmann dan Saha untuk menghubungkan intensitas emisi dengan konsentrasi unsur.

**Metode CF-LIBS:** *one-line, multi-line, iterative.*

**Asumsi LTE dalam CF-LIBS dan validitasnya pada plasma yang dihasilkan oleh laser Nd:YAG.**

**Keunggulan CF-LIBS:** analisis kuantitatif tanpa standar, kecepatan, fleksibilitas.

**Tantangan CF-LIBS:** akurasi pengukuran parameter plasma, efek *self-absorption*.

#### **2.2.5 Aplikasi LIBS**

**Contoh aplikasi LIBS di berbagai bidang:** Industri: kontrol kualitas, analisis logam dan paduan, identifikasi material.

Lingkungan: pemantauan polusi, analisis tanah dan air, karakterisasi limbah.

Kedokteran: analisis jaringan biologis, deteksi kanker, diagnosis penyakit.

Arkeologi: analisis artefak, identifikasi pigmen, penentuan asal-usul material.

Forensik: analisis material bukti, identifikasi bahan peledak, analisis *gunshot residue*.

### **2.3 MODEL LONG SHORT-TERM MEMORY (LSTM)**

Long Short-Term Memory (LSTM) adalah jenis jaringan saraf yang dirancang untuk mengatasi tantangan dalam memproses dan memprediksi urutan data. LSTM sangat efektif dalam menangkap dependensi jangka panjang dalam data sekuensial, yang membuatnya berguna dalam berbagai aplikasi, termasuk analisis deret waktu, pemrosesan bahasa alami, dan pengenalan suara. Dengan kemampuannya untuk mengingat informasi dalam jangka waktu yang lebih lama, LSTM telah menjadi alat penting dalam bidang pembelajaran mesin (?).

LSTM telah terbukti menjadi alat yang sangat efektif dalam berbagai aplikasi pembelajaran mesin. Dengan kemampuannya untuk menangkap dependensi jangka panjang dalam data sekuensial, LSTM telah membuka banyak peluang baru dalam analisis data dan pengembangan model prediktif. Meskipun ada tantangan yang perlu

diatasi, potensi LSTM dalam berbagai bidang menjadikannya salah satu teknik yang paling menarik dalam pembelajaran mesin (?).

Bagi para praktisi yang ingin mempelajari lebih lanjut tentang LSTM, disarankan untuk membaca literatur tambahan dan melakukan eksperimen dengan model. Menggunakan pustaka seperti Keras dapat mempermudah proses pengembangan dan pelatihan model. Selain itu, mengikuti perkembangan terbaru dalam penelitian LSTM dapat memberikan wawasan baru dan teknik yang lebih baik.

## **2.4 STRUKTUR DAN MEKANISME LSTM**

Struktur LSTM terdiri dari beberapa komponen kunci, termasuk sel memori dan tiga pintu: pintu input, pintu lupa, dan pintu output. Pintu input mengontrol informasi baru yang masuk ke dalam sel memori, pintu lupa menentukan informasi mana yang harus dihapus, dan pintu output mengatur informasi yang akan dikeluarkan dari sel memori. Mekanisme ini memungkinkan LSTM untuk mempertahankan informasi yang relevan dan mengabaikan informasi yang tidak penting, sehingga meningkatkan kinerja model dalam memprediksi urutan data (?).

Dalam analisis deret waktu, LSTM digunakan untuk memprediksi nilai masa depan berdasarkan data historis. Misalnya, dalam prediksi harga saham, model LSTM dapat dilatih menggunakan data harga historis untuk memprediksi harga di masa depan. Keunggulan LSTM dalam menangkap pola temporal membuatnya lebih unggul dibandingkan model tradisional, seperti regresi linier, yang sering kali gagal menangkap hubungan jangka panjang dalam data.

## **2.5 APLIKASI LSTM**

LSTM juga banyak digunakan dalam pemrosesan bahasa alami (NLP), di mana urutan kata dalam kalimat sangat penting. Dalam tugas-tugas seperti penerjemahan bahasa dan analisis sentimen, LSTM dapat digunakan untuk memahami konteks dan makna dari urutan kata. Dengan memanfaatkan kemampuan LSTM untuk mengingat informasi dari kata-kata sebelumnya, model dapat menghasilkan terjemahan yang lebih akurat dan analisis sentimen yang lebih tepat (?).

Dalam pengenalan suara, LSTM digunakan untuk mengubah sinyal audio menjadi teks. Model ini dilatih menggunakan data audio yang telah dilabeli untuk mengenali pola dalam suara dan menghasilkan transkripsi yang akurat. Dengan kemampuan LSTM untuk menangkap informasi temporal, model dapat mengenali kata-kata dalam konteks yang lebih luas, meningkatkan akurasi pengenalan suara.

## 2.6 PELATIHAN DAN EVALUASI MODEL LSTM

Pelatihan model LSTM melibatkan penggunaan algoritma optimasi untuk meminimalkan fungsi kerugian. Proses ini biasanya dilakukan dengan menggunakan teknik backpropagation melalui waktu (BPTT), yang memungkinkan model untuk memperbarui bobot berdasarkan kesalahan prediksi. Pemilihan hyperparameter, seperti jumlah unit LSTM dan tingkat dropout, juga sangat penting untuk mencapai kinerja optimal (?).

Evaluasi model LSTM dilakukan dengan menggunakan metrik seperti akurasi, presisi, dan recall, tergantung pada jenis masalah yang dihadapi. Untuk masalah regresi, metrik seperti Mean Squared Error (MSE) sering digunakan. Evaluasi yang tepat sangat penting untuk memastikan bahwa model dapat diandalkan dan memberikan hasil yang akurat pada data baru.

### 2.6.1 Input Layer

Input layer adalah komponen pertama dalam arsitektur LSTM yang bertanggung jawab untuk menerima data masukan. Dalam konteks analisis spektrum, data yang dimasukkan biasanya berupa serangkaian nilai intensitas yang diukur pada berbagai panjang gelombang. Data ini perlu diubah menjadi format tiga dimensi, yaitu (samples, timesteps, features), agar dapat diproses oleh model LSTM. Misalnya, jika kita memiliki 1000 sampel, 10 timestep, dan 5 fitur, maka bentuk inputnya adalah:

$$\text{Input Shape} = (1000, 10, 5)$$

Pengaturan yang tepat dari input layer sangat penting untuk memastikan bahwa model dapat belajar dari data dengan efektif ?.

Selain itu, input layer juga berfungsi untuk menstandarisasi dan menormalisasi data sebelum diproses lebih lanjut. Proses ini penting untuk menghindari masalah yang dapat muncul akibat skala data yang berbeda, yang dapat mempengaruhi kinerja model. Normalisasi data dapat dilakukan dengan berbagai metode, seperti Min-Max Scaling atau Z-score Normalization. Dengan memastikan bahwa data berada dalam rentang yang sesuai, model dapat belajar dengan lebih cepat dan efisien, serta mengurangi risiko konvergensi yang lambat selama pelatihan.

Konfigurasi untuk input layer dalam kode Python menggunakan Keras dapat dituliskan sebagai berikut:

```
model.add(InputLayer(input_shape=(timesteps, features)))
```

### 2.6.2 LSTM Layer

LSTM layer adalah inti dari model ini, yang dirancang untuk menangkap dependensi temporal dalam data sekuensial. Layer ini terdiri dari unit-unit LSTM yang memiliki kemampuan untuk menyimpan informasi dalam jangka waktu yang lebih lama dibandingkan dengan jaringan saraf tradisional. Setiap unit LSTM memiliki tiga pintu: pintu input, pintu lupa, dan pintu output, yang mengatur aliran informasi ke dalam dan keluar dari sel memori. Dengan mekanisme ini, LSTM dapat mengatasi masalah vanishing gradient yang sering terjadi pada jaringan saraf konvensional ?.

Mekanisme pintu dalam LSTM memungkinkan model untuk memutuskan informasi mana yang harus disimpan dan mana yang harus dilupakan. Pintu input mengontrol informasi baru yang masuk ke dalam sel memori, pintu lupa menentukan informasi mana yang harus dihapus dari sel memori, dan pintu output mengatur informasi yang akan dikeluarkan dari sel memori. Dengan cara ini, LSTM dapat mempertahankan informasi yang relevan untuk jangka waktu yang lebih lama, sehingga sangat efektif dalam aplikasi yang memerlukan pemahaman konteks temporal, seperti analisis deret waktu dan pemrosesan bahasa alami.

Konfigurasi untuk LSTM layer dalam kode Python menggunakan Keras dapat dituliskan sebagai berikut:

```
model.add(LSTM(64, return_sequences=True, input_shape=(timesteps, features)))
```

### 2.6.3 Dropout Layer

Dropout layer ditambahkan setelah LSTM layer untuk mengurangi risiko overfitting, yang merupakan masalah umum dalam pelatihan model dengan banyak parameter. Layer ini bekerja dengan cara mengabaikan sejumlah neuron secara acak selama proses pelatihan, sehingga model tidak terlalu bergantung pada neuron tertentu. Dengan demikian, dropout membantu meningkatkan generalisasi model terhadap data yang belum pernah dilihat sebelumnya ?.

Penentuan tingkat dropout yang tepat sangat penting untuk mencapai keseimbangan antara pelatihan yang efektif dan generalisasi yang baik. Biasanya, tingkat dropout berkisar antara 0.2 hingga 0.5, tergantung pada kompleksitas model dan ukuran dataset. Penelitian menunjukkan bahwa penggunaan dropout dapat secara signifikan meningkatkan akurasi model dalam berbagai aplikasi, termasuk analisis spektrum. Dengan mengurangi overfitting, model dapat memberikan prediksi yang lebih akurat dan dapat diandalkan ketika diterapkan pada data baru.

Konfigurasi untuk dropout layer dalam kode Python menggunakan Keras dapat dituliskan sebagai berikut:

```
model.add(Dropout(0.3))
```

#### 2.6.4 Dense Layer

Dense layer, atau fully connected layer, berfungsi untuk mengolah informasi yang diekstrak oleh LSTM layer dan menghasilkan output akhir dari model. Dalam konteks LSTM, dense layer biasanya digunakan setelah LSTM layer dan dropout layer. Layer ini menghubungkan setiap neuron di layer sebelumnya dengan setiap neuron di layer berikutnya, memungkinkan model untuk belajar dari kombinasi fitur yang lebih kompleks. Dengan cara ini, dense layer dapat menangkap interaksi yang lebih dalam antara fitur-fitur yang ada dalam data ?.

Fungsi aktivasi yang umum digunakan dalam dense layer adalah ReLU (Rectified Linear Unit) atau tanh. Pemilihan fungsi aktivasi yang tepat dapat mempengaruhi kinerja model secara signifikan. Setelah dense layer, biasanya terdapat satu output layer yang menghasilkan prediksi akhir, baik untuk regresi maupun klasifikasi. Dengan mengoptimalkan arsitektur dense layer, model dapat mencapai kinerja yang lebih baik dalam memprediksi hasil yang diinginkan, sehingga meningkatkan akurasi dan efisiensi dalam analisis data.

Konfigurasi untuk dense layer dalam kode Python menggunakan Keras dapat dituliskan sebagai berikut:

```
model.add(Dense(1024, activation='tanh'))
```

Simulasi spektrum adalah alat penting dalam analisis spektroskopi, yang memungkinkan peneliti untuk memprediksi intensitas cahaya yang dipancarkan atau diserap oleh suatu substansi pada berbagai panjang gelombang. Kelas `SpectrumSimulator` dirancang untuk melakukan simulasi ini dengan mempertimbangkan berbagai parameter fisik, termasuk tingkat energi, degenerasi, dan koefisien Einstein.

### 2.7 STRUKTUR KELAS

Kelas `SpectrumSimulator` memiliki beberapa atribut penting, termasuk `nist_data`, `temperature`, dan `resolution`. Atribut `nist_data` berisi data dari National Institute of Standards and Technology (NIST), yang mencakup informasi tentang tingkat energi dan koefisien transisi. Atribut `temperature` menentukan suhu dalam Kelvin, dan `resolution` mengatur resolusi spektrum yang dihasilkan.

### 2.8 METODE UTAMA

Kelas ini memiliki beberapa metode statis dan non-statis yang berfungsi untuk menghitung berbagai aspek dari simulasi spektrum:



### 2.8.1 Fungsi Partisi

Metode `partition_function` menghitung fungsi partisi

$$Z$$

berdasarkan tingkat energi dan degenerasi. Fungsi partisi adalah konsep penting dalam statistik termal yang digunakan untuk menghitung probabilitas distribusi keadaan dalam sistem termal. Fungsi ini didefinisikan sebagai:

$$Z = \sum_i g_i e^{-\frac{E_i}{k_B T}}$$

di mana

$$g_i$$

adalah degenerasi,

$$E_i$$

adalah tingkat energi,

$$k_B$$

adalah konstanta Boltzmann, dan

$$T$$

adalah suhu (?).

### 2.8.2 Menghitung Intensitas

Metode `calculate_intensity` digunakan untuk menghitung intensitas spektrum berdasarkan suhu, energi, degenerasi, dan koefisien Einstein. Intensitas dapat dihitung dengan rumus:

$$I = \frac{g \cdot e^{-\frac{E}{k_B T}} \cdot A}{Z}$$

di mana

$$I$$

adalah intensitas,

$$g$$

adalah degenerasi,

$$E$$

adalah energi,

$$A$$

adalah koefisien Einstein, dan

$$Z$$

adalah fungsi partisi (?).

## 2.9 PROFIL VOIGT DALAM ANALISIS SPEKTRUM

Profil Voigt adalah gabungan dari dua fungsi profil spektral: profil Lorentzian dan Gaussian. Fungsi Voigt sering digunakan untuk mendeskripsikan bentuk garis spektrum yang diukur dalam berbagai teknik spektroskopi, termasuk Laser Induced Breakdown Spectroscopy (LIBS). Fungsi Voigt dapat didefinisikan sebagai konvolusi dari fungsi Lorentzian dan Gaussian.

### 2.9.1 Profil Lorentzian

Fungsi Lorentzian,  $L(\lambda)$ , menggambarkan lebar garis spektrum yang disebabkan oleh efek tekanan atau damping. Dalam konteks panjang gelombang  $\lambda$ , fungsi Lorentzian dapat dinyatakan sebagai:

$$L(\lambda) = \frac{\Gamma/2\pi}{(\lambda - \lambda_0)^2 + (\Gamma/2)^2} \quad (2.4)$$

di mana  $\lambda$  adalah panjang gelombang yang diukur,  $\lambda_0$  adalah panjang gelombang pusat dari garis spektrum, dan  $\Gamma$  adalah lebar garis Lorentzian yang berhubungan dengan lebar dari garis spektrum akibat efek tekanan. Dalam hal ini,  $\Gamma$  adalah Full Width at Half Maximum (FWHM) dari profil Lorentzian.

### 2.9.2 Profil Gaussian

Fungsi Gaussian,  $G(\lambda)$ , menggambarkan lebar garis spektrum yang disebabkan oleh efek Doppler, yang terkait dengan pergerakan relatif antara sumber spektrum dan detektor. Fungsi Gaussian dalam konteks panjang gelombang  $\lambda$  dinyatakan sebagai:

$$G(\lambda) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(\lambda - \lambda_0)^2}{2\sigma^2}\right) \quad (2.5)$$

di mana  $\sigma$  adalah deviasi standar dari distribusi Gaussian, yang berkaitan dengan lebar dari garis spektrum dalam konteks efek Doppler. Deviasi standar  $\sigma$  berhubungan dengan Half Width at Half Maximum (HWHM) Gaussian, yang dinyatakan sebagai  $b = \sigma\sqrt{2 \ln 2}$ .

### 2.9.3 Konvolusi Gaussian dan Lorentzian

Fungsi Voigt,  $V(\lambda; \Gamma, \sigma)$ , adalah hasil konvolusi dari fungsi Lorentzian dan Gaussian. Konvolusi ini menghasilkan profil spektrum yang menggabungkan kedua efek. Fungsi Voigt didefinisikan sebagai:

$$V(\lambda; \Gamma, \sigma) = \int_{-\infty}^{\infty} G(\lambda - \lambda') L(\lambda') d\lambda' \quad (2.6)$$

di mana  $\Gamma$  adalah lebar garis Lorentzian (FWHM), dan  $\sigma$  adalah deviasi standar Gaussian (HWHM).

Fungsi Voigt dapat disederhanakan dengan menggunakan fungsi  $W$  (fungsi Voigt) yang merupakan integral konvolusi dari fungsi Lorentzian dan Gaussian. Fungsi Voigt dapat dinyatakan sebagai:

$$V(\lambda; a, b) = \text{Re} \left[ W \left( \frac{\lambda - \lambda_0 + ia}{b} \right) \right] \quad (2.7)$$

di mana  $a$  adalah lebar Lorentzian (FWHM) dan  $b$  adalah deviasi standar Gaussian (HWHM). Fungsi Voigt kompleks,  $W(z)$ , didefinisikan sebagai:

$$W(z) = e^{-z^2} (1 + \text{erfi}(z)) \quad (2.8)$$

dengan  $\text{erfi}(z)$  sebagai fungsi kesalahan kompleks.

## 2.10 SIMULASI SPEKTRUM

Metode `simulate` menggabungkan semua komponen di atas untuk menghasilkan spektrum. Metode ini pertama-tama menginisialisasi panjang gelombang dan intensitas, kemudian mengumpulkan tingkat energi dan degenerasi dari data NIST. Setelah menghitung fungsi partisi, intensitas dihitung untuk setiap transisi dan digabungkan menggunakan profil Gaussian. Hasil akhirnya adalah panjang gelombang dan intensitas yang dinormalisasi.

Kelas `SpectrumSimulator` menyediakan alat yang kuat untuk simulasi spektrum berdasarkan data fisik yang relevan. Dengan menggunakan metode statistik dan fisika dasar, kelas ini memungkinkan peneliti untuk memodelkan perilaku spektrum dengan akurasi yang tinggi. Simulasi ini dapat diterapkan dalam berbagai bidang, termasuk kimia, fisika, dan ilmu material.

## **BAB III**

### **METODE PENELITIAN**

#### **3.1 WAKTU DAN LOKASI PENELITIAN**

Penelitian ini akan bertempat pada beberapa ruangan yang digunakan oleh mahasiswa Jurusan Informatika USK yang umumnya terletak pada lantai 3 blok A dan blok E Gedung Fakultas MIPA USK. Waktu yang dibutuhkan agar penelitian ini dapat di implementasikan adalah 4 bulan terhitung dari bulan Januari 2024 hingga Mei 2024.

#### **3.2 ALAT DAN BAHAN**

Alat dan Bahan yang akan digunakan pada penelitian ini terdiri dari beberapa perangkat keras (*hardware*) dan perangkat lunak (*software*) yang dijabarkan sebagai berikut:

##### **1. Perangkat Keras**

- Laptop Apple Macbook Air M1 2020 dengan RAM 8 *Gigabyte*.

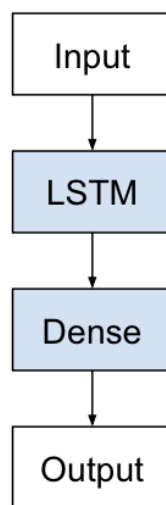
##### **2. Perangkat Lunak**

- MacOS (18.1)
- Jupyter
- Python 3.8.17
- NumPy (1.24.0)
- SQLite3 (3.39.4)
- Pandas (1.5.3)
- Joblib (1.2.0)
- scikit-learn (1.2.0)
- TensorFlow (2.11.0)
- Keras (2.11.0)
- Matplotlib (3.6.2)
- TensorFlow Keras Layers (2.11.0)
- itertools (built-in)

### 3.3 METODE PENELITIAN

Dalam pengembangan model Long Short-Term Memory (LSTM), persiapan data yang tepat adalah langkah krusial yang dapat mempengaruhi kinerja model secara signifikan. Bab ini membahas berbagai teknik untuk mempersiapkan data numerik dan kategorikal, serta bagaimana menangani urutan dengan panjang yang bervariasi.

Gambar ?? menunjukkan arsitektur dasar dari LSTM. Dalam arsitektur ini, terdapat tiga komponen utama: input layer, hidden LSTM layer, dan output layer.



Gambar 3.1 LSTM Data kategorikal perlu diubah menjadi format numerik agar dapat digunakan dalam model LSTM. Dua metode umum untuk melakukan ini adalah One-Hot Encoding dan Label Encoding. One-Hot Encoding mengubah setiap kategori menjadi vektor biner, sedangkan Label Encoding memberikan setiap kategori nilai integer (?)

### 3.4 PERSIAPAN DATA NUMERIK

Data numerik memerlukan normalisasi atau standardisasi sebelum digunakan dalam model LSTM. Normalisasi membantu dalam mengurangi skala data, sehingga mempercepat proses pelatihan dan meningkatkan konvergensi model. Metode umum yang digunakan termasuk Min-Max Scaling dan Z-score Normalization (?).

#### 3.4.1 Normalisasi

Normalisasi data dilakukan dengan rumus:

$$I' = \frac{I - I_{min}}{I_{max} - I_{min}}$$

di mana

$$I'$$

adalah data yang dinormalisasi,

$$I$$

adalah data asli,

$$I_{min}$$

dan

$$I_{max}$$

adalah nilai minimum dan maksimum dari dataset.

### 3.5 PERSIAPAN DATA KATEGORIKAL

Data kategorikal perlu diubah menjadi format numerik agar dapat digunakan dalam model LSTM. Dua metode umum untuk melakukan ini adalah One-Hot Encoding dan Label Encoding. One-Hot Encoding mengubah setiap kategori menjadi vektor biner, sedangkan Label Encoding memberikan setiap kategori nilai integer (?).

#### 3.5.1 One-Hot Encoding

One-Hot Encoding dapat dilakukan dengan menggunakan pustaka seperti scikit-learn. Misalnya, untuk mengubah kolom kategorikal menjadi format one-hot:

```
from sklearn.preprocessing import OneHotEncoder
encoder = OneHotEncoder()
encoded_data = encoder.fit_transform(data).toarray()
```

### 3.6 MENANGANI URUTAN DENGAN PANJANG BERVARIASI

Dalam banyak kasus, urutan data yang digunakan dalam LSTM memiliki panjang yang bervariasi. Untuk menangani ini, kita dapat menggunakan padding untuk menambahkan nilai nol ke urutan yang lebih pendek, sehingga semua urutan memiliki panjang yang sama. Keras menyediakan fungsi `pad_sequences()` untuk tujuan ini.

#### 3.6.1 Padding

Padding dapat dilakukan sebagai berikut:

```
from keras.preprocessing.sequence import pad_sequences
padded_sequences = pad_sequences(sequences, maxlen=max_length)
```

### 3.7 PREDIKSI URUTAN SEBAGAI PEMBELAJARAN TERAWASI

Dalam konteks LSTM, prediksi urutan dapat dipandang sebagai masalah pembelajaran terawasi. Ini melibatkan transformasi data urutan menjadi format yang dapat digunakan untuk pelatihan model, di mana input dan output ditentukan berdasarkan langkah waktu tertentu (?).

Persiapan data yang tepat adalah langkah penting dalam pengembangan model LSTM. Dengan menggunakan teknik yang tepat untuk normalisasi, encoding, dan padding, kita dapat meningkatkan kinerja model dan memastikan bahwa data siap untuk analisis lebih lanjut.

$$\frac{n_i \cdot n_e}{n_a} = \left( \frac{2}{\Lambda^3} \right) \cdot \left( \frac{g_i}{g_a} \right) \cdot \exp \left( -\frac{E_{ion}}{kT} \right) \quad (3.1)$$

Program 3.1 Contoh Fungsi Python

```
1 def fibonacci(n):
2     """Menghitung deret Fibonacci"""
3     a, b = 0, 1
4     for _ in range(n):
5         print(a, end=' ')
6         a, b = b, a + b
7     return a
```

Program 3.2 Ini adalah sebuah teorema. Ini adalah sebuah teorema. Ini adalah sebuah teorema. Ini adalah sebuah teorema. Implementasi Algoritma Dijkstra

```
1 import tensorflow as tf
2 from tensorflow.keras import layers, models
3 from sklearn.model_selection import train_test_split
4 from utils import load_libs_data
5 from preprocessing.normalize import minmax_normalize
6
7 # Muat data
8 spectra, labels = load_libs_data('data/raw')
9 spectra = np.array([minmax_normalize(s) for s in spectra])
10
11 # Encode label
12 label_to_id = {label: i for i, label in enumerate(np.unique(
    labels))}
```

```

13 y = np.array([label_to_id[label] for label in labels])
14
15 # Split data
16 X_train, X_test, y_train, y_test = train_test_split(spectra, y
    , test_size=0.2)
17
18 # Bangun model 1D CNN
19 model = models.Sequential([
20     layers.Input(shape=(X_train.shape[1], 1)),
21     layers.Conv1D(32, kernel_size=5, activation='relu'),
22     layers.MaxPooling1D(2),
23     layers.Conv1D(64, kernel_size=3, activation='relu'),
24     layers.GlobalAveragePooling1D(),
25     layers.Dense(len(label_to_id), activation='softmax')
26 ])
27
28 model.compile(
29     optimizer='adam',
30     loss='sparse_categorical_crossentropy',
31     metrics=['accuracy']
32 )
33
34 # Pelatihan
35 history = model.fit(
36     X_train[..., np.newaxis], # Tambah dimensi channel
37     y_train,
38     epochs=50,
39     batch_size=32,
40     validation_data=(X_test[..., np.newaxis], y_test)
41 )
42
43 # Simpan model
44 model.save('models/saved_models/lib_cnn_1d.h5')

```

**Teorema 3.7.1.** *Ini adalah sebuah teorema.*

**Definisi 3.7.1.** *Ini adalah definisi.*

*Catatan.* *Ini adalah catatan tanpa nomor.*

*Proof.* *Ini adalah bukti.*

□



**Contoh 3.7.1. f**

Tabel 3.1 Perbandingan Performa Model Performa Model Performa Model Performa Model Machine Learning pada Data LIBS

Model	Akurasi (%)	Presisi (%)	Recall (%)	RMSE
Random Forest	95.2	94.8	95.1	0.12
SVM	89.7	88.5	90.2	0.21
Transformer	97.1	96.9	97.0	0.08
CNN	93.4	92.7	93.5	0.15

*Keterangan:* Data diperoleh dari 100 sampel logam dengan 5 kelas komposisi.