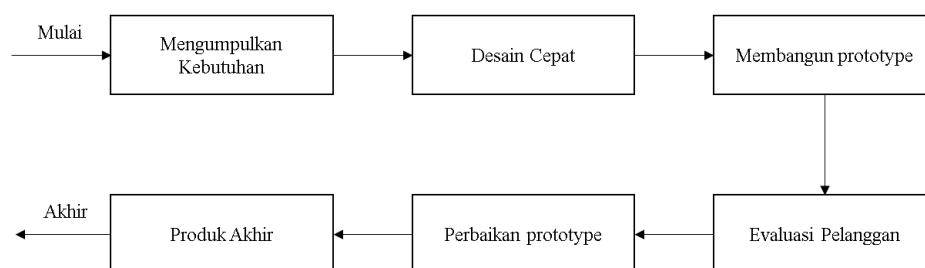


BAB III METODE PENELITIAN

3.1 Desain Penelitian

Metode rancang bangun digunakan dalam penelitian ini. Menurut Pressman, metode rancang bangun adalah serangkaian prosedur untuk menerjemahkan hasil analisa ke bahasa pemrograman untuk mendeskripsikan secara mendalam mengenai komponen – komponen yang hendak diimplementasikan. Implementasi memiliki maksud untuk menciptakan atau memperbaiki sistem yang telah ada, baik secara keseluruhan maupun sebagian saja (Pressman, 2002).

Model SDLC (*Software Development Life Cycle*) yang digunakan adalah model *prototyping*. Model *prototyping* memuat 6 tahapan. Tahapannya dapat dilihat pada Gambar 3.1.



Gambar 3.1 Model SDLC Prototyping

Pada Gambar 3.1 (Seema Suresh Kute & Surabhi Deependra Thorat, 2017), pengumpulan segala informasi dan kebutuhan lainnya yang berkaitan dengan pembuatan *machine learning* pada sistem RSYS dilakukan pada tahapan pertama. Setelah semua kebutuhan dikumpulkan dan diidentifikasi, proses desain perangkat lunak *machine learning* dilakukan dengan cepat. Kemudian, pembangunan prototype dilakukan dengan melakukan *coding* (menerjemahkan rancangan ke dalam bentuk perangkat lunak). Setelah terbentuk, proses dilanjutkan ke tahapan evaluasi dari pelanggan. Evaluasi dilakukan secara *alpha-beta* (Mohd & Shahbodin, 2015), sehingga pada tahapan selanjutnya dilakukan perbaikan pada hasil evaluasi atau pengujian,

yang pada akhirnya menghasilkan produk sebenarnya (*actual product*) untuk digunakan oleh *user*.

3.2 Partisipan Penelitian

Partisipan yang dilibatkan dalam penelitian ini meliputi 3 mahasiswa Sarjana dan 6 mahasiswa Pascasarjana. Pada jenjang Sarjana, mahasiswa berasal dari kampus UNIKOM (Universitas Komputer Indonesia) dengan jurusan Teknik Informatika yang sedang melaksanakan Skripsi. Dan untuk Mahasiswa Pascasarjana berjurusan PTK (Pendidikan Teknik dan Kejuruan), Universitas Pendidikan Indonesia.

Alasan penulis memilih 9 partisipan di dua kampus yang berbeda dikarenakan hal – hal teknis yang berasal dari mahasiswa ataupun dosen, diantaranya:

1. Keterbatasan penulis memperoleh akses ke mahasiswa di kampus UPI yang sedang bimbingan skripsi atau thesis.
2. Adanya kendala eksternal dari beberapa pihak dosen di kampus UPI yang menyebabkan data proposal thesis mahasiswanya hilang.
3. Penulis lebih memilih mahasiswa yang memiliki perizinan legal (melalui metode tanya jawab) dari orang – orang terdekat, seperti teman dan rekan kerja.
4. Penulis lebih memilih kampus UNIKOM karena berdasarkan wawancara terhadap salah satu mahasiswa yang sedang melaksanakan skripsi, secara mayoritas, metode revisi yang dilakukan oleh masing – masing dosennya menggunakan fitur *Comment* dalam *Mircosoft Word*.
5. Dan yang terakhir, mahasiswa PTK UPI memperoleh komentar revisi melalui teks via media sosial, salah satunya seperti WhatsApp (*Personal Chat*).

3.3 Kebutuhan *Software* dan *Hardware*

a. Kebutuhan *Software*

Perangkat lunak (*software*) yang digunakan untuk mendukung penelitian ini meliputi:

1. *Laravel Framework 5.4.36*

Digunakan untuk pembuatan website/*interface* berbasis bahasa pemrograman PHP (*Hypertext Preprocessor*).

2. *Python 3.7.4*

Digunakan untuk keperluan pembuatan proses *machine learning*.

3. *WAMPP Server 3.1.4 64 bit*

Digunakan untuk keperluan eksekusi *Laravel Framework*. Didalamnya sudah termasuk fitur *phpMyAdmin/MySQL* untuk penyimpanan *data/database*.

4. *Microsoft Visual Studio Code*

Digunakan untuk pembuatan *script/code*/bahasa pemrograman PHP dan *Python*.

5. *Browser Mozilla Firefox*

Digunakan untuk mengeksekusi bahasa pemrograman PHP dalam bentuk website.

6. *Command Prompt Windows*

Digunakan untuk mengeksekusi bahasa pemrograman *Python*.

b. Kebutuhan *Hardware*

Dalam rangka mendukung pembangunan perangkat lunak berbasis *machine learning* diperlukan spesifikasi perangkat keras (*hardware*) yang disesuaikan. Spesifikasi PC/Laptop yang diperlukannya meliputi *processor* Intel® Core™ i7-8750H CPU @ 2.20GHz (12 CPU), Memori RAM 16384 MB, dan VGA NVIDIA GeForce GTX 1050 Ti 12145 MB, Merk LEGION, Lenovo.

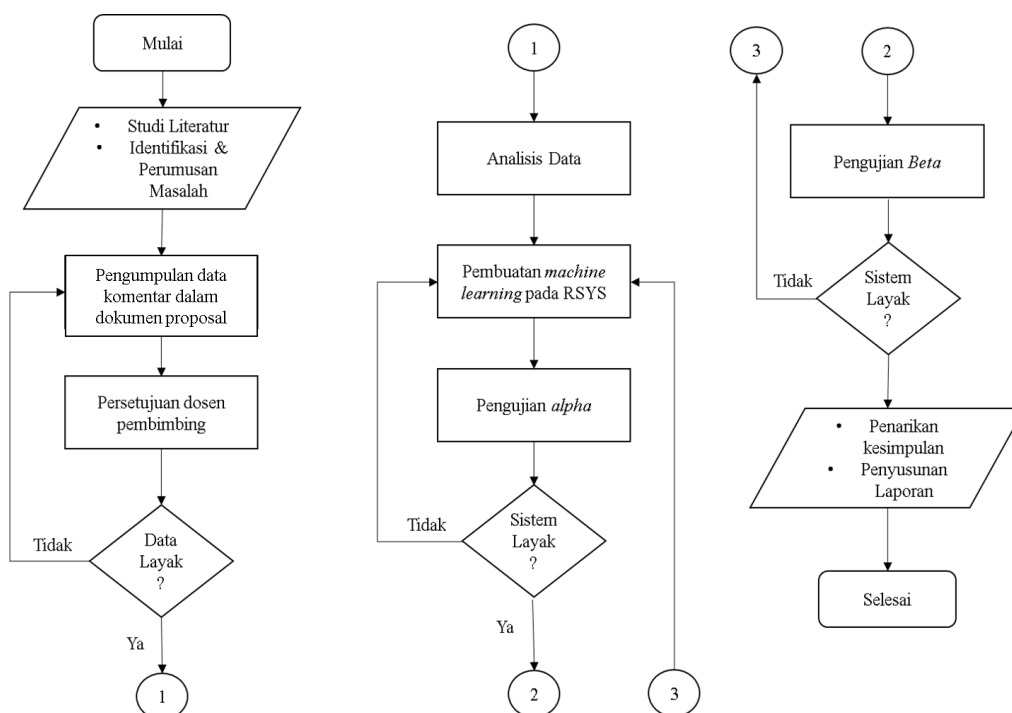
3.4 Instrumen Penelitian

Penulis mengumpulkan data dengan cara meminta, mendengar, serta mengambil (Afrizal, 2014). Teknik pengumpulan data menggunakan instrumen dokumentasi (Cooper et al., 2002). Instrumen dokumentasi digunakan oleh penulis untuk memperoleh dokumen – dokumen yang subjek penelitiannya berupa dokumen proposal bimbingan skripsi, tesis (Clemmens, 2003) (Herdiansyah, 2010) atau paper *conference*.

Dokumen proposal yang penulis peroleh dari mahasiswa UNIKOM berjumlah 32 Proposal. Dan dari mahasiswa Pascasarjana PTK berjumlah 15 Dokumen (termasuk Proposal dan *Paper Conference*) dalam format .docx (*Microsoft Word*). Dalam satu proposal memuat beberapa atau banyak komentar hasil revisi yang diberikan oleh dosen pembimbing. Dari 47 dokumen yang terkumpul, total keseluruhan komentar berjumlah 273 komentar. Keseluruhan komentar akan dibagi menjadi dua bagian, yaitu *dataset training* dan *dataset testing* dengan rasio yang sudah ditentukan, yaitu 9.5:0.05, 95% dari *dataset training* adalah 44.65, dan 5% dari *dataset testing* adalah 2.35. Maka dari itu, jika dibulatkan, *dataset training* yang digunakan 45 dokumen dan *dataset testing* yang digunakan 2 dokumen.

3.5 Prosedur Penelitian

Prosedur penelitian dapat dilihat pada Gambar 3.2. Tahapannya meliputi (1) melakukan studi literatur, mengidentifikasi & merumuskan masalah, (2) mengumpulkan data, (3) persetujuan dosen pembimbing, (4) analisis data, (5) pembuatan *machine learning* pada RSYS, (6) pengujian sistem secara alpha, (7) pengujian sistem secara beta, (8) penarikan kesimpulan dan penyusunan laporan. Berikut diagram flowchart prosedur penelitian dapat dilihat pada Gambar 3.2.



Gambar 3.2 Diagram *Flowchart* Prosedur Penelitian

Pada Gambar 3.2, studi literatur dilakukan untuk tujuan memahami segala apapun yang berkaitan dengan proses revisi dan *machine learning* itu sendiri. Sumber yang dimaksud berupa paper berjenis jurnal atau konferensi, bahkan informasi tambahan yang diperoleh melalui beberapa situs. Kemudian, tahapan dilanjutkan ke ruang lingkup masalah, yaitu identifikasi dan perumusan masalah. Setelah dirumuskan permasalahan, pengumpulan data dilakukan sesuai dengan instrumen yang ditentukan sebelumnya, yaitu melalui instrumen dokumentasi. Setiap dokumen yang diperoleh dari proses pengumpulan dicek dan disetujui oleh dosen pembimbing. Jika tidak disetujui, maka proses pengumpulan data dilakukan kembali sesuai dengan arahan dari dosen pembimbing. Jika disetujui, dokumen layak dan dilanjutkan ke proses analisis data.

Analisis data dilakukan untuk memberikan spesifikasi data atau alur data yang dibutuhkan sebelum melakukan pembuatan *machine learning*. Kemudian, proses dilanjutkan ke tahapan pembuatan machine learning. Pembuatan *machine learning* disesuaikan dengan data yang sudah dianalisis

sebelumnya sampai menghasilkan suatu produk sistem. Produk sistem itulah yang selanjutnya akan dilakukan pengujian secara *alpha* dan *beta*.

Pengujian *alpha* dilakukan oleh tim pengembang (Suandi et al., 2017). Tim pengembang disini yaitu penulis sendiri. Sebelum dirilis, baik fungsionalitas pada sistem perangkat lunak *interface (framework laravel)* dalam bentuk website ataupun proses *machine learning (python)* dalam bentuk CLI (*Command Line Interface*) penulis uji coba sebelum dirilis kepada pengujian secara *beta*.

Lalu, setelah pengujian *alpha* selesai, selanjutnya, pengujian dilakukan oleh pihak eksternal, yaitu pihak yang tidak ada keterkaitan dengan pengembangan perangkat lunak, yang disebut dengan pengujian secara *beta* (Suandi et al., 2017). Pengujian tersebut dilakukan oleh mahasiswa pascasarjana yang sedang melaksanakan bimbingan thesis dan memperoleh salah satu atau beberapa revisi dalam bentuk komentar di dalam aplikasi *Microsoft word* melalui fitur *comment*, yang berjurusan Pendidikan Teknologi Kejuruan (PTK). Tujuan dari dilakukannya pengujian *beta* adalah untuk memperoleh kesalahan / *error* yang dirasakan ketika menggunakan perangkat lunak RSYS berbasis *machine learning*. Sehingga, penulis sebagai tim internal dapat terus menerus memperbaiki hingga tahapan pengujian *beta* dinyatakan selesai dilakukan.

Pengujian akhir secara *beta* menandakan bahwa kesimpulan dapat ditarik dan disusun dalam laporan. Penarikan kesimpulan dan penyusunan laporan dibuat berdasarkan tujuan yang hendak dicapai dan fakta yang ditemukan dari hasil penelitian ini.

3.5.1 Analisis Kebutuhan Perangkat Lunak *Machine Learning* RSYS

3.5.1.1 Desain Aplikasi

Desain aplikasi merupakan representasi dari analisis kebutuhan yang sudah dilakukan sebelumnya pada tahapan analisis kebutuhan. Proses desain aplikasi disesuaikan dengan kebutuhan sistem *machine learning* RSYS.

3.5.1.2 Implementasi Aplikasi

Implementasi merupakan perwujudan dari proses desain yang telah dilakukan (produk nyata). Sistem RSYS berbasis *machine learning* diimplementasikan sesuai dengan desain yang sudah ditentukan.

3.5.2 Pengujian/Tes Aplikasi

3.5.2.1 Pengujian *Alpha*

Pengujian *alpha* itu sendiri memiliki maksud untuk menguji dalam ruang lingkup *internal*. Artinya, pengujian dilakukan tanpa melibatkan pihak eksternal / bukan tim pengembang aplikasi.

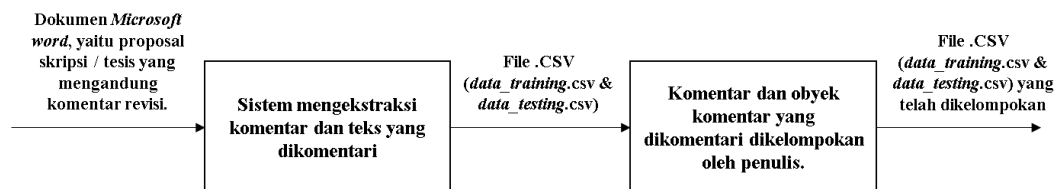
3.5.2.2 Pengujian *Beta*

Pengujian *beta* adalah pengujian secara eksternal, yaitu dilakukan setelah proses pengujian *alpha* selesai, melibatkan user sesungguhnya untuk menggunakan sistem RSYS (Mohd & Shahbodin, 2015).

Setelah tahapan pengujian *Alpha* dan *Beta* dilaksanakan, tahapan dilanjutkan ke penarikan kesimpulan dan penyusunan laporan. Kesimpulan memuat hasil akhir dari serangkaian proses yang telah dilalui, sehingga hal itu dapat dituangkan kedalam bentuk laporan atau penyusunan laporan.

3.6 Analisis Data

Teknik analisis data yang penulis gunakan adalah teknik analisis konten (*Content Analysis*). Teknik analisis konten dilakukan dengan cara mengelompokan / mengkategorikan konten secara sistematis (Shelton, 2015). Dalam penelitian ini, data masukan berupa dokumen proposal dengan format *Microsoft Word* dianalisis setiap data didalamnya, yaitu teks komentar dan obyek yang dikomentari. Berikut proses analisis data yang dilakukan dapat dilihat pada Gambar 3.3.

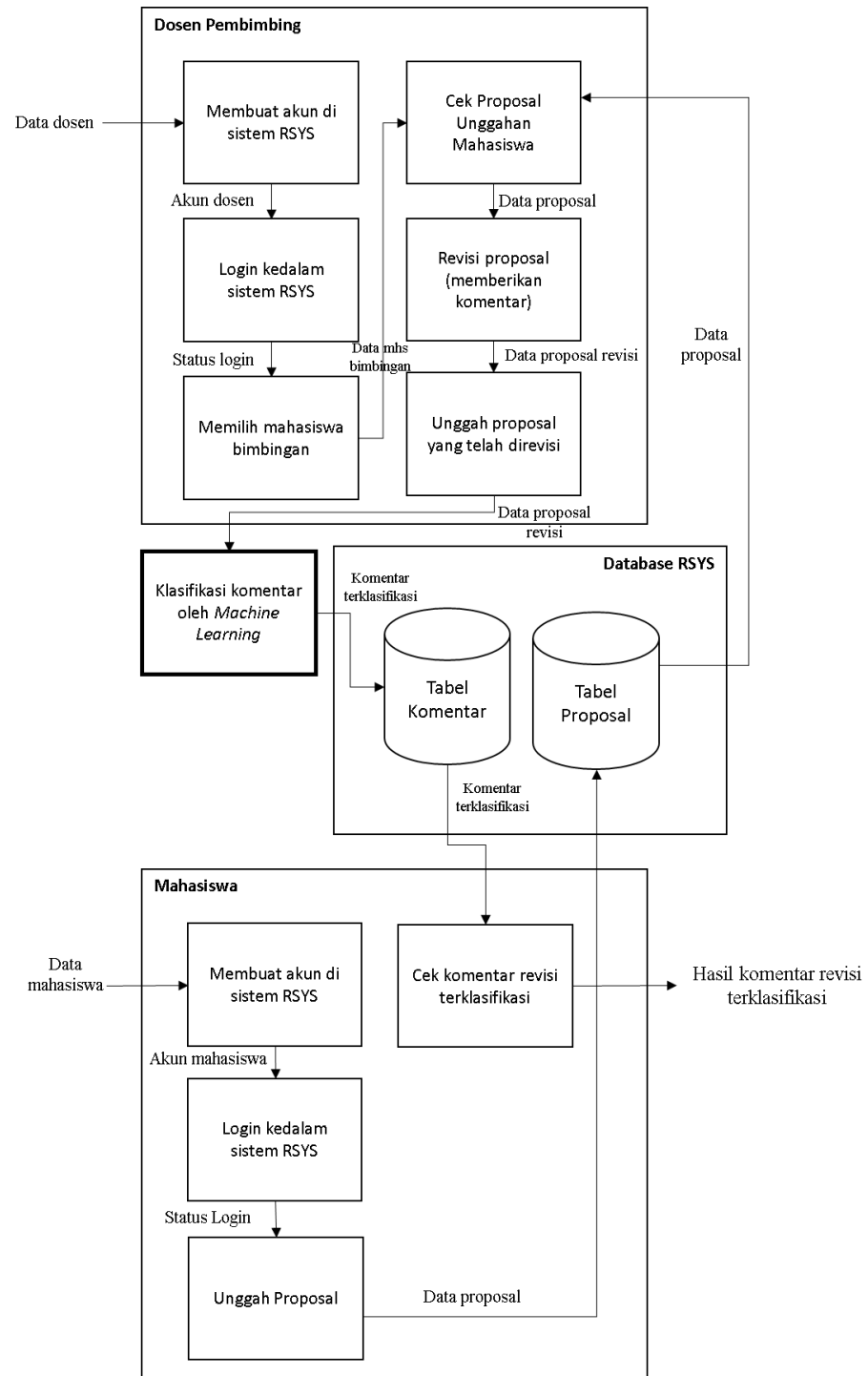


Gambar 3.3 Pengelompokan data komentar revisi

Berdasarkan pengamatan yang dilakukan pada setiap dokumen proposal, dalam satu dokumen proposal terdapat beberapa komentar, minimal 4 – 5 komentar. Seperti pada Gambar 3.3, setiap komentar dikelompokan setelah dianalisis oleh penulis, baik komentar maupun obyek yang dikomentarnya. Pengelompokan tersebut dilakukan dalam dokumen yang berformat CSV (*Comma Seperated Values*). Urgensi penggunaan dokumen CSV karena kebutuhan dalam proses *machine learning* yang digunakan oleh penulis. Pengelompokan tersebut dimaksudkan untuk memahami mesin, mana komentar yang lokal dan mana komentar yang global. Dalam arti lain adalah memberi label. Dokumen CSV *data_training* dan *data_testing* dapat dilihat pada lembar lampiran.

3.7 Arsitektur Sistem RSYS

Dalam mencapai kesempurnaan sistem RSYS yang diharapkan, seperti sistem yang dikehendaki dari mulai tampilan (*interface*) sampai dengan proses yang berjalan di belakang tampilan sistem, maka dibutuhkan suatu arsitektur yang jelas dan sistematis. Dalam point 3.7 ini, penulis memaparkan arsitektur sistem RSYS yang dibagi menjadi dua bagian, yang pertama arsitektur sistem secara keseluruhan, dan yang kedua arsitektur sistem pada proses *machine learning* yang berjalan di belakang tampilan untuk mengklasifikasikan komentar. Berikut arsitektur secara keseluruhan dapat dilihat pada Gambar 3.4.

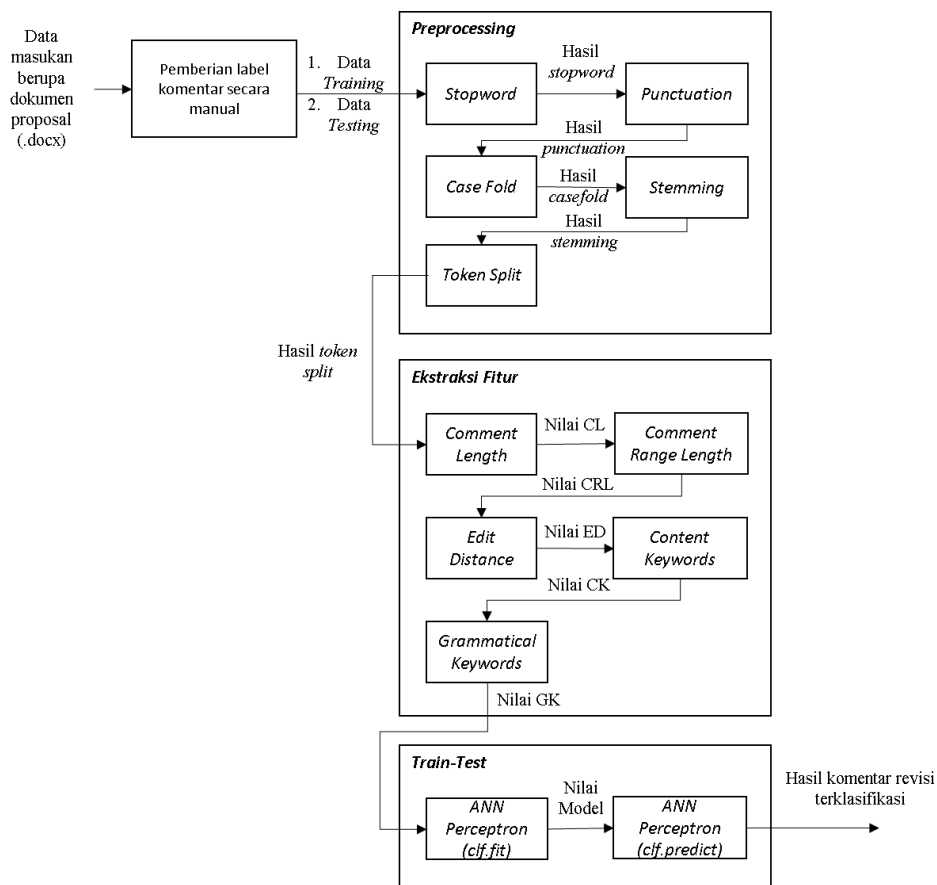


Gambar 3.4 Blok Diagram Arsitektur Sistem RSYS

Pada Gambar 3.4, awal tahapan dari semua proses yang berjalan pada sistem dimulai dari pembuatan akun dosen pembimbing. Dosen yang dipilih oleh pihak prodi untuk bertanggung jawab dalam membimbing wajib untuk

membuat akun dalam sistem RSYS. Setelah membuat akun, dosen pembimbing dipersilahkan untuk login / masuk kedalam sistem. Berhasil atau gagalnya login / masuk kedalam sistem ditandai dengan status login. Dari halaman dosen pembimbing, dosen pembimbing diharapkan langsung memilih mahasiswa yang hendak dibimbingnya sesuai arahan dari prodi. Setelah itu, dosen diharuskan mengecek proposal yang telah di unggah oleh mahasiswa yang telah melalui proses yang sama seperti dosen, yaitu membuat akun mahasiswa, login, dan mengunggah proposal yang telah dibuatnya. Sehingga, proposal yang telah di unggah oleh mahasiswa disimpan dalam *database* RSYS di tabel proposal.

Setelah data proposal sudah dicek oleh dosen pembimbing, dosen pembimbing mengunduh dokumen proposalnya dan memberikan revisi. Tentu, setiap revisi dalam bentuk komentar dan wajib memberikan komentar pada aplikasi *Microsoft Word* dengan fitur *Comment*. Setelah itu, proses dilanjutkan untuk mengunggah proposal hasil revisi kedalam sistem. Sistem RSYS akan menerima unggahan proposal revisi yang berisi komentar - komentar untuk diklasifikasikan / dikategorikan setiap komentarnya. Kategorinya antara lain adalah kategori komentar lokal dan global. Hasil komentar yang telah terklasifikasi tersebut disimpan dalam *database* RSYS di tabel komentar. Dan yang terakhir, mahasiswa dapat mengecek komentar revisi yang telah terklasifikasi oleh *machine learning*, seperti pada Gambar 3.4, yaitu Hasil klasifikasi komentar dalam bentuk perangkat lunak. Adapun arsitektur sistem klasifikasi komentar oleh *machine learning* dapat dilihat pada Gambar 3.5.



Gambar 3.5 Blok Diagram Arsitektur Sistem Klasifikasi Komentar

Pada Gambar 3.5, data masukan merupakan komentar – komentar yang terkandung di dalam dokumen proposal bimbingan dalam bentuk *file Microsoft Word* (dengan ekstensi .docx). Rasio dokumen proposal untuk pembagiannya terhadap *data training* dan *data testing* sebesar 9.5:0.05. Setiap komentar yang terkandung dalam dokumen proposal akan melalui proses *Preprocessing*.

Pada tahapan *preprocessing*, tahapannya tidak hanya meliputi *tokenization*, *sentence splitting*, dan *lemmatization* (H. N. Ocharo & Hasegawa, 2018), namun ada beberapa tambahannya, seperti *stopword / filtering* (Anantharaman et al., 2019), *punctuation* (Bulut et al., 2019), dan *case fold / lowercase* (Sangounpao & Muenchaisri, 2019). Pada penelitian ini, proses *stopword / filtering* dan *Stemming* menggunakan *library* Sastrawi. Sedangkan untuk *library* Sastrawi, tingkat akurasinya hanya 92% (Rosid et al., 2020b). Dan untuk *punctuation*, *token split*, dan *case fold*, penulis menggunakan fungsi

dari *python*, yaitu *string.punctuation()*, *split()* (Igal & Segui, 2017), dan *casefold()*.

Setelah data berbentuk token, proses dilanjutkan ke tahapan *feature engineering / feature selection / feature enrichment* (Alsmadi & Gan, 2020) / ekstraksi fitur. Fitur yang digunakan dalam penelitian meliputi *comment length* (panjang komentar), *comment range length* (panjang objek yang diberikan komentar), *edit distance* (pengukuran jarak antara teks komentar dan objek yang dikomentari), *content keywords* (kata kunci konten), *grammatical keywords* (kata kunci tata bahasa) (H. N. Ocharo & Hasegawa, 2018) . Panjang komentar (*comment length*) dan panjang objek yang dikomentari (*comment range length*) dilakukan menggunakan fungsi dari *python*, yaitu *len()*. Sedangkan, *edit distance* menggunakan *Jaro Winkle Distance* melalui library *Jellyfish*. Algoritma *Jaro Winkle Distance* memiliki tingkat *similarity* sebesar 80,92% dibandingkan dengan *Levensthein Distance* (H. N. Ocharo & Hasegawa, 2018) yang tingkat *similarity*nya sebesar 49,43% (Tannga et al., 2017). Dan untuk *content keywords* serta *grammatical keywords* diserupakan dengan apa yang digunakan pada penelitian Ocharo & Hashegawa (H. Ocharo & Hasegawa, 2016), hanya saja kata kuncinya dalam Bahasa Indonesia.

Kemudian, proses dilanjutkan ke tahapan pelatihan / *training* setelah diperolehnya nilai – nilai atau bobot – bobot (*weight*) dari proses ekstraksi fitur. Metode yang digunakan untuk *training* dan *testing* dalam penelitian ini adalah metode ANN (*Artificial Neural Network*) MLP (*MultiLayer Perceptron*) / Jaringan Syaraf Tiruan MLP. Proses *training* dilakukan menggunakan library *scikit-learn*. Dimana *output* dari proses *training* (*clf.fit*) itu sendiri menghasilkan *model*, yaitu bobot – bobot (*weight*) yang disertai dengan kelas 1 (*global revision*) atau 0 (*local revision*). Setelah *model* terbentuk, proses dilanjutkan ke tahapan *testing* (*clf.predict*). Proses *testing* dilakukan untuk menghasilkan komentar yang sudah terklasifikasi, yang berupa hasil komentar revisi terklasifikasi.