



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk mengubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB III

METODOLOGI DAN PERANCANGAN SISTEM

3.1 Metodologi Penelitian

Metode penelitian adalah cara yang dilakukan dalam penelitian untuk mendapatkan sebuah hasil yang diharapkan. Dalam penelitian ini metode yang digunakan adalah pendekatan yang sistematis dan juga berurutan pada pengembangan perangkat lunak, dimulai dengan spesifikasi kebutuhan pengguna lalu berlanjut melalui tahapan-tahapan perencanaan (*planning*), permodelan (*modeling*), konstruksi (*construction*), dan diakhiri dengan melakukan evaluasi terhadap hasil konstruksi. Adapun tahapan dari metode penelitian yang dilaksanakan adalah sebagai berikut:

1. Studi Literatur

Tahap pertama dari penelitian ini adalah dengan melakukan studi terhadap kebutuhan yang akan digunakan dalam pembuatan aplikasi, dimulai dengan studi terhadap referensi mengenai algoritma Huffman dalam melakukan kompresi *file*. Referensi berupa jurnal, buku dan situs-situs terpercaya yang dapat dipertanggungjawabkan isinya.

2. Perencanaan dan Perancangan Sistem

Melakukan perencanaan dan analisis terlebih dahulu sebelum mengerjakan. Misalnya tabel yang dibutuhkan, flowchart, pengumpulan bahan dan sebagainya serta merancang hasil analisa yang akan dibuat dan diimplementasikan kedalam aplikasi. Tahap perancangan dibagi menjadi 2 tahap, yang pertama adalah mengimplementasikan algoritma Huffman untuk melakukan kompresi pada *file*

teks yang akan dikirim. Tahap kedua adalah mengimplementasikan proses transmisi data menggunakan Bluetooth pada perangkat Android. Aplikasi ini dirancang menggunakan bahasa pemrograman Kotlin yang telah terintegrasi dengan Java Class Library (JCL).

3. Pembuatan Aplikasi

Tahap penterjemah data yang telah dirancang kedalam bahasa pemrograman tertentu. Dalam pembangunan aplikasi ini berbasis android dan menggunakan bahasa pemrograman Kotlin dengan Java Class Library (JVC).

4. Pengujian

Pada tahap ini merupakan tahapan pengujian dari hasil implementasi yang telah dibuat. Tahap ini sangat penting untuk menjaga kualitas aplikasi yang dibuat. Pengujian fokus pada perangkat lunak secara dari segi logika dan fungsional dan memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan/*error* dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan. Data yang digunakan pada pengujian adalah data spesifik yang bertipe *file* teks.

5. Evaluasi

Proses evaluasi dilakukan berdasarkan hasil dari pengujian dengan tujuan menarik kesimpulan atas hasil penelitian dan dapat menjadi referensi untuk pengembangan selanjutnya.

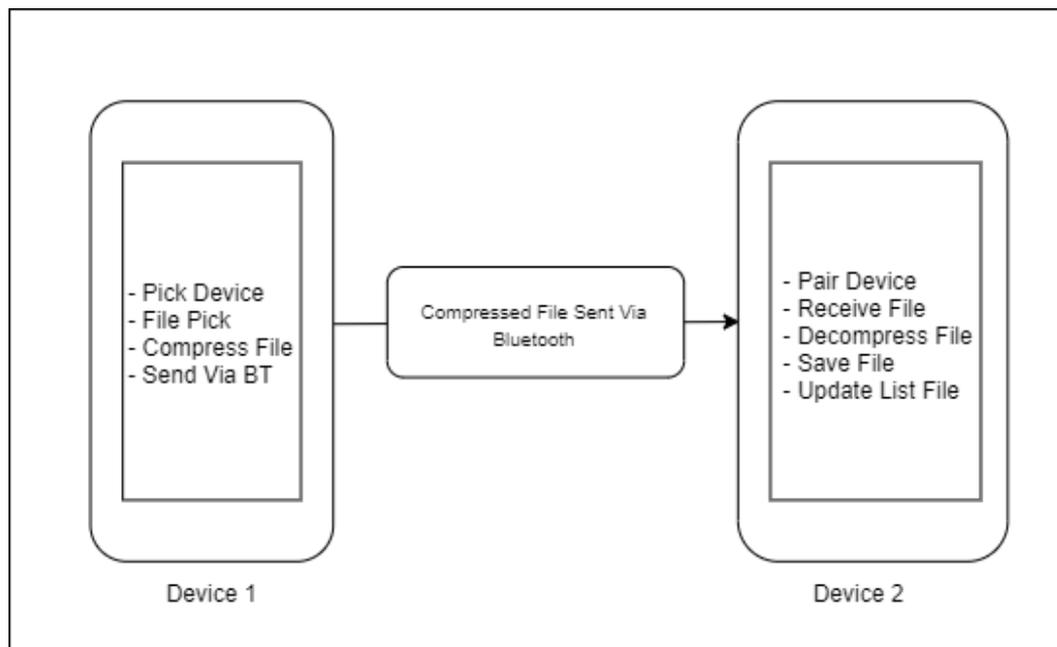
6. Penulisan Laporan

Penulisan laporan dilakukan untuk memuat hasil dokumentasi dan penjelasan mengenai program atas metode, proses dan kode yang digunakan dan dilakukan dalam penelitian ini.

3.2 Perancangan Aplikasi Textra

Perancangan sistem pada penelitian ini menggunakan diagram-diagram perancangan yaitu model aplikasi, *flowchart* dan desain tampilan antar muka.

3.2.1 Fungsionalitas Aplikasi

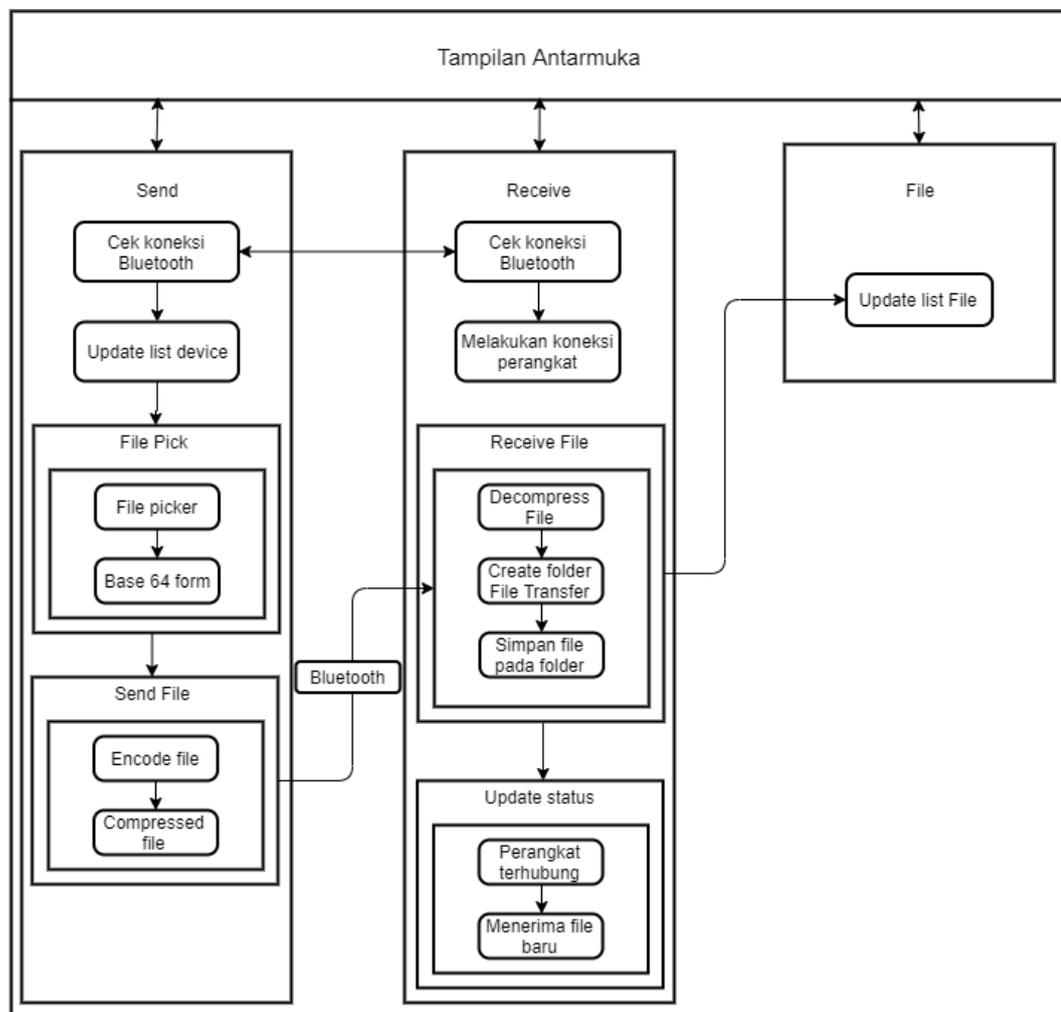


Gambar 3.1 Model Fungsionalitas Aplikasi Textra

Gambar 3.1 menunjukkan bahwa aplikasi Textra membutuhkan dua perangkat agar dapat digunakan. Gambar 3.1 juga menunjukkan tahapan-tahapan yang terjadi ketika kedua perangkat akan melakukan proses transmisi data. Dimulai dari perangkat pertama, pengguna akan memulai dengan memilih dan meminta persetujuan perangkat yang menjadi tujuan pengiriman data, lalu pengguna dapat memilih *file* yang akan dikompres dan dikirim. Ketika pengguna menekan tombol *send*, aplikasi akan secara otomatis melakukan kompresi pada *file* dan *file* yang sudah dikompres akan dikirim menggunakan layanan Bluetooth.

Pada perangkat kedua, pengguna dapat memberikan izin kepada perangkat yang meminta untuk dihubungkan dengan menekan tombol *listen device*. Ketika perangkat pertama sudah mengirimkan *file* melalui layanan Bluetooth, aplikasi pada perangkat kedua akan melakukan menyimpan *file* di folder yang sudah ditentukan dan *file* akan di dekompres untuk mengembalikan bentuk aslinya secara otomatis. Pengguna dapat melakukan pengecekan data yang sudah pernah diterima.

3.2.2 Model Aplikasi



Gambar 3.2 Model Aplikasi Textra

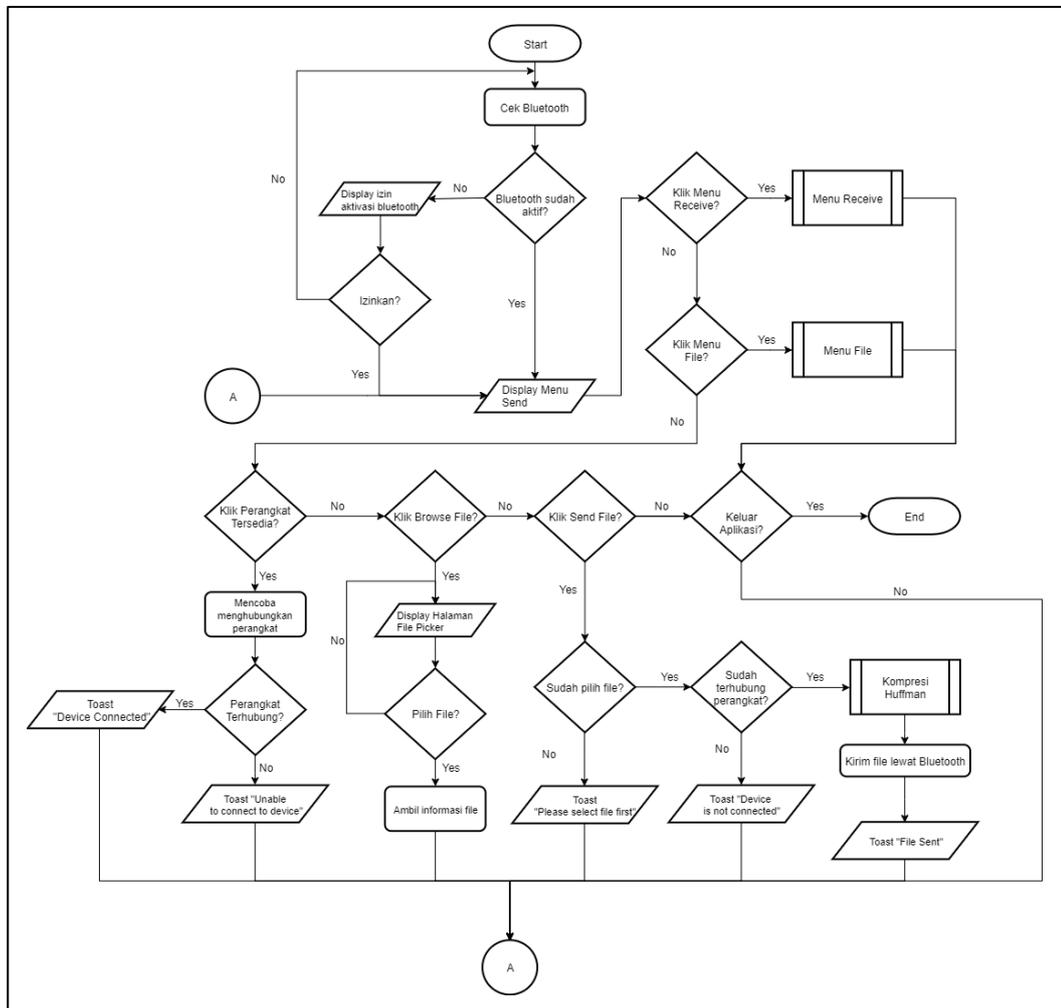
Gambar 3.2 menunjukkan tiga menu utama pada Aplikasi Textra. Dari ketiga menu ini terdiri atas tahap-tahap yang berbeda. Proses pengiriman dan penerimaan *file* ditampilkan pada menu *Send* dan *Receive*, sedangkan menu *File*, berisi daftar dari *file* yang sudah pernah diproses dan diurutkan berdasarkan waktu pemrosesannya. Pada menu *Send* terdiri secara berurutan atas proses pengecekan status koneksi perangkat, *update* daftar perangkat yang tersedia untuk dihubungkan melalui Bluetooth, pemilihan *file* dan pengiriman *file*.

Menu kedua adalah menu *Receive*, proses yang terjadi secara berurutan dalam menu ini adalah pengecekan status koneksi perangkat, melakukan koneksi dengan perangkat yang memancarkan sinyal Bluetooth, proses penerimaan *file*, dan proses *update* status. Proses penerimaan *file* terdiri dari tiga tahap, yang pertama adalah melakukan proses dekompres pada *file* agar menjadi ke bentuk semula, lalu tahap kedua adalah melakukan pengecekan terhadap ketersediaan perangkat atas folder penyimpanan yang sudah ditetapkan, apabila perangkat tidak memiliki folder tersebut maka program akan membuat folder tersebut secara otomatis. Tahap ketiga adalah penyimpanan *file* yang telah di dekompres pada folder penyimpanan. Proses selanjutnya adalah proses *update* status, proses ini dilakukan ketika perangkat terhubung dan siap untuk melakukan pertukaran *file* atau perangkat telah menerima *file*.

Menu ketiga adalah menu *File*, menu ini menampung daftar nama *file* yang telah selesai melakukan proses pertukaran *file*. Pada menu ini, program akan secara otomatis untuk melakukan *update* list ketika *file* telah diterima.

3.2.3 Menu Send

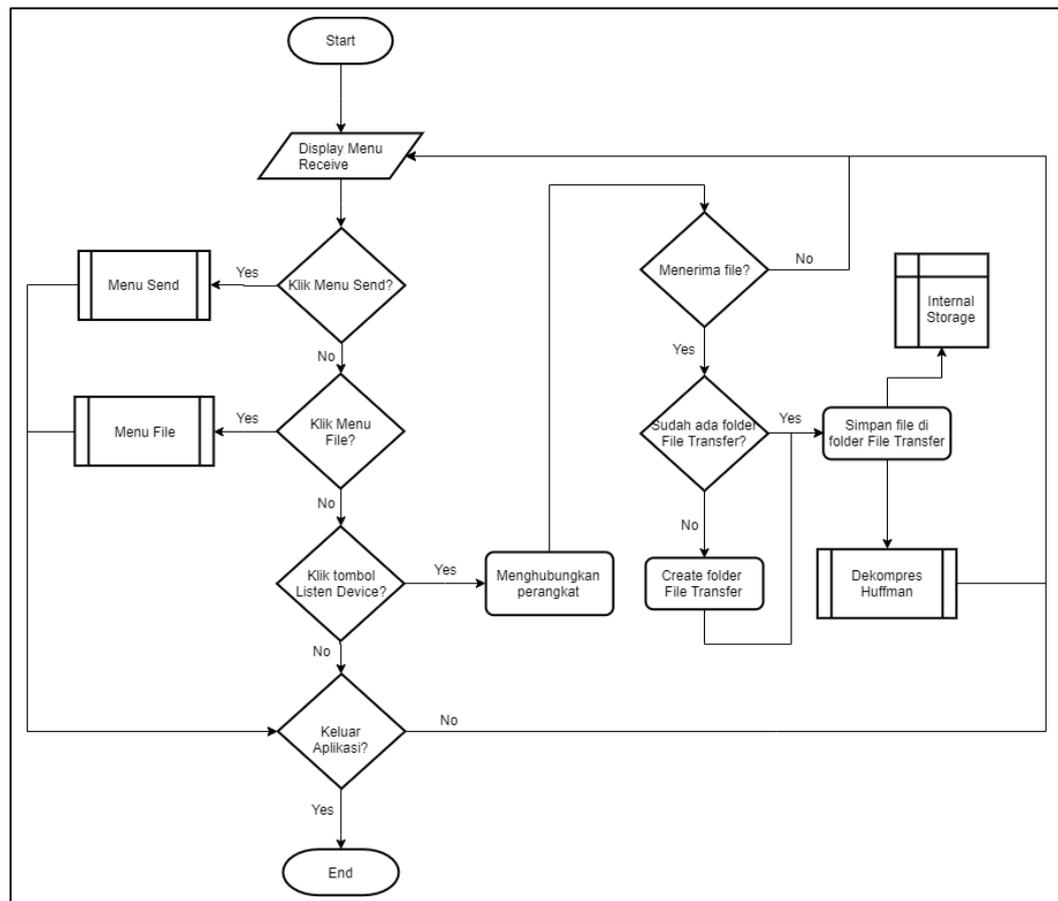
Gambar 3.3 merupakan *flowchart* menu utama pada aplikasi Textra, yaitu menu *Send*. Pada saat pertama kali aplikasi dibuka, pengguna akan menghadapi menu utama yaitu menu *Send*. Pada menu ini terdapat tiga buah tombol navigasi, jika pengguna menekan tombol *Receive* maka halaman akan berpindah ke halaman menu *Receive*, apabila tombol *File* ditekan maka halaman akan berpindah ke halaman menu *File*.



Gambar 3.3 Flowchart Menu *Send*

Pada menu *Send*, pengguna dapat memilih perangkat yang akan dihubungkan untuk transmisi *file* dan memilih *file* yang akan dikirim. Tombol *send file* tidak dapat berfungsi jika *file* belum dipilih atau perangkat tidak terhubung dengan perangkat lain. Ketika pengguna sudah memilih *file* dan terhubung dengan perangkat lain untuk menjalankan proses transmisi, *file* akan diubah ke bentuk terkompresi ketika tombol *send file* ditekan, lalu *file* yang telah dikompres dikirimkan melalui Bluetooth.

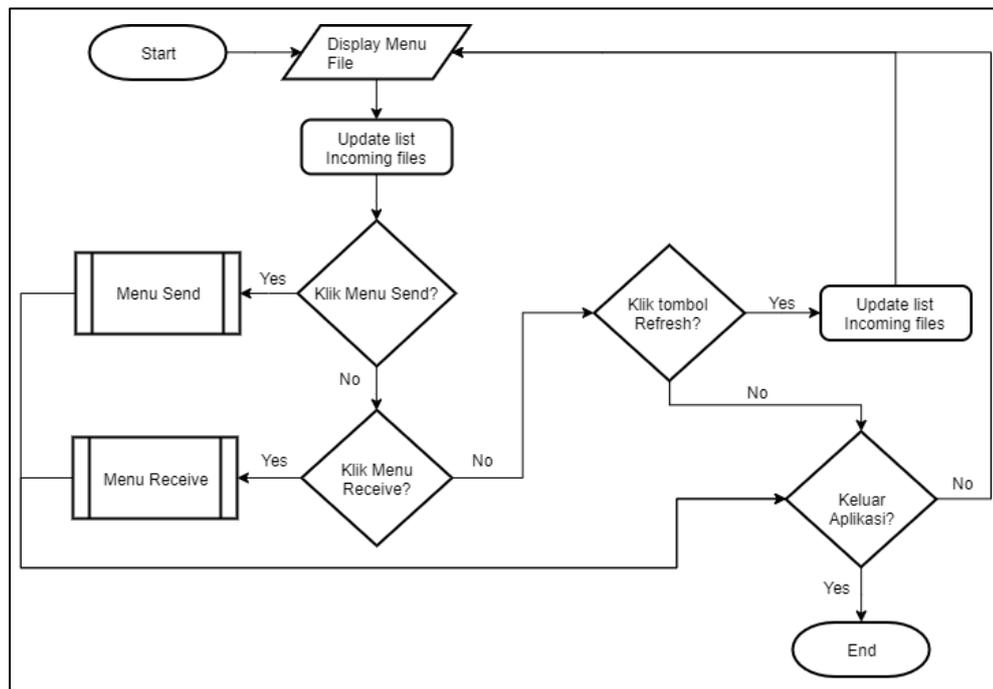
3.2.4 Menu Receive



Gambar 3.4 Flowchart Menu *Send*

Gambar 3.4 menunjukkan bahwa menu ini dibuat khusus untuk menangani proses koneksi dengan perangkat lain dan sebagai penunjuk status dari proses transmisi *file*. Pada menu ini terdapat tombol *Listen Device*. Tombol ini berfungsi sebagai penangkap sinyal Bluetooth yang disebarkan oleh perangkat lain dan konfirmasi untuk melakukan koneksi. Apabila perangkat sedang tidak terhubung dengan perangkat lain, maka menu ini akan mengeluarkan pesan "- -", apabila perangkat sedang terhubung dengan perangkat lain maka menu ini akan mengeluarkan pesan "*Device connected!*", apabila perangkat selesai menerima sebuah *file* maka menu ini akan mengeluarkan pesan "*New file received!*". Gambar 3.5 menunjukkan proses ketika menerima sebuah *file*.

3.2.5 Menu File



Gambar 3.5 Flowchart Menu File

Ketika *file* sudah selesai ditransmisikan, maka nama *file* akan ditambahkan kedalam daftar nama *file* pada menu *File*. Seperti proses pada Gambar 3.5.

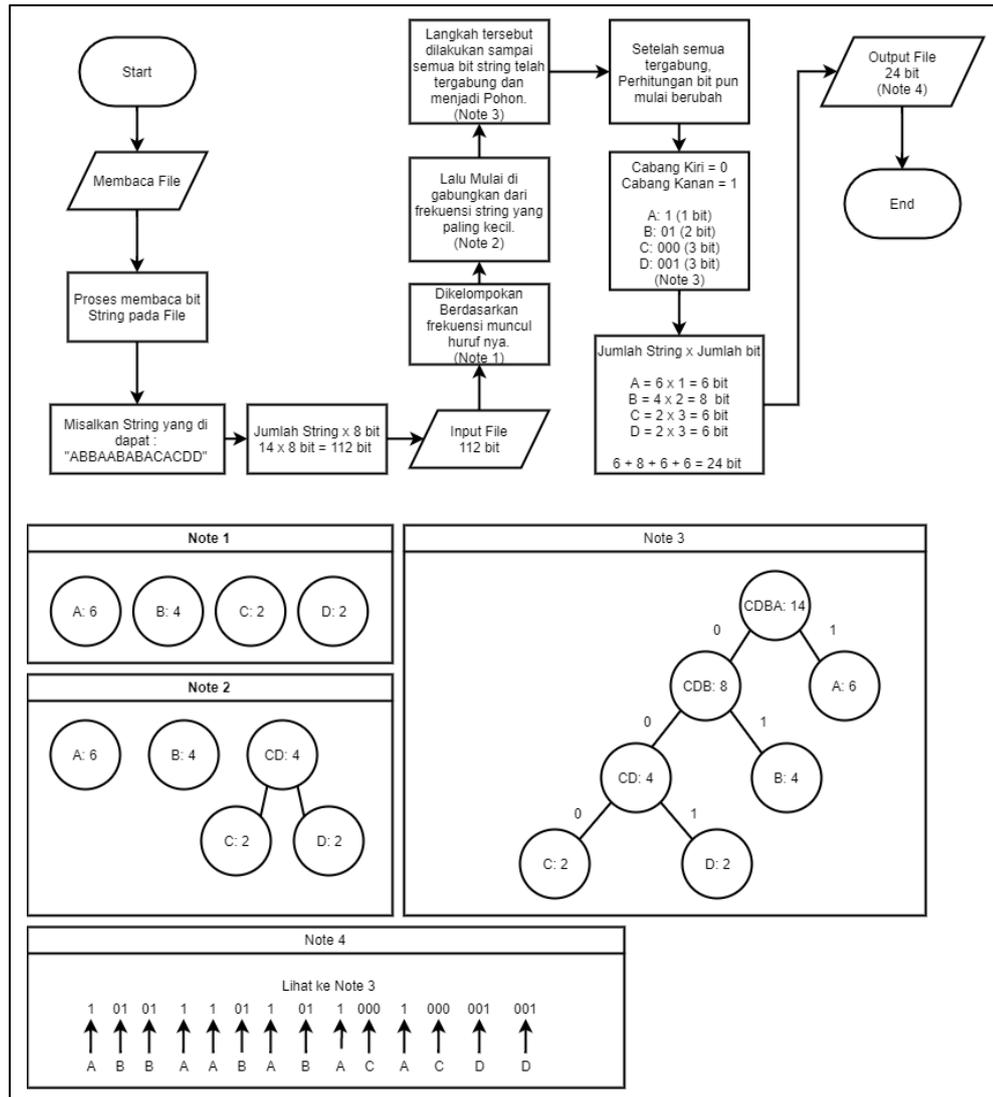
3.3 Reference Huffman Coding

Penelitian ini memanfaatkan sebuah proyek *open source*, yang sumber kodenya tersedia secara bebas dan dapat didistribusikan serta dimodifikasi. Proyek yang dimaksud adalah proyek berjudul *Reference Huffman Coding*, yang mana proyek ini memberikan penjelasan, contoh kasus dan sumber kode mengenai kompresi dan dekompres Huffman secara terpisah dalam bahasa pemrograman Java, Python, dan C ++, dan merupakan *open source*.

Metode kompresi dan dekompres yang tersedia terdiri atas proses *encode* dan *decode*, pembentukan pohon Huffman, dan pembentukan tabel frekuensi untuk memberikan representasi baru pada *input*. Berdasarkan proyek tersebut, penelitian ini mengambil metode kompresi dan dekompresi Huffman yang tersedia dalam bahasa pemrograman Java dan dimodifikasi ke dalam bentuk bahasa pemrograman Kotlin pada Android yang terintegrasi dengan *Java Class Library*-nya. Lalu aplikasi dilengkapi dengan penambahan layanan Bluetooth untuk proses transmisi data sebelum dan setelah *file* dikompres menggunakan algoritma Huffman.

3.4 Kompresi Huffman

Gambar 3.6 adalah ilustrasi proses pembuatan pohon biner dan kompresi yang terjadi ketika melakukan sebuah proses transmisi *file* yang memanfaatkan algoritma Huffman.



Gambar 3.6 Flowchart Menu *Send*

Dalam proses kompresi algoritma Huffman, prosesnya terdiri dari tiga tahapan utama yaitu proses pembuatan pohon biner, *encoding* dan *decoding*. Urutan tahapan ini dapat dilihat seperti pada Gambar 2.2.

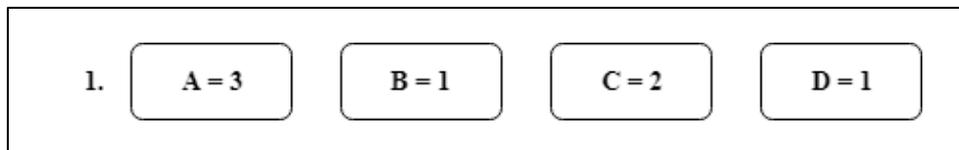
Tahapan pertama dari algoritma Huffman dimulai dengan membuat sebuah pohon biner. Sebagai contoh, dalam kode ASCII *string* 7 huruf “ABACCDA” membutuhkan representasi 7 x 8 bit = 56 bit (7 byte), dengan rincian seperti pada Tabel 3.1.

Tabel 3.1 Tabel konversi ASCII ke biner pada string “ABACCCA”

Contoh	Nilai Bit
A	01000001
B	01000010
A	01000001
C	01000011
C	01000011
D	01000100
A	01000001

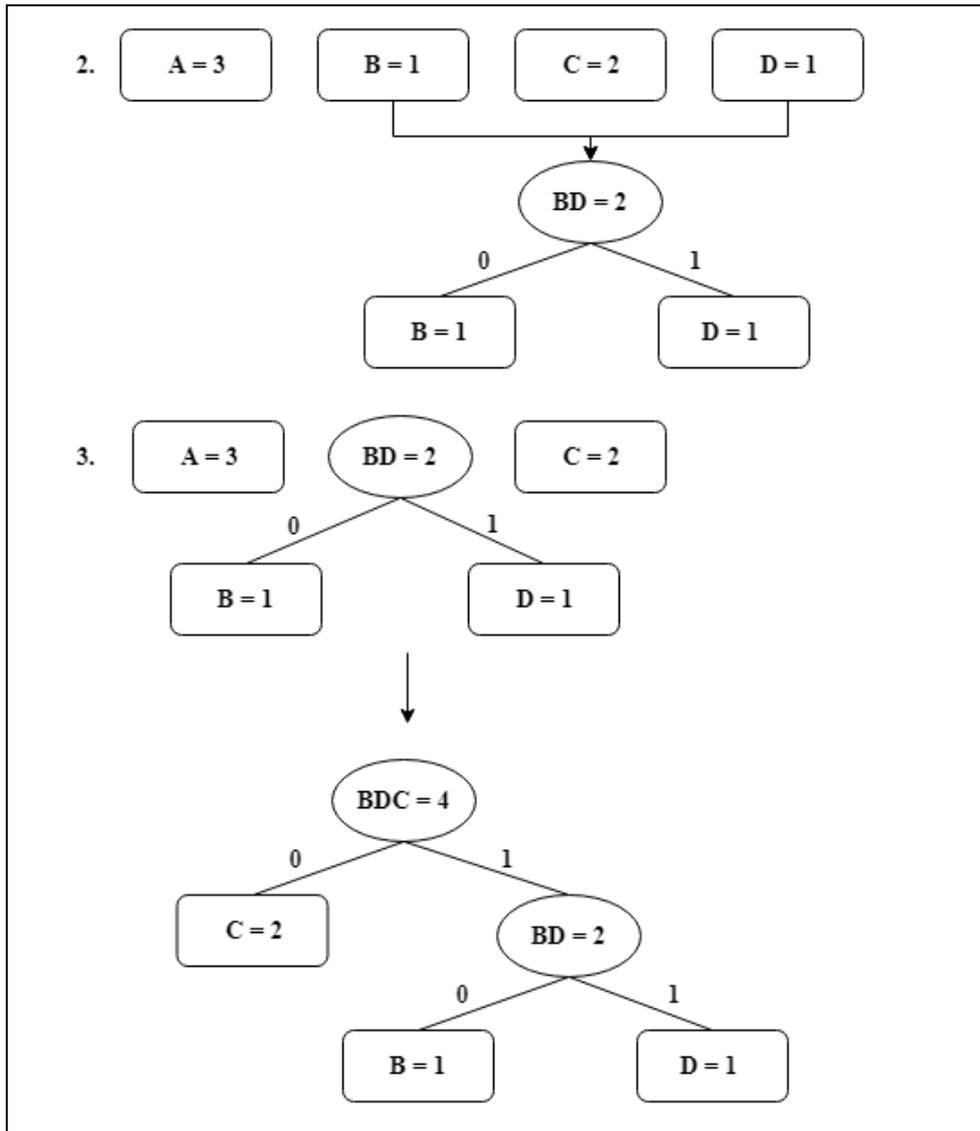
Berikut adalah langkah-langkah untuk membuat pohon Huffman:

- Menentukan frekuensi kemunculan karakter seperti pada Gambar 3.7.

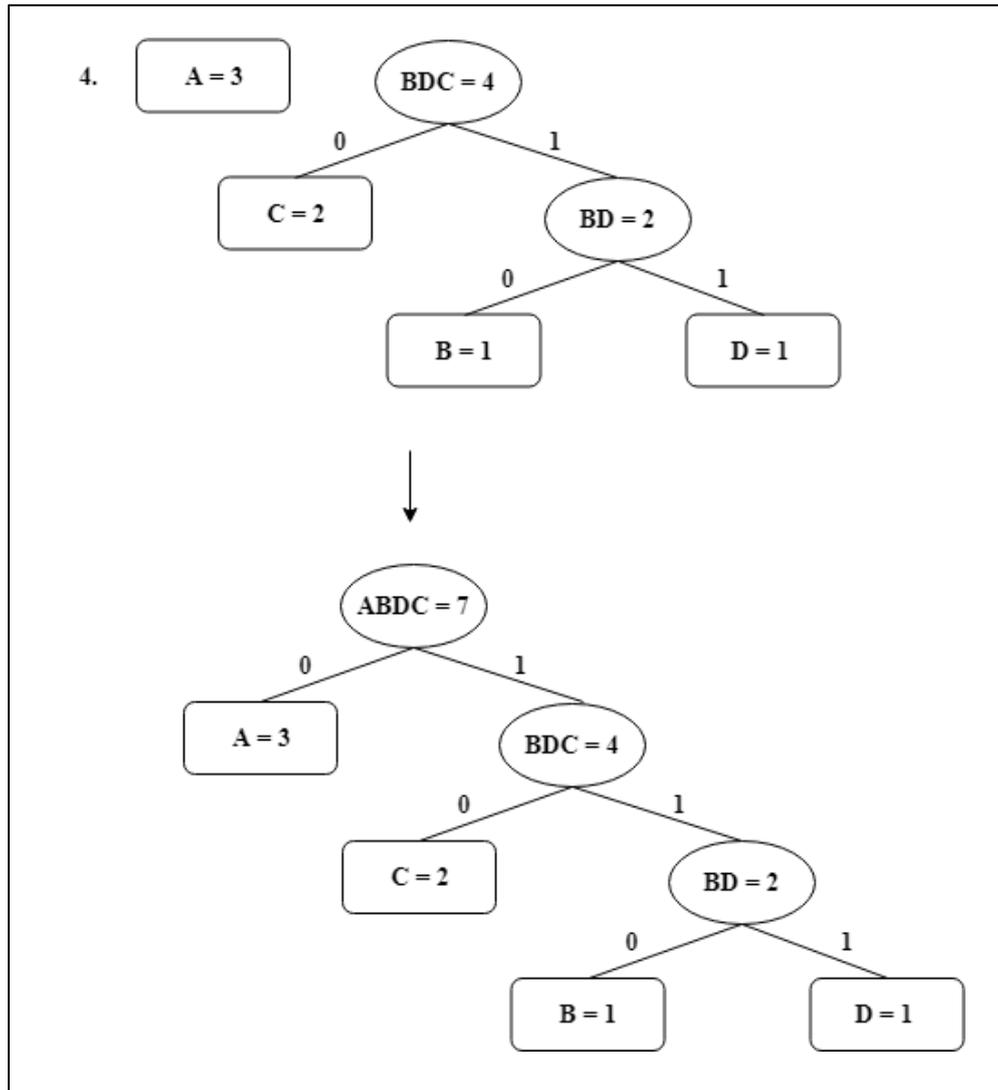


Gambar 3.7 Pengelompokan frekuensi pada string “ABACCCA”

- Melakukan penggabungan karakter dengan frekuensi karakter yang sama atau yang terdekat satu sama lainnya menjadi sebuah cabang.
- Penyusunan cabang dilakukan dari urutan frekuensi terkecil sampai frekuensi terbesar dan bergerak dari bawah ke atas.
- Pada pencabangan pohon Huffman diberi kode angka 0 pada cabang bagian kiri dan 1 pada cabang bagian kanan untuk menandakan pergerakan dari cabang ke cabang pohon pada saat melakukan proses *encoding*. Hasil pengurutan dalam penyusunan pohon Huffman dapat dilihat pada Gambar 3.8 dan Gambar 3.9.



Gambar 3.8 Penyusunan pohon Huffman pada string “ABACCCDA”



Gambar 3.9 Penyusunan pohon Huffman pada String “ABACCCDA”

Tahap kedua adalah proses *encoding* dengan langkah-langkah sebagai berikut:

- Menentukan identitas baru berdasarkan pohon Huffman.
- Cara menentukannya dengan menelusuri pohon Huffman dari atas sampai tertuju pada karakter yang bersangkutan.
- Representasi identitas karakter diambil dari kode setiap pergerakan cabang.

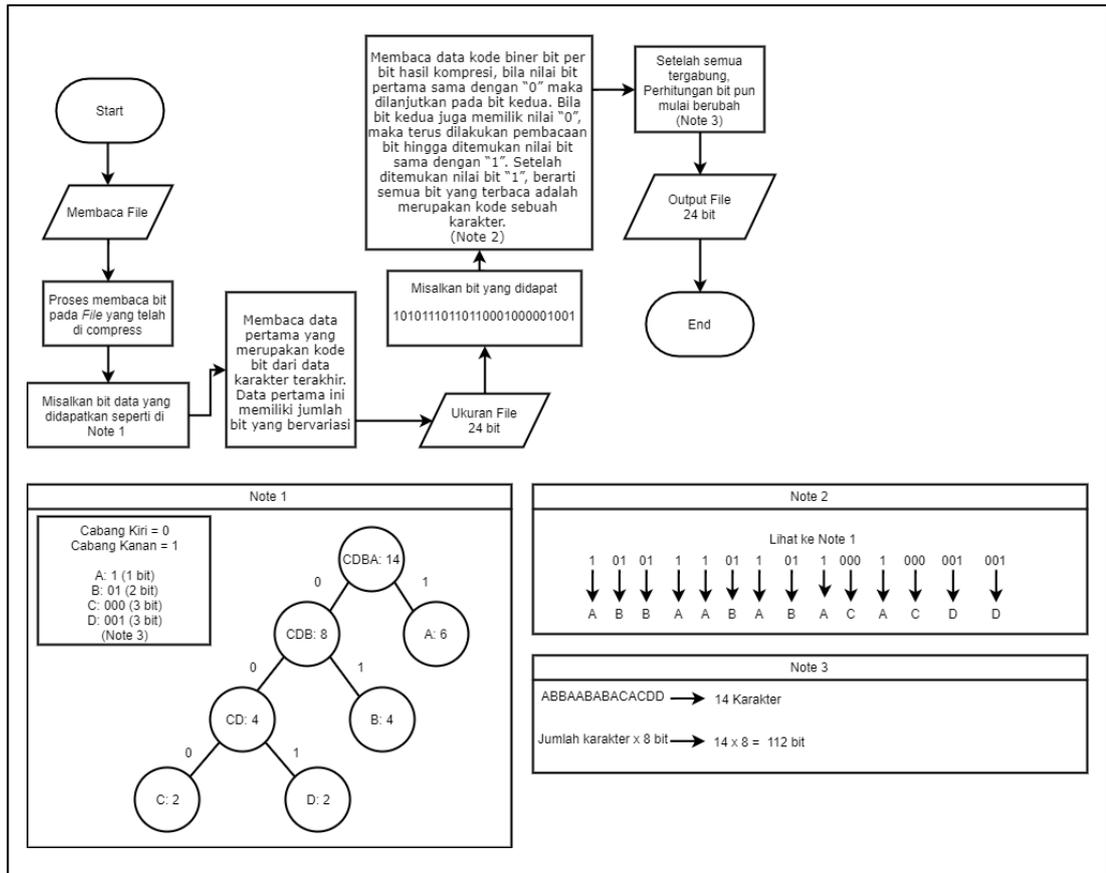
- Contohnya pada *string* “ABACCDA”, karakter pertama adalah “A”, maka identitas bilangan biner berdasarkan pohon Huffman-nya adalah 0, dilanjutkan ke karakter kedua yaitu “B”, identitas bilangan binernya adalah 110. Hasil representasi pohon Huffman untuk string “ABACCDA” dapat dilihat pada Tabel 3.2.
- Gabungkan hasil representasi identitas dengan bilangan biner sebelumnya menjadi 0110, dan berlangsung terus sampai semua karakter sudah di *encode* menjadi 0110010101110.

Tabel 3.2 Hasil representasi pohon Huffman

Contoh	Nilai Bit
A	0
B	110
C	10
D	111

3.5 Dekompres Huffman

Tahap ketiga yaitu proses *decoding* adalah proses kebalikan dari *encoding*, yaitu mengembalikan dari angka-angka biner yang pendek diubah lagi menjadi karakter yang panjang lagi dan tanpa ada kehilangan data dari data asli. Gambar 3.10 merupakan contoh ilustrasi dari proses dekompres algoritma Huffman.



Gambar 3.10 Flowchart Dekompres *File* menggunakan algoritma Huffman.

Seperti pada gambar, hasil representasi bit dikembalikan menjadi bentuk *string* berdasarkan frekuensi karakter pohon biner yang sudah ada. Pengembalian ke bentuk *string* dimulai dengan melakukan pembacaan representasi bit. Pertama sistem harus mengetahui informasi dari representasi bit yang telah dibuat berdasarkan pohon binernya.

Informasi tersebut berupa data hasil representasi bit dan frekuensi karakter pada pohon biner serta tabel yang menyimpan karakter berdasarkan representasi bitnya. Apabila pembacaan representasi bit dimulai dari nilai “0” maka pembacaan akan diteruskan sampai menemukan nilai “1” atau panjang nilai “0” sama dengan lengan pada pohon biner.

3.6 Perancangan Antar Muka

Perancangan antarmuka menjadi dasar untuk melakukan pembuatan sistem usulan. Pada aplikasi ini tampilan antarmuka dibagi menjadi tiga bagian. Halaman menu *Send* yang juga merupakan menu utama aplikasi, menu *Receive*, dan menu *File*. Pada setiap halaman terdapat tiga tombol navigasi agar pengguna dapat berpindah ke halaman yang bersangkutan.

Tampilan utama saat aplikasi dibuka adalah halaman menu *Send*, tampilan halaman ini dapat dilihat pada Gambar 3.11.



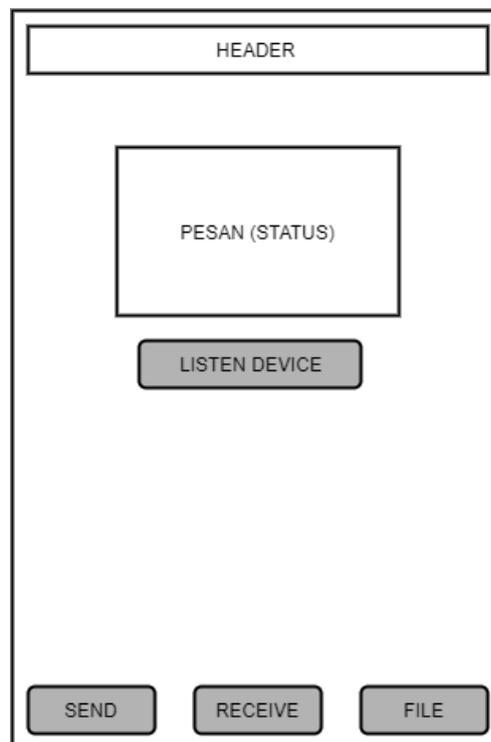
Gambar 3.11 Rancangan halaman Menu *Send*

Apabila aplikasi dijalankan ketika Bluetooth dalam keadaan tidak aktif, maka pada menu *Send* akan muncul sebuah pesan *toast* untuk mengaktifkan Bluetooth terlebih dahulu. Apabila perangkat tidak terhubung dengan perangkat lain, maka

status pada *header* akan menampilkan pesan “*Disconnected*”, pesan ini dapat berubah apabila perangkat telah terhubung dengan perangkat lain.

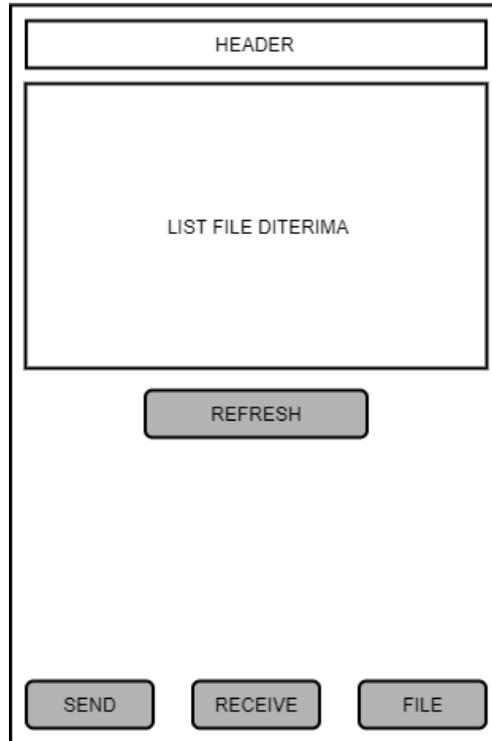
Tombol *browse file* akan meminta pengguna untuk memilih *file*. Tombol *send file* berfungsi sebagai tombol yang akan mengeksekusi tugas aplikasi untuk mengompres dan mengirimkan *file*, apabila ketika tombol *send file* ditekan dalam kondisi *file* belum dipilih atau perangkat tidak terhubung ke perangkat lain, maka program akan memberikan peringatan dan menyarankan pengguna untuk memperbaiki kesalahan.

Pada halaman menu *Receive* terdapat status koneksi, status *message box* dan tombol *listen device* dengan susunan seperti pada Gambar 3.12.



Gambar 3.12 Ranacangan halaman Menu *Receive*

Menu ketiga adalah menu *File*. seperti pada Gambar 3.13, menu ini hanya menampilkan sebuah daftar nama *file* dan sebuah tombol untuk memuat ulang daftar tersebut.



Gambar 3.13 Rancangan halaman Menu *File*