

## BAB 3 METODOLOGI PENELITIAN

Pada bagian ini akan dijelaskan langkah-langkah yang dilakukan dalam penyusunan dan pengerjaan penelitian yang dilakukan. Langkah-langkah akan dijelaskan dari awal hingga akhir penelitian.

### 3.1 Metodologi Penelitian

Berikut merupakan metodologi penelitian yang dilakukan dalam pengembangan *chatbot helpdesk* menggunakan metode *Token-Based Similarity Algorithm*, yaitu:

1. Observasi Observasi dilakukan untuk melihat alur kerja dari tim *helpdesk* perusahaan untuk mengetahui kekurangan pada saat menerima aduan masalah karyawan dan cara menyelesaikan permasalahan karyawan tersebut. Hasil observasi akan dijadikan bahan membuat *flowchart chatbot helpdesk*.
2. Studi Literatur Pada tahap ini akan berfokus dalam mencari referensi dan pembelajaran yang berkaitan dengan *chatbot helpdesk*, *Token-Based Similarity Algorithm*, serta informasi yang berkaitan untuk mendukung dalam penelitian. Studi literatur bisa dilakukan dengan mencari referensi dari buku, jurnal, artikel dan lain - lain.
3. Pembuatan Flowchart Pada tahap pembuatan Flowchart akan dilakukan pembuatan *flowchart* sebelum dan sesudah *chatbot helpdesk*. Pembuatan Flowchart ini untuk mengetahui cara kerja tim *Helpdesk* sebelum adanya *chatbot helpdesk* dan juga menjadi landasan pembuatan *chatbot helpdesk* untuk tidak keluar dari jalur yang sudah ditentukan.
4. Pembuatan Database Dalam pembuatan *database* akan dilakukan dengan memulai membuat *Entity Relationship Diagram*. Lalu membuat *database* yang menggunakan format CSV untuk membuat *database* yang sederhana dalam melakukan penelitian *chatbot helpdesk*.
5. Perancangan Chatbot Helpdesk Pada tahap perancangan *chatbot helpdesk* dimulai dari pembuatan struktur menggunakan *framework Next.js* untuk

*Natural Language Process (NLP)*, struktur *Token-Based Similarity Algorithm*, membuat *frontend chatbot helpdesk*.

6. Testing Pada tahap *testing* akan dilakukan dengan 2 cara, yaitu pertama melakukan testing input ke sistem secara manual dengan menggunakan proporsi 70:30 dan membuat *form black box testing* dan *white box testing* untuk dibagikan kepada tim *helpdesk* perusahaan.
7. Dokumentasi Pada tahap ini akan dilakukan dokumentasi dari hasil akhir penelitian *chatbot helpdesk*. Dokumentasi bisa berbentuk laporan yang menjelaskan dari awal proses penelitian hingga hasil dari penelitian.

### 3.2 Observasi

Sebelum dimulai proses penelitian, dilakukan observasi sehari-hari kegiatan yang dilakukan oleh tim *helpdesk* dalam menyelesaikan permasalahan dari *user*. Kegiatan observasi ini dilakukan untuk melihat bagaimana alur kerja yang diterapkan oleh tim *helpdesk* untuk menjadi sebuah dasar dan mengamati kekurangan kerja. Pada observasi yang dilakukan selama seminggu dengan melihat permasalahan yang diajukan oleh *user* dari kantor dan lapangan kerja yang beragam permasalahan. Setelah selesai melakukan observasi, mengetahui beberapa permasalahan dan kondisi yang dialami oleh tim *helpdesk*, sebagai berikut:

1. Perusahaan memiliki perkembangan otomatisasi yang cukup signifikan pada setiap departemen. Hampir semua departemen yang ada di perusahaan memiliki proyek aplikasi yang dibutuhkan untuk melakukan efisiensi dan efektif dalam pekerjaan setiap departemen.
2. Dengan adanya aplikasi setiap departemen maka akan meningkatkan beban pekerjaan yang dikerjakan oleh tim *helpdesk* yang harus menghadapi permasalahan yang sering terjadi dan juga *error* yang muncul pada aplikasi yang dimiliki oleh setiap departemen.

Hasil observasi yang didapatkan dijadikan landasan dan bahan diskusi dengan mentor serta tim *helpdesk* yang menyimpulkan bahwa penelitian *chatbot helpdesk* akan dilakukan pengembangan pertama dengan permasalahan yang masuk kategori ringan dan sering terjadi. Permasalahan yang dimaksud adalah permasalahan yang masih bisa diselesaikan secara mandiri dahulu oleh *user*. Ketika permasalahan yang dialami oleh *user* tidak bisa diatasi oleh *user*, maka pada saat

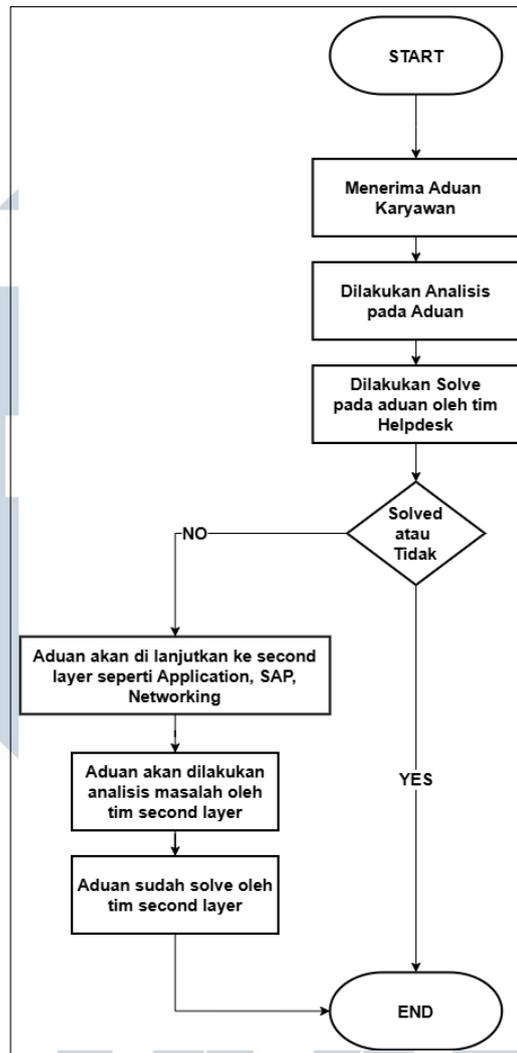
itu tim *helpdesk* akan membantu *user*. Hal ini ditentukan untuk mengurangi beban pekerjaan tim *helpdesk* sehingga bisa secara efektif dan efisien dalam membantu para *user* dengan permasalahan yang lebih besar dari kondisi yang ditentukan sebelumnya. Dengan proses otomatisasi di perusahaan maka akan bertambah beban pekerjaan tim *helpdesk*.

### 3.3 Pembuatan *Flowchart*

*Flowchart* adalah alur diagram dari sebuah sistem atau aplikasi yang dibuat untuk menjelaskan alur kerja sistem atau aplikasi yang akan dibangun. Alur yang dimiliki oleh *chatbot helpdesk* dimulai dari *input user*, Proses *validation*, proses *Natural Language Process* dan proses *Token-Based Similarity Algorithm*. Selain itu terdapat juga *flowchart* yang dibuat dari alur kerja tim *helpdesk* sebelum dilakukan otomatisasi.

#### 3.3.1 *Flowchart* Alur Kerja Helpdesk

Pada Gambar 3.1 merupakan *flowchart* alur kerja tim *helpdesk* perusahaan sebelum ada *Chatbot Helpdesk*. *User* akan melakukan komunikasi melalui *MS Teams* atau *IP Phone* dan beberapa kali *user* datang langsung ke *helpdesk* hal tersebut yang seharusnya tidak diperbolehkan. Setelah *user* menghubungi tim *helpdesk* *user* akan memberitahu permasalahan yang sedang dialami, lalu permasalahan tersebut akan dilakukan analisis oleh tim *helpdesk*. Ketika tim *helpdesk* sudah menemukan sumber masalah akan dicari solusi untuk permasalahan *user* tersebut. Jika permasalahan tersebut telah mendapatkan solusi oleh tim *helpdesk* akan dibuatkan *ticket* dan dibuat status *close*, tetapi jika tidak terselesaikan masalahnya akan diserahkan kepada tim *second layer* sesuai dengan permasalahan. Ketika tim *second layer* sudah berhasil mencari solusi dan *user* juga melakukan pengecekan kembali untuk memastikan permasalahan tersebut sudah *solve* maka status *ticket* diubah menjadi *close*.

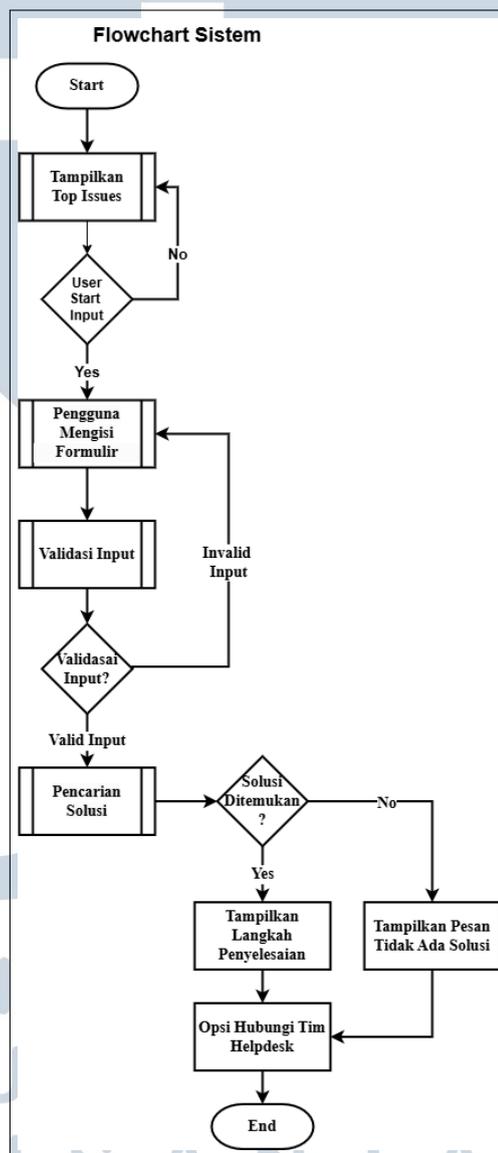


Gambar 3.1. Flow ALur Kerja Helpdesk Secara Manual

### 3.3.2 Flowchart Umum Chatbot Helpdesk

Pada Gambar 3.2 merupakan *flowchart* secara garis besar untuk sistem *Chatbot Helpdesk*. Sistem akan menampilkan lima *Top Issues* yang paling banyak dialami oleh *user*, *user* bisa memilih satu masalah yang sama sedang dialami dan akan menampilkan *pop up* yang akan dijelaskan proses di Gambar 3.3. Setelah itu, jika *user* tidak menemukan permasalahan yang sama di lima *top issues* maka bisa langsung melakukan *input* dari *category*, *subcategory* dan deskripsi permasalahan yang sedang dialami. Untuk penjelasan dari proses *input user* akan dijelaskan lebih lanjut pada Gambar 3.4. Setelah melakukan *input* permasalahan yang sedang dialami akan dilakukan proses *validation* untuk mengecek parameter *input user*

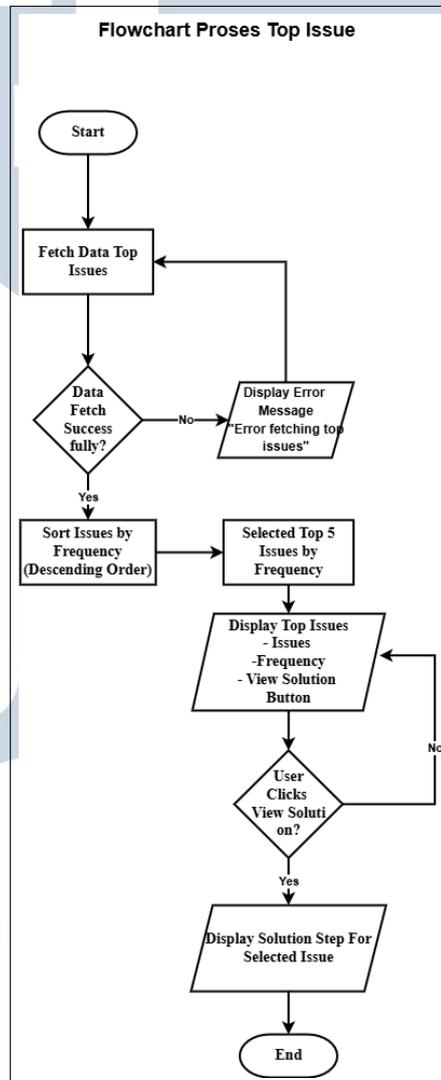
sudah terisi atau belum untuk bisa melanjutkan ke proses berikutnya. Untuk proses *validation* akan dijelaskan pada Gambar 3.9, selanjutnya sistem akan melakukan proses pencarian solusi menggunakan algoritma *Token-Based Similarity Algorithm* proses ini akan dijelaskan lebih detail pada Gambar 3.10. Jika proses dari pencarian solusi sudah selesai dan mendapatkan *similarity* tertinggi akan ditampilkan solusi *ste-by-step* ke *user*. Jika sistem tidak menemukan jawaban maka menampilkan pesan tidak ada solusi dengan tulisan "*No matching issues found.*"



Gambar 3.2. Flowchart Sistem Chatbot Helpdesk

### 3.3.3 Flowchart Top Issue

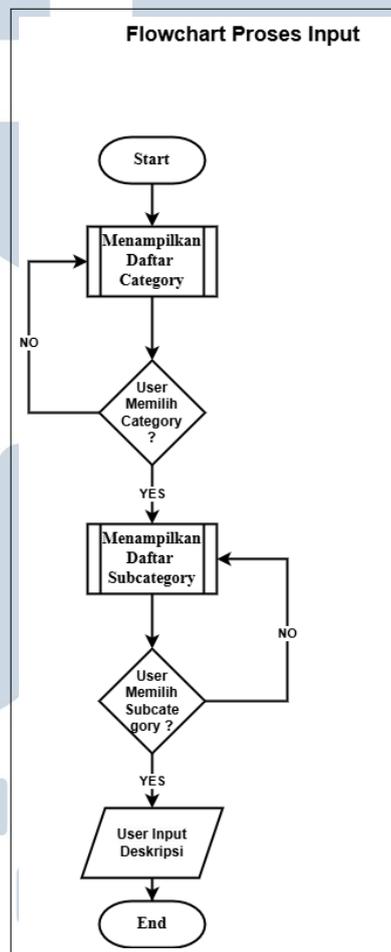
Pada Gambar 3.3 merupakan *flowchart* dari proses untuk menampilkan lima *top issues*. Sistem akan memulai membaca *database* dengan format CSV, lalu jika sistem berhasil membaca *issues* dari *database* akan mengurutkan frekuensi yang tertinggi dan jika sistem tidak berhasil membaca *database*, maka akan menampilkan *Error fetching top issues*. Setelah berhasil mengurutkan *issues* dari frekuensi tertinggi hingga terendah, maka sistem akan mengambil lima teratas dan ditampilkan ke user. Jika ada permasalahan *user* yang sama di lima *top issues* bisa klik *button View Solution* untuk menampilkan *pop up* solusi *Step-by-step*.



Gambar 3.3. Flowchart Top Issue

### 3.3.4 Flowchart Input User

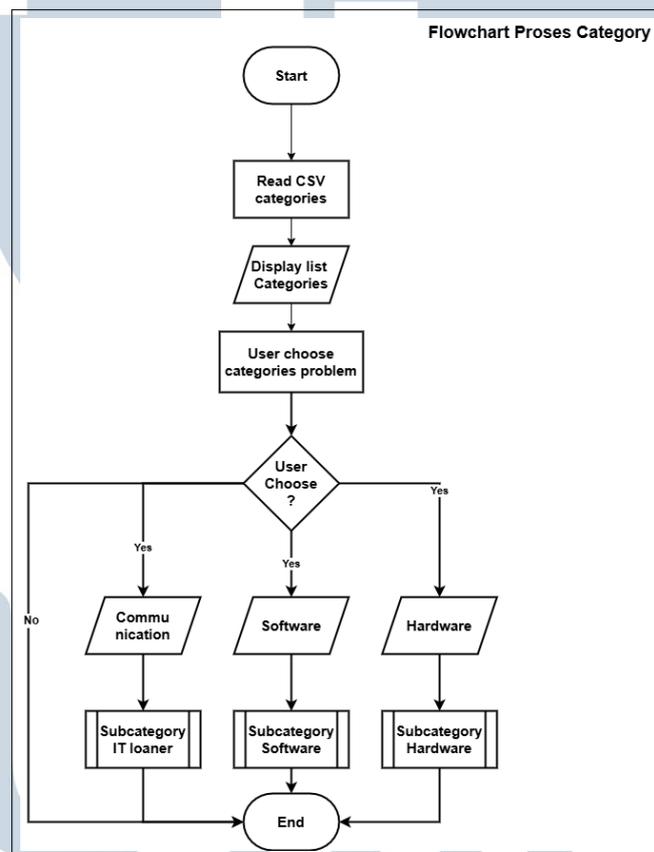
Pada Gambar 3.4 merupakan *flowchart* dari proses *user* melakukan *input* ke dalam sistem. Pertama akan memilih *category issues* yang sedang dialami, penjelasan proses sistem menampilkan *category* pada Gambar 3.5. Dalam sistem *chatbot helpdesk* mempunyai tiga *category*, yaitu *hardware*, *software* dan *communication*. Setelah *user* memilih *category*, *user* diminta memilih *subcategory* sesuai pilihan *category user*. Penjelasan untuk proses sistem menampilkan *subcategory hardware* pada Gambar 3.6, *subcategory software* pada Gambar 3.7 dan *subcategory communication* pada Gambar 3.8. Setelah memilih *category* dan *subcategory*, *user* mengisi deskripsi pertanyaan permasalahan yang sedang dialami.



Gambar 3.4. Flowchart Sistem Input

### 3.3.5 Flowchart Pemilihan Category

Pada Gambar 3.5 merupakan *flowchart* sistem menampilkan pilihan *category*. Sistem pertama membaca *file categories.csv* untuk mengambil *list category* sesuai dengan yang akan ditampilkan ke *user*. Setelah berhasil membaca dan menampilkan *list category*, *user* akan memilih salah satu *category* yang ditampilkan sistem. Setelah memilih *category*, *user* memilih *subcategory* dari *category* yang sudah dipilih.

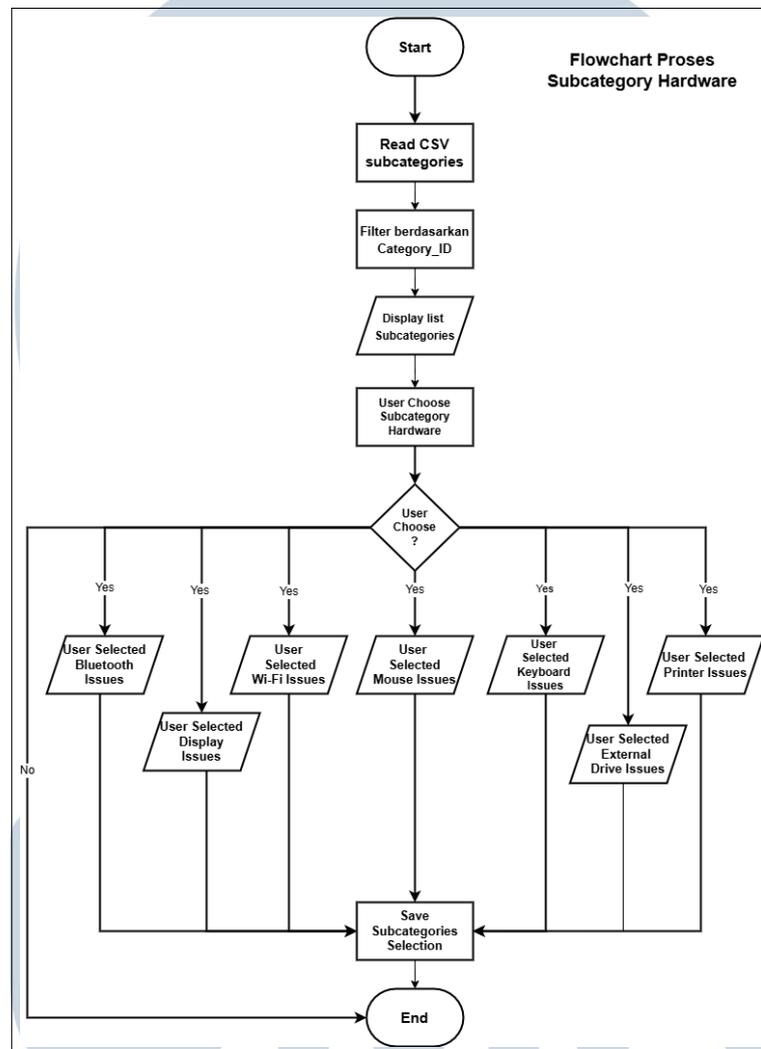


Gambar 3.5. Flowchart Pemilihan Category

### 3.3.6 Flowchart Pemilihan Subcategory

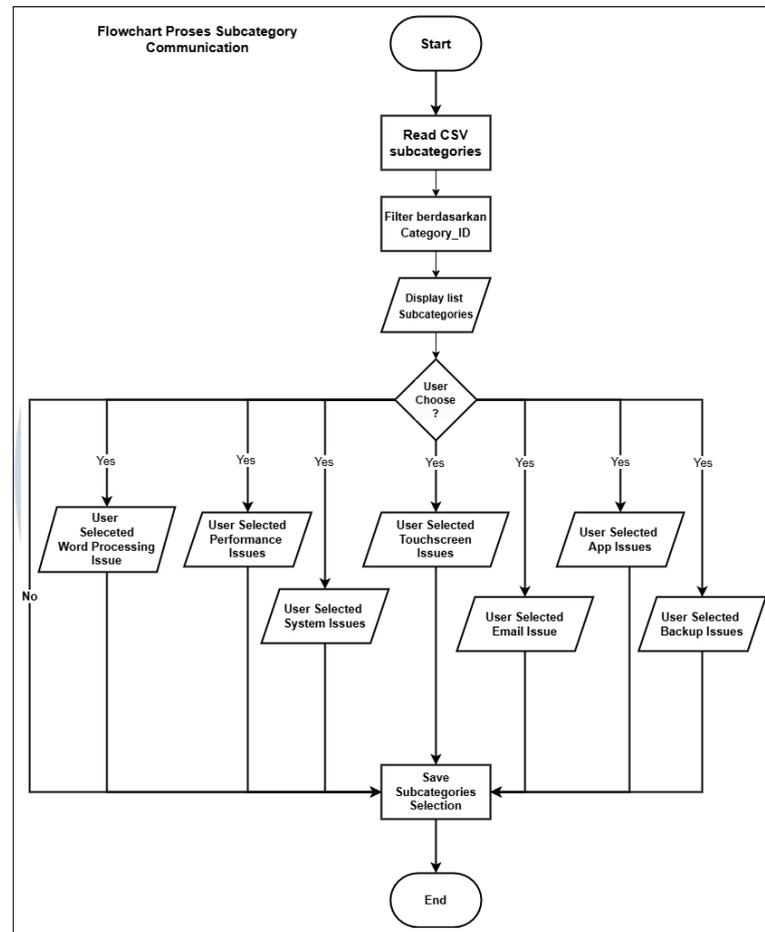
Pada Gambar 3.6 merupakan *flowchart* tentang proses sistem menampilkan pilihan *subcategory* untuk *category hardware*. Sistem akan membaca *file subcategories.csv* dan akan dilakukan *filter* sesuai dengan *category id* yang dipilih oleh *user*. Setelah mendapatkan *subcategory* sesuai dengan pilihan *category* dari *user* dan memilih *subcategory hardware* yang sesuai permasalahan. Setelah

*user* memilih salah satu *subcategory hardware* akan disimpan oleh sistem untuk digunakan untuk proses sistem melakukan *filter issues* pada proses penghitungan *similarity*.



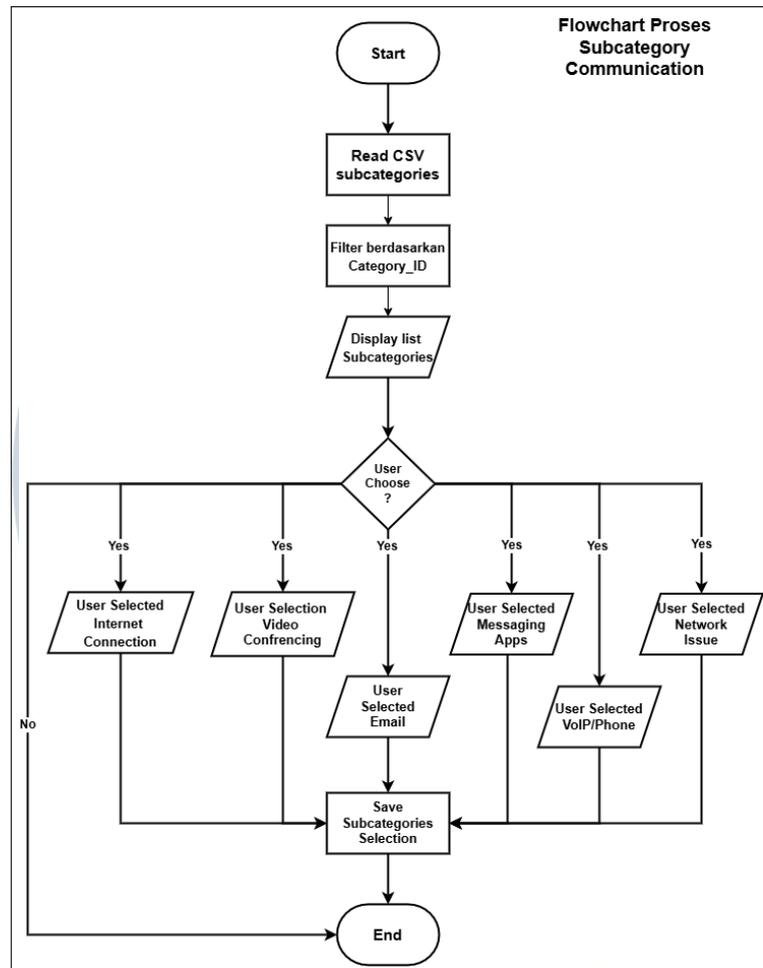
Gambar 3.6. Flowchart Pemilihan Subcategory Hardware

Pada Gambar 3.7 merupakan *flowchart* tentang proses sistem menampilkan pilihan *subcategory* untuk *category software*. Sistem akan membaca *file subcategories.csv* dan akan dilakukan *filter* sesuai dengan *category id* yang dipilih oleh *user*. Setelah mendapatkan *subcategory* sesuai dengan pilihan *category* dari *user*. *User* akan memilih *subcategory software* yang sesuai dengan permasalahan *user*. Setelah *user* memilih salah satu *subcategory software* akan disimpan oleh sistem untuk digunakan untuk proses sistem melakukan *filter issues* pada proses penghitungan *similarity*.



Gambar 3.7. Flowchart Pemilihan Subcategory Software

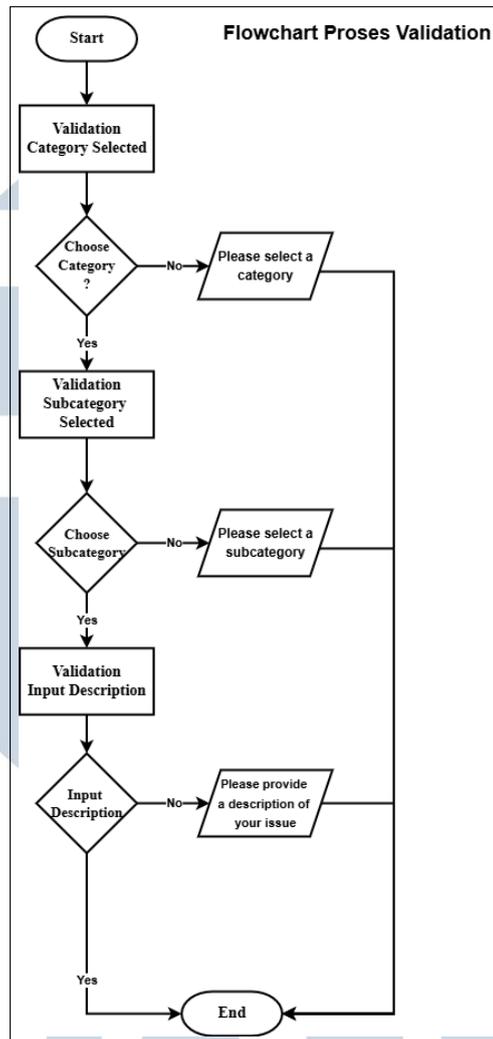
Pada Gambar 3.8 merupakan *flowchart* tentang proses sistem menampilkan pilihan *subcategory* untuk *category communication*. Sistem akan membaca *file subcategories.csv* dan akan dilakukan *filter* sesuai dengan *category id* yang dipilih oleh *user*. Setelah mendapatkan *subcategory* sesuai dengan pilihan *category* dari *user*. *User* akan memilih *subcategory communication* yang sesuai dengan permasalahan *user*. Setelah *user* memilih salah satu *subcategory communication* akan disimpan oleh sistem untuk digunakan untuk proses sistem melakukan *filter issues* pada proses penghitungan *similarity*.



Gambar 3.8. Flowchart Pemilihan Subcategory Communication

### 3.3.7 Flowchart Proses Validation

Pada Gambar 3.9 merupakan flowchart proses validation setelah user melakukan input category, subcategory dan deskripsi permasalahan. Sistem akan melakukan validasi user sudah memilih category, jika tidak memilih category akan memunculkan pesan "please select a category". Setelah itu, sistem akan melakukan validasi subcategory, jika tidak memilih subcategory akan memunculkan pesan "please select a subcategory". Setelah sistem melakukan validasi category dan subcategory sudah dipilih, maka dilanjutkan untuk melakukan validasi input deskripsi pertanyaan, jika tidak melakukan input deskripsi pertanyaan ditampilkan pesan "Please provide a description of your issue".

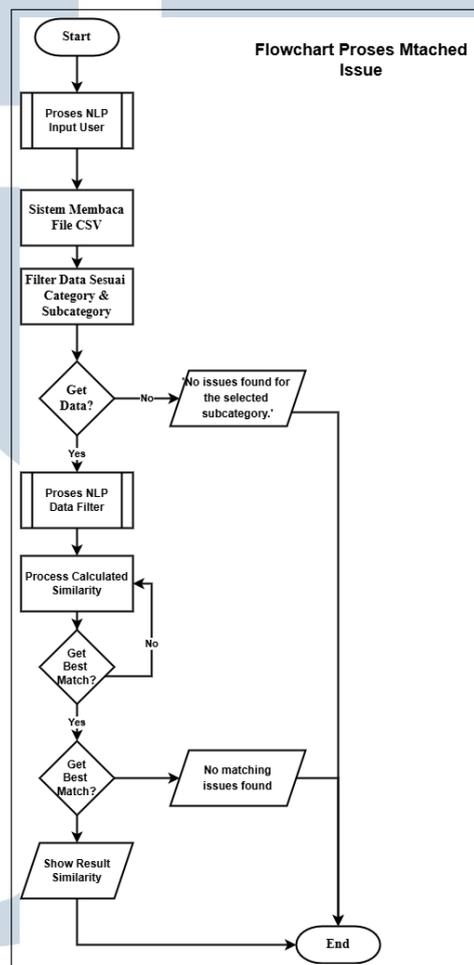


Gambar 3.9. Flowchart Proses Validation

### 3.3.8 Flowchart Proses Penghitungan Similarity

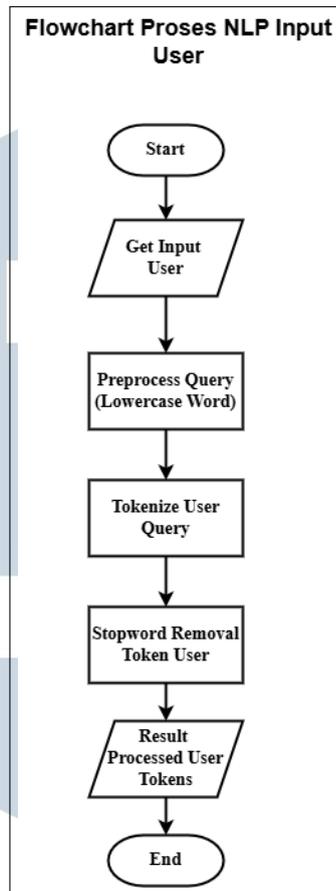
Pada Gambar 3.10 setelah melewati proses *validation*, sistem akan dilanjutkan ke proses penghitungan *similarity input user* dengan *database* yang dimiliki sistem *chatbot helpdesk*. Sistem akan dimulai dengan melakukan proses *Natural Language Processing (NLP) input user* untuk mempermudah sistem melakukan penghitungan *similarity* dan penjelasan proses ini pada Gambar 3.11. Setelah menyelesaikan proses *NLP input user*, sistem akan menyimpan hasil *NLP*. Lalu, sistem akan membaca *file issues.csv* untuk melakukan *filter* sesuai dengan *category* dan *subcategory*. Jika sistem tidak mendapatkan *issue* yang sesuai dengan *category* dan *subcategory* yang dipilih oleh *user*, maka akan menampilkan "*No issues found for the selected subcategory*". Jika sistem mendapatkan data

yang sesuai dengan *category* dan *subcategory* yang dipilih oleh *user*, sistem akan melakukan proses *NLP* untuk hasil filter dari *issues* yang didapatkan dan penjelasan proses ini pada Gambar 3.12. Setelah proses *NLP* dari *input user* dan *filter issues* akan dilanjutkan dengan penghitungan *similarity*, lalu setelah mendapatkan *Best Match* dengan *similarity* tertinggi akan dipilih untuk ditampilkan ke *user*. Jika tidak mendapatkan *Best Match issues* akan muncul pesan "No matching issues found".



Gambar 3.10. Flowchart Proses Penghitungan Similarity

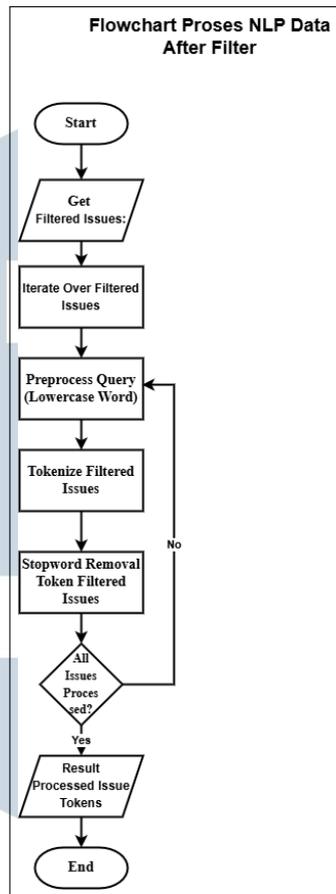
Pada Gambar 3.11 sistem menerima *input user*, setelah itu sistem akan memulai proses *Natural Language Processing (NLP)* dimulai dari *preprocess query* untuk mengubah *lowercase word* dari deskripsi pertanyaan *user*. Kedua, hasil dari *preprocess query* akan dilakukan *tokenize* untuk diubah menjadi *token* dan terakhir sistem akan dilakukan *stopword removal* untuk menghilangkan kata-kata yang tidak memiliki arti dan juga menambah tingkat hasil *similarity*.



Gambar 3.11. *Flowchart Proses NLP Input User*

Pada Gambar 3.11 sistem menerima *input user*, setelah itu sistem akan memulai proses *Natural Language Processing (NLP)* dimulai dari *preprocess query* untuk mengubah *lowercase word* dari *issues* yang dihasilkan dari *filter* sesuai dengan *category* dan *subcategory*. Kedua, hasil dari *preprocess query* akan dilakukan *tokenize* untuk diubah menjadi *token* dan terakhir sistem akan dilakukan *stopword removal* untuk menghilangkan kata-kata yang tidak memiliki arti dan juga mempermudah sistem mencocokkan dengan *input user* yang sudah melalui proses *Natural Language Processing (NLP)*

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA



Gambar 3.12. Flowchart Proses NLP After Filter

### 3.4 Perancangan Chatbot Helpdesk

Perancangan *chatbot helpdesk* menggunakan *framework next.js* dan struktur kode yang dibuat merupakan *sequential*, karena kode dieksekusi secara berurutan dari atas ke bawah. Karena menggunakan *framework Next.js* pembuatan *frontend* dan *backend* bisa dilakukan menggunakan bahasa *JavaScript*. Struktur kode saya memiliki fungsi yang berkaitan dengan API dan tidak berkaitan dengan objek-objek tertentu, melainkan sebagai API yang dapat dipanggil secara independen yang diartikan fungsi-fungsi ini tidak memerlukan *output* dari fungsi lain untuk dapat berfungsi.

Pada Lampiran 4 merupakan potongan kode dari *frontend* menggunakan *framework Next.js* untuk tampilan *chatbot helpdesk*.

Pada Kode 3.1 merupakan potongan kode salah satu API dari *backend* menggunakan *framework Next.js* untuk fungsi yang digunakan di *chatbot helpdesk*.

```
1 import path from 'path'; // Modul untuk bekerja dengan path file
```

```

2 import { readCSV } from '../lib/csv'; // Import fungsi baca CSV
3
4 /**
5  * API handler untuk mengambil data kategori
6  * @param {Object} req - Request object
7  * @param {Object} res - Response object
8  */
9 export default async function handler(req, res) {
10   const filePath = path.join(process.cwd(), 'data', 'categories.
11     csv'); // Path file kategori
12
13   try {
14     const categories = await readCSV(filePath); // Baca data
15     kategori
16     console.log('Categories fetched:', categories);
17     res.status(200).json(categories); // Kembalikan data kategori
18     dalam JSON
19   } catch (error) {
20     console.error('Error fetching categories:', error);
21     res.status(500).json({ error: 'Error fetching categories' });
22     // Tangani error
23   }
24 }

```

Kode 3.1: Potongan kode *backend* menggunakan *framework Next.js*

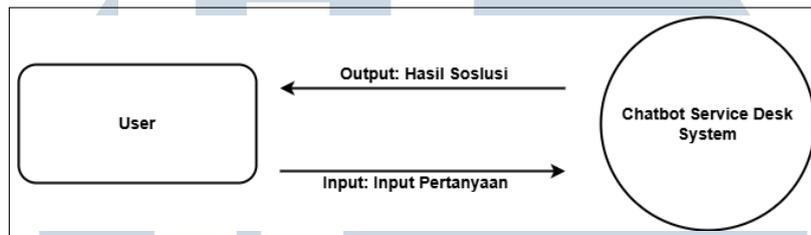
### 3.5 Pembuatan *Database*

Pembuatan *database* yang digunakan sistem *chatbot helpdesk* menggunakan format CSV. Dimulai dari membuat *Data Flow Diagram (DFD)*, membuat *Entity Relationship Diagram (ERD)* dan membuat *database* menggunakan format CSV.

#### 3.5.1 Pembuatan *Data Flow Diagram (DFD)*

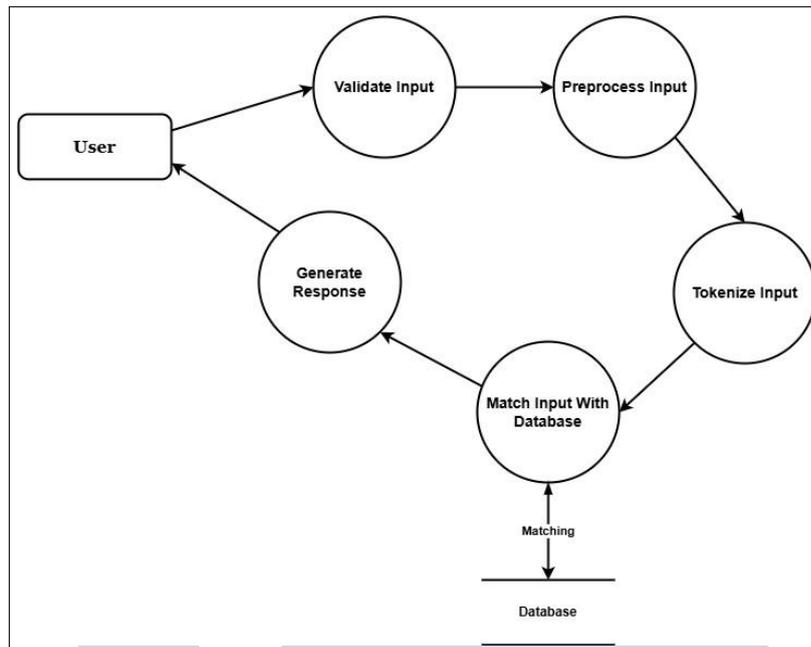
Pada Gambar 3.13 merupakan *Data Flow Diagram (DFD) Level 0* menggambarkan sistem secara umum dengan menunjukkan aliran data antara entitas eksternal, yaitu pengguna (*user*), dan sistem utama, yaitu sistem *Chatbot Service Desk*. Pada level ini, aliran data yang terjadi hanya melibatkan dua komponen utama, yaitu input dari pengguna dan proses yang dijalankan oleh sistem. Pengguna memberikan input berupa pertanyaan atau masalah yang ingin diselesaikan, dan sistem *Chatbot Service Desk* memproses input tersebut untuk

menghasilkan *output* berupa solusi yang relevan atau pesan kesalahan apabila solusi tidak ditemukan. Sebagai contoh, ketika pengguna mengajukan pertanyaan seperti, "Laptop saya tidak bisa tersambung ke *Wi-Fi*," sistem akan merespons dengan memberikan solusi yang sesuai, seperti "Coba *restart router* Anda atau periksa *driver Wi-Fi* Anda."



Gambar 3.13. *Data Flow Diagram (DFD) Level 0*

pada Gambar 3.14 merupakan *Data Flow Diagram (DFD) Level 1* menjelaskan proses-proses yang terjadi dalam sistem *Chatbot Service Desk* dengan menggambarkan modul-modul fungsional yang lebih spesifik. Proses pertama dalam sistem adalah *Validasi Input*, di mana sistem memeriksa keabsahan input yang diberikan oleh pengguna, seperti memastikan bahwa *input* tidak kosong dan panjangnya memenuhi batas minimum yang ditentukan. Apabila input tidak valid, sistem akan meminta pengguna untuk memberikan *input* ulang. Selanjutnya, pada tahap *Preprocessing Input*, sistem menghapus karakter atau simbol yang tidak diperlukan serta mengubah huruf kapital menjadi huruf kecil untuk menjaga konsistensi data. Proses berikutnya adalah *Tokenisasi Input*, yaitu memecah *input* menjadi token-token (kata-kata) yang dapat diproses lebih lanjut. Setelah itu, dalam tahap pencocokan *Input* dengan *database*, token yang telah diproses akan dibandingkan dengan data yang ada dalam *database*, yang berisi daftar masalah dan solusi yang relevan, disusun berdasarkan kategori dan subkategori. Jika ditemukan kecocokan, sistem akan memberikan solusi yang sesuai pada tahap *Generasi Respon*. Jika tidak ada kecocokan, sistem akan memberikan pesan kesalahan atau meminta klarifikasi lebih lanjut. *database* yang digunakan berfungsi sebagai tempat penyimpanan informasi terkait masalah dan solusi yang digunakan dalam pencocokan dengan *input* pengguna. Aliran data dalam sistem dimulai dari *input* yang diberikan oleh pengguna, kemudian diproses dalam berbagai tahap hingga menghasilkan *output* berupa solusi atau pesan kesalahan yang dikirimkan kembali ke pengguna.



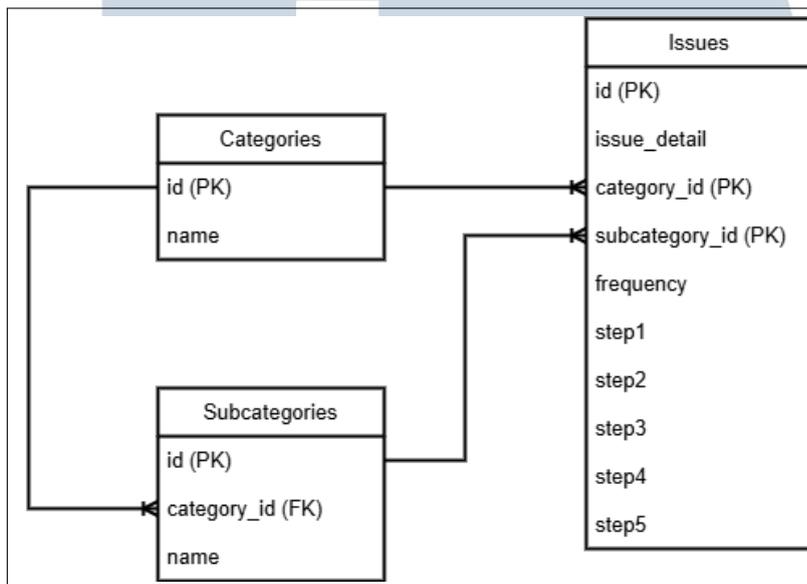
Gambar 3.14. Data Flow Diagram (DFD) Level 1

### 3.5.2 Pembuatan *Entity Relationship Diagram (ERD)*

Pada Gambar 3.15 merupakan *Entity-Relationship Diagram (ERD)* ini menggambarkan struktur *database* yang terdiri dari tiga entitas utama, yaitu *Categories*, *Subcategories*, dan *Issues*, serta hubungan antar entitas tersebut. Entitas *Categories* merepresentasikan kategori masalah yang dapat dihadapi oleh pengguna, seperti "Masalah Jaringan" atau "Masalah Perangkat Keras", dengan setiap kategori memiliki *id* unik dan nama kategori. *Subcategories* mengelompokkan lebih lanjut masalah dalam kategori tertentu, seperti "Koneksi Wi-Fi Lemah" atau "Koneksi VPN Tidak Berfungsi". Setiap subkategori terhubung dengan kategori melalui kolom *category\_id*, yang merupakan kunci asing yang merujuk ke ID kategori pada entitas *Categories*. Sementara itu, *Issues* menyimpan masalah atau isu spesifik yang dialami pengguna, disertai dengan langkah-langkah penyelesaian yang dapat diikuti. *Issues* terhubung dengan *Categories* dan *Subcategories* melalui kolom *category\_id* dan *subcategory\_id*, yang merujuk ke *id* kategori dan subkategori terkait.

Hubungan antar entitas ini menunjukkan hubungan *one-to-many* antara *Categories* dan *Subcategories*, serta antara *Subcategories* dan *Issues*, di mana setiap kategori dapat memiliki banyak subkategori, dan setiap subkategori dapat memiliki banyak isu. *Issues* juga terhubung dengan *Categories* melalui kolom

*category.id*, menciptakan hubungan *one-to-many* yang memastikan setiap isu hanya berhubungan dengan satu kategori dan subkategori. Struktur ini memudahkan pengelolaan data yang terkait dengan kategori, subkategori, dan isu dalam sistem informasi, memastikan integritas data terjaga, dan mempermudah pencarian solusi berdasarkan kategori dan subkategori masalah. Dengan hubungan yang jelas antara tabel-tabel ini, sistem dapat berjalan efisien dan memudahkan pengelolaan *database* dalam jangka panjang.



Gambar 3.15. *Entity Relationship Diagram (ERD)*

Pada Gambar 3.16 merupakan tampilan dari *Database categories.csv* yang berisi data tentang *categories* yang dipilih oleh *user* dan menjadi *filter* yang akan digunakan *chatbot helpdesk* untuk mempermudah pencarian solusi untuk *user*.

```

    natbot-service-desk > data > categories.csv > data
    1 id,name
    2 1,Hardware
    3 2,Software
    4 3,Communication
    
```

Gambar 3.16. *Database Categories*

Pada Gambar 3.17 merupakan tampilan dari *Database subcategories.csv* yang memiliki 100 *list subcategories* berisi data tentang *subcategories* yang bisa dipilih oleh *user* untuk lebih spesifik permasalahan yang dialami. *Database* ini

juga membantu *chatbot helpdesk* lebih spesifik melakukan *filter* permasalahan *user* dan menemukan jawaban yang tepat.

```
chatbot-service-desk > data > subcategories.csv > data

1  id,category_id,name
2  1,1,Keyboard Issues
3  2,1,Mouse Issues
4  3,2,Word Processing Issues
5  4,1,Printer Issues
6  5,1,Wi-Fi Issues
7  6,1,Bluetooth Issues
8  7,2,Performance Issues
9  8,2,Email Issues
10 9,1,Display Issues
11 10,1,External Drive Issues
12 11,2,Touchscreen Issues
13 12,3,Internet Connection
14 13,3,Email Communication
15 14,3,Video Conferencing Tools
16 15,3,Messaging Apps
17 16,3,VoIP/Phone Issues
18 17,2,App Issues
19 18,2,System Issues
20 19,2,Backup Issues
21 20,3,Network Issues
```

Gambar 3.17. Database Subcatgeories

Pada Gambar 3.18 merupakan tampilan dari Database *issues.csv* memiliki 100 *list issues* yang berisi tentang informasi solusi yang dimiliki oleh *chatbot helpdesk* untuk melakukan *matched issue* setelah *filter* menggunakan *categories* dan *subcategories*. Database ini merupakan *database* yang penting dikarenakan menjadi otak untuk memberikan jawaban yang tepat sesuai dengan permasalahan *user*.

id	issue_detail	category_id	subcategory_id	frequency	step1	step2	step3	step4	step5
1	Wi-Fi not connecting	1	1	20	Restart the router.	Reconnect to the network.	Update wireless drivers.	Check router settings.	Contact support.
2	Wi-Fi speed slow	1	1	18	Restart the router.	Clear cache.	Use different channel.	Update router firmware.	Contact support.
3	Bluetooth not pairing	2	1	15	Ensure Bluetooth is enabled.	Restart devices.	Update drivers.	Forget and re-pair device.	Contact support.
4	Bluetooth disconnects frequently	2	2	12	Update drivers.	Reconnect device.	Check battery levels.	Disable power saving.	Contact support.
5	Email not syncing	3	1	14	Check server settings.	Restart app.	Update email client.	Check internet.	Contact support.
6	Email sending failed	3	2	10	Check SMTP settings.	Verify recipient email.	Restart client.	Update app.	Contact support.
7	Slow browsing speed	4	1	13	Restart router.	Clear browser cache.	Use a wired connection.	Check for malware.	Contact support.
8	Network cable unplugged	4	2	9	Check cable connection.	Replace cable.	Check adapter settings.	Restart router.	Contact support.
9	App notifications delayed	5	1	8	Check app settings.	Update app.	Restart phone.	Clear app cache.	Contact support.
10	App not opening	5	2	6	Restart device.	Reinstall app.	Check permissions.	Update app.	Contact support.
11	Wi-Fi disconnects frequently	1	1	16	Restart router.	Change Wi-Fi channel.	Move closer.	Update firmware.	Contact support.
12	Bluetooth audio lagging	2	1	11	Check battery levels.	Restart devices.	Reduce distance.	Disable unused devices.	Contact support.
13	Email marked as spam	3	3	7	Check spam filters.	Whitelist sender.	Contact provider.	Update client.	Contact support.
14	Slow upload speed	4	1	10	Restart router.	Close background uploads.	Update		

Gambar 3.18. Database Issues

UMN  
UNIVERSITAS  
MULTIMEDIA  
NUSANTARA