BAB 3
METODOLOGI PENELITIAN

3.1 Metodologi Penelitian

Pada bagian ini dijabarkan langkah-langkah yang hendak dilakukan dalam menyusun dan mengerjakan penelitan. Langkah-langkah penelitian akan dijabarkan sebagai berikut.

1. Studi Literatur

   Pada langkah studi literatur akan dilakukan pembelajaran pada teori-teori dengan mencari literatur, jurnal, *paper*, dan referensi lain dari penelitian sebelumnya yang berhubungan dengan topik yang dipilih.

2. Analisis Kebutuhan

   Dalam langkah analisis kebutuhan, hal yang dilakukan adalah menganalisa dan merancang kebutuhan aplikasi. Hasil dari rancangan dan analisa ini akan diterjemahkan untuk melakukan penelitian.

3. Perancangan

   Perancangan aplikasi adalah langkah atau proses dilakukannya berbagai perancangan yang dibutuhkan oleh aplikasi, yaitu *Data Flow Diagram* atau DFD *flowchart*, *Entity Relationship Diagram*, *Database Schema*, dan rancangan antar muka.

4. Implementasi

   Langkah implementasi merupakan langkah di mana hasil dari perancangan akan diimplementasikan ke dalam *code* dengan bahasa pemograman yang telah dipilih yaitu JAVA, dan *tools* yaitu Android Studio, NFC Reader, dan API OCR.

5. *Testing* dan *Debugging*

   Langkah *testing* dan *debugging* akan dilakukan bersamaan dengan implementasi dengan tujuan untuk mengetahui kekurangan dan melakukan perbaikan pada aplikasi yang dibuat serta melakukan pengukuran akurasi pada pengambilan data tamu dari *tools* yang digunakan. Pengukuran akurasi akan menggunakan cara *black box testing*.

6. Konsultasi dan Penulisan Laporan

   Pada langkah ini akan dilakukan pencatatan dari pembangunan aplikasi untuk menjadi dokumentasi dan bisa digunakan untuk sarana ilmu pengatahuan atau menjadi bahan referensi untuk penelitian kedepannya.

3.2 Analisis Kebutuhan

Analisis kebutuhan yang menentukan seluruh kebutuhan aplikasi. Analisis kebutuhan dibagi menjadi dua yaitu analisis kebutuhan fungsional dan analisis kebutuhan non fungsional.

3.2.1 Analisis Kebutuhan Fungsional

1. Sistem aplikasi harus memiliki halaman *login* agar pengguna dapat mengakses halaman utama.
2. Sistem aplikasi dapat melakukan pengolahan data yaitu :
   (a) Dapat melihat, menambah, dan menghapus data tamu
   (b) Dapat melakukan *backup* data tamu
3. Sistem aplikasi dapat mendeteksi teknologi NFC pada perangkat pintar yang digunakan.

4. Sistem aplikasi dapat membaca dan mengambil *Tag-id* dengan NFC dari KTP elektronik.
5. Sistem aplikasi dapat mengubah status tamu
6. Sistem aplikasi dapat menampilkan pesan dalam :
   (a) melakukan *login*
   (b) menambah, dan menghapus tamu
   (c) mengganti status tamu
7. Sistem aplikasi dapat mencari tamu dari *Tag-id*
8. Sistem aplikasi dapat melakukan *logout*

3.2.2 Analisis Kebutuhan Non Fungsional

1. *Operational*
   (a) Perangkat Keras dan Lunak untuk membangun sistem.
      - Prosesor: AMD Ryzen 3 2200G with Radeon Vega Graphics
      - RAM: 8GB
      - VGA: MSI AMD Radeon RX570
      - Windows 10 Pro 64-bit
      - IDE : Android Studio
      - Firebase
      - OCR *Library*
   (b) Perangkat Keras dan Lunak untuk pengujian sistem.
      - *Smartphone* Redmi POCO X3 NFC.
      - Sistem Operasi Android versi 10.0
2. *Security*

(a) Dilengkapi dengan *email* dan *password* untuk mengakses halaman utama aplikasi

3. *Information*

(a) Ditampilkan pesan apabila input *username* dan *password* salah

(b) Ditampilkan pesan bila salah satu input pada tambah tamu baru kosong

(c) Ditampilkan pesan bila data tamu dihapus

(d) Ditampilkan pesan bila logout

4. *Performance*

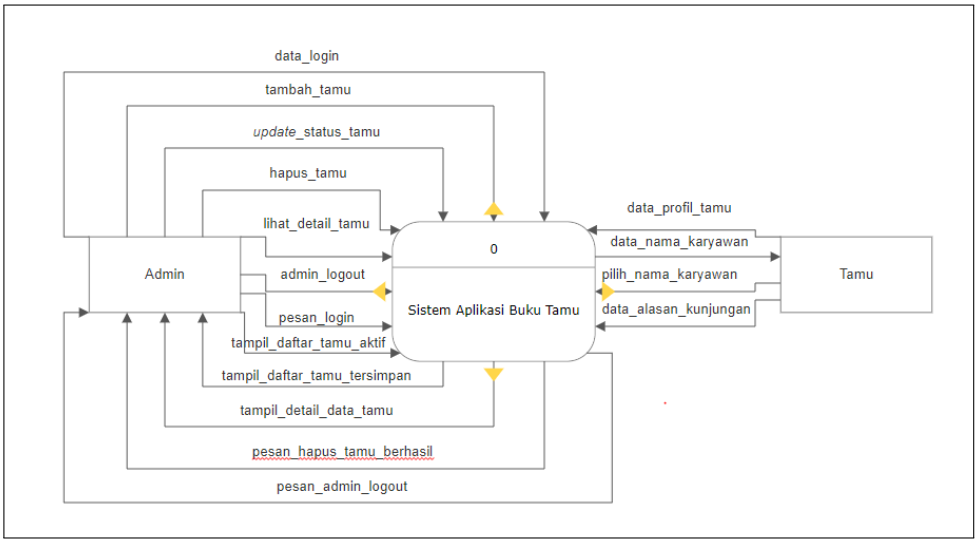
(a) Aplikasi dapat digunakan secara *offline* dan *online*

(b) Data tamu tersimpan pada database lokal jadi bisa dilakukan penambahan tamu baru secara *offline*

3.3 Perancangan Aplikasi

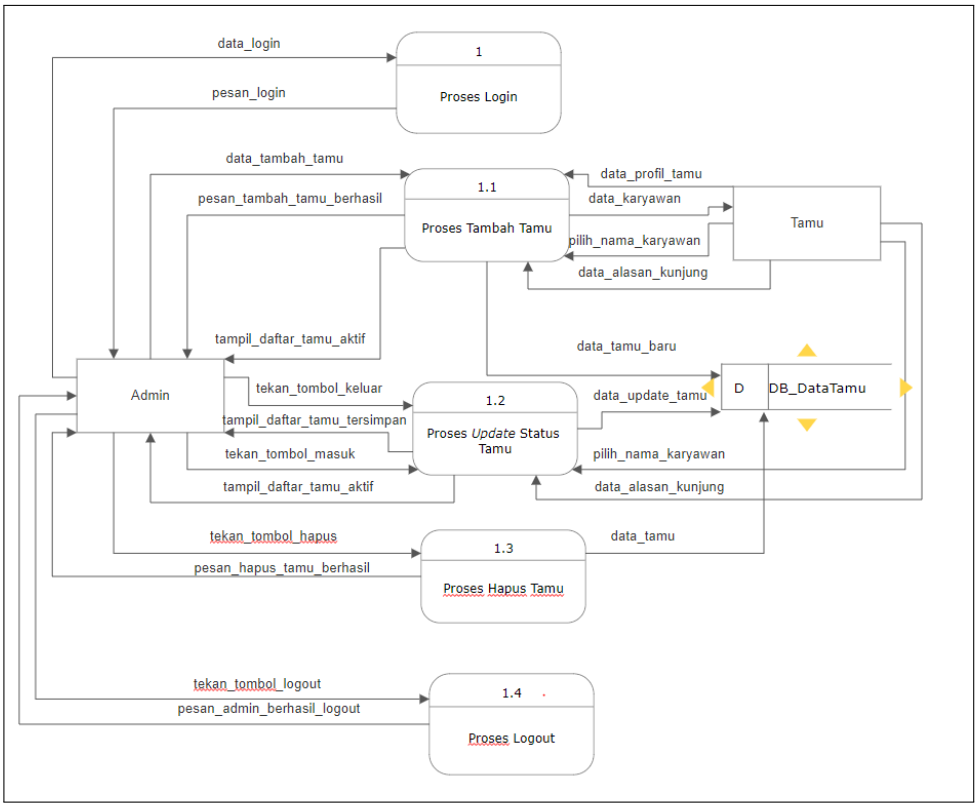
Perancangan aplikasi direpresentasikan menggunakan *flowchart* yang menggambarkan alur proses yang berjalan pada aplikasi, *Data Flow Diagram* yang berfungsi untuk menggambarkan aliran data dari suatu proses ke proses yang lain pada aplikasi. Terdapat *flowchart* bagian utama yang menjelaskan alur kerja aplikasi secara keseluruhan, kemudian dibagi secara lebih mendetail pada bagian berikutnya. Selain itu terdapat pembuatan *Entity Relationship Diagram* yang berfungsi untuk menggambarkan hubungan antar tabel di dalam struktur database yang digunakan oleh aplikasi. Dalam perancangan aplikasi juga terdapat pembuatan tampilan aplikasi antarmuka atau disebut juga *User Interface*.

3.3.1 Data Flow Diagram



Gambar 3.1. *Data Flow Diagram* Level 0

Pada Gambar 3.1 dapat dilihat *Data Flow Diagram Level 0*. Diagram ini menjelaskan tentang garis besar alur data dari setiap entitas ke sistem. Entitas yang dimaksud yaitu Admin dan Tamu. Entitas Admin ini bisa melakukan login atau logout, dan yang mengurus dibagian penambahan tamu baru, mengubah status tamu aktif atau tersimpan, menghapus data tamu di dalam sistem aplikasi. Balasan dari sistem untuk Admin yaitu menampilkan pesan tambah tamu telah berhasil, menampilkan daftar tamu aktif, dan tersimpan, menampilkan pesan data tamu berhasil dihapus, dan juga menampilkan pesan admin berhasil *logout* atau keluar. Sedangkan untuk entitas Tamu ini yang memberikan data profiln berdasarkan KTP-el yang dimiliki, memilih nama karyawan yang ingin dikunjung, dan memberikan data alasan kunjungan ke sistem. Namun tidak ada balasan apapun dari sistem kepada entitas Tamu.



Gambar 3.2. *Data Flow Diagram* Level 1

Kemudian dilanjutkan dengan *Data Flow Diagram Level 1* yang dapat dilihat pada Gambar 3.2. Terdapat lima proses yang terjadi di dalam sistem aplikasi. Kelima proses tersebut dapat dijelaskan sebagai berikut.

1. Proses *Login* yang merupakan proses di mana admin perlu memasukkan data *email* dan *password* untuk *login*, dan masuk ke halaman utama aplikasi.

2. Proses tambah tamu yaitu proses penambahan tamu baru yang dilakukan oleh admin, dan tamu, di mana tamu hanya memberikan data profil, memilih nama yang ingin dikunjungi, dan alasan kunjungan. Sedangkan admin yang melakukan konfirmasi untuk menambah data tamu baru. Data ini akan tersimpan ke database lokal buku tamu. Data tamu baru itu akan muncul di daftar tamu pada *tab* Tamu Aktif.

3. Proses *update* status tamu yaitu perubahan antara tamu aktif atau masuk menjadi tamu tersimpan atau sudah keluar atau sebaliknya. Proses ini terjadi apa-

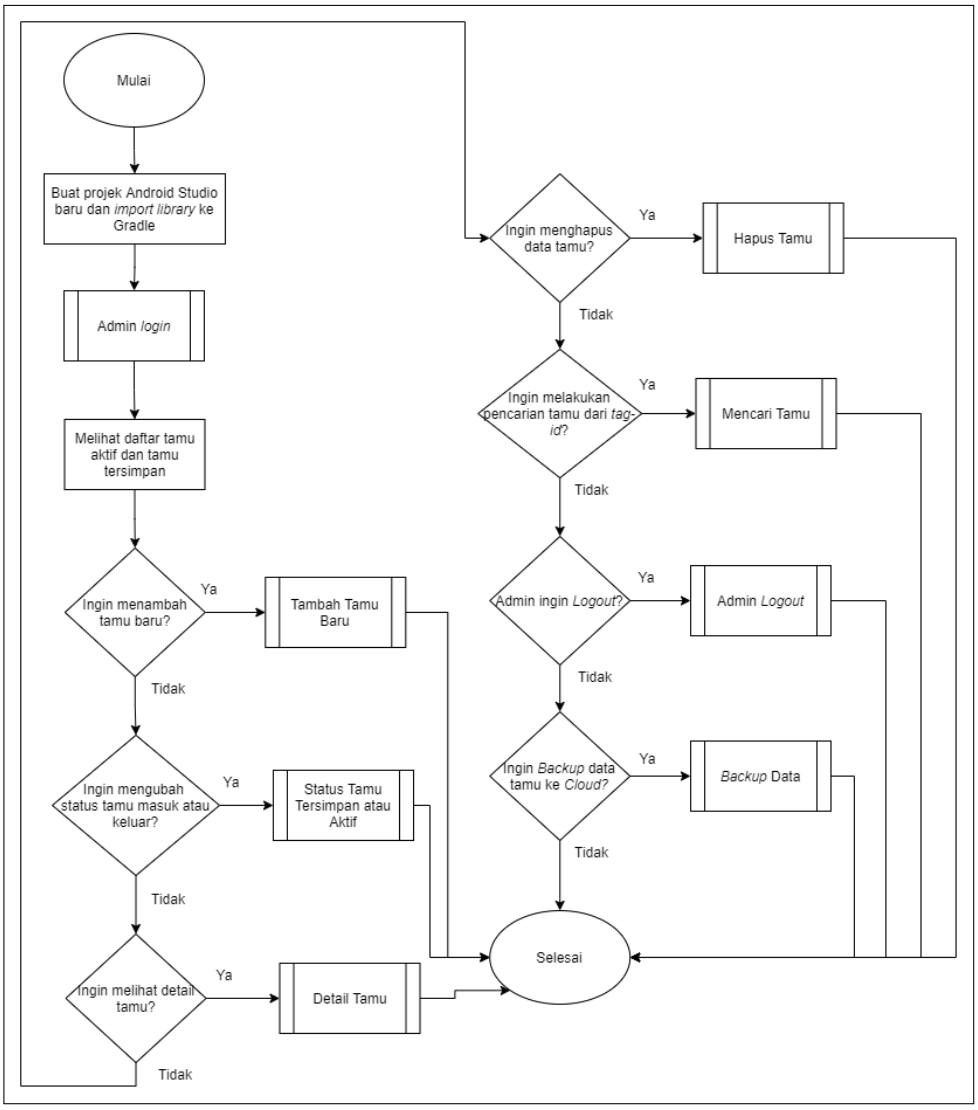
bila admin menekan tombol Keluar untuk mengubah tamu aktif menjadi tersimpan, sedangkan apabila admin menekan tombol Masuk, maka terjadi perubahan tamu tersimpan menjadi tamu aktif. Perubahan ini membuat proses *update* data tamu pada database buku tamu.

4. Proses hapus tamu yaitu proses penghapusan data tamu dari database buku tamu. Proses ini terjadi apabila admin menekan tombol Hapus.

5. Proses terakhir yaitu proses Logout merupakan proses yang terjadi apabila admin menekan *icon logout* pada *toolbar*, dan proses ini membuat admin *logout* atau keluar dari aplikasi, dan menuju ke halaman *login*.

3.3.2 Flowchart Aplikasi

Proses diawali dengan membuat projek aplikasi baru di Android studio, dan melakukan *import library* ke dalam gradle pada Android Studio. *Library* yang digunakan dalam penelitian ini yaitu *library* Firebase Cloud Firestore, ML Kit OCR, dan untuk melakukan *Cropping* gambar. Proses kemudian dilanjutkan pada *user admin login*. Setelah berhasil *login*, maka akan ke halaman utama. Di halaman utama terdapat dua *tab*. *Tab* pertama berisi daftar tamu aktif atau masuk, dan *tab* kedua berisi tamu yang tersimpan atau sudah keluar. Proses kemudian masuk ke fungsionalitas aplikasi yang terdiri dari beberapa modul yaitu modul tambah tamu baru, melakukan perubahan status tamu, melihat detail tamu, menghapus data tamu, mencari tamu, *logout user* admin, dan melakukan *backup* data tamu ke *Cloud Firestore*. Alur proses awal dapat dilihat pada Gambar 3.3.

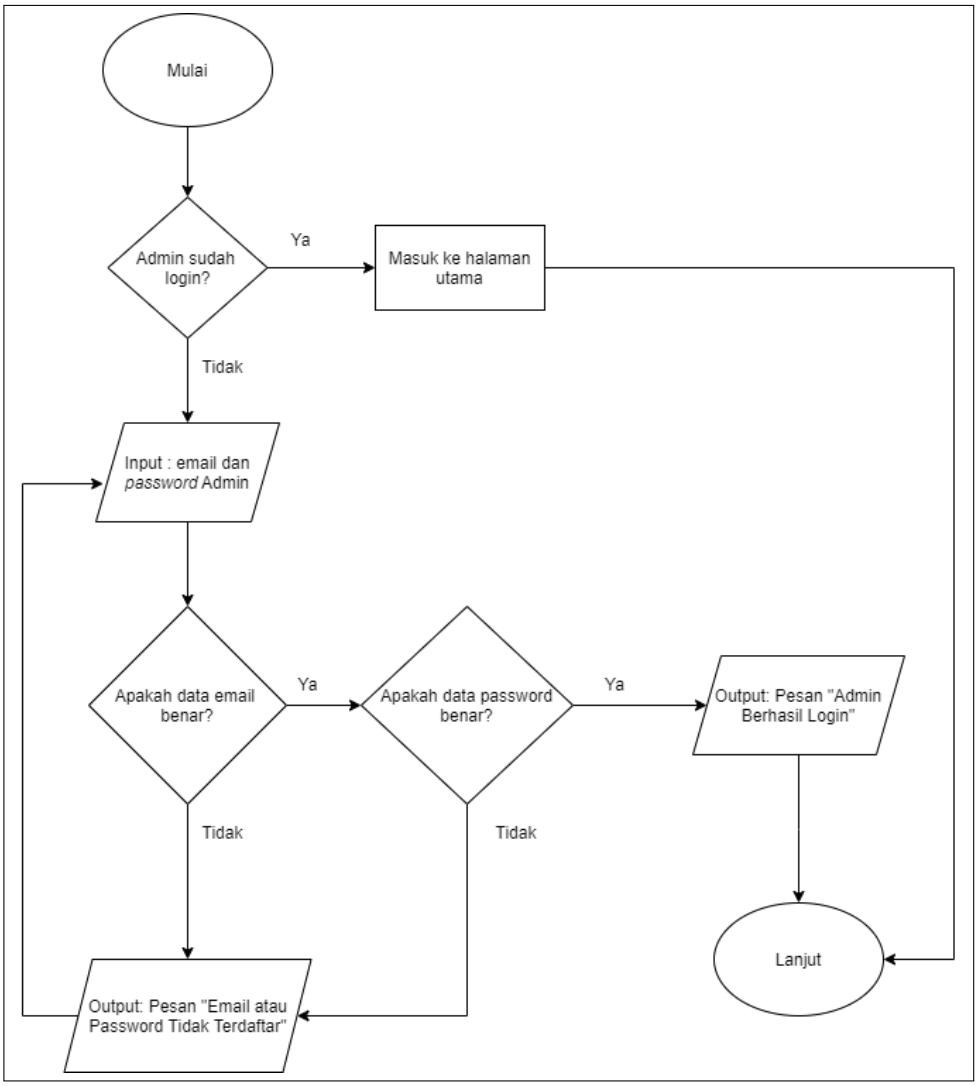


Gambar 3.3. Flowchart awal aplikasi

Setiap modul pada *flowchart* awal aplikasi dijelaskan secara lebih lengkap sebagai berikut.

A Flowchart Admin Login

Pada alur proses Admin login, diawali dengan mengisi atau melakukan *input email* dan *password*. Selanjutnya proses pengecekan *input* data itu akan dilakukan ke Firebase Cloud Firestore. Apabila data ada di dalam Firebase Cloud Firestore, maka proses pengecekan berhasil. Alur proses dapat dilihat pada Gambar 3.4.

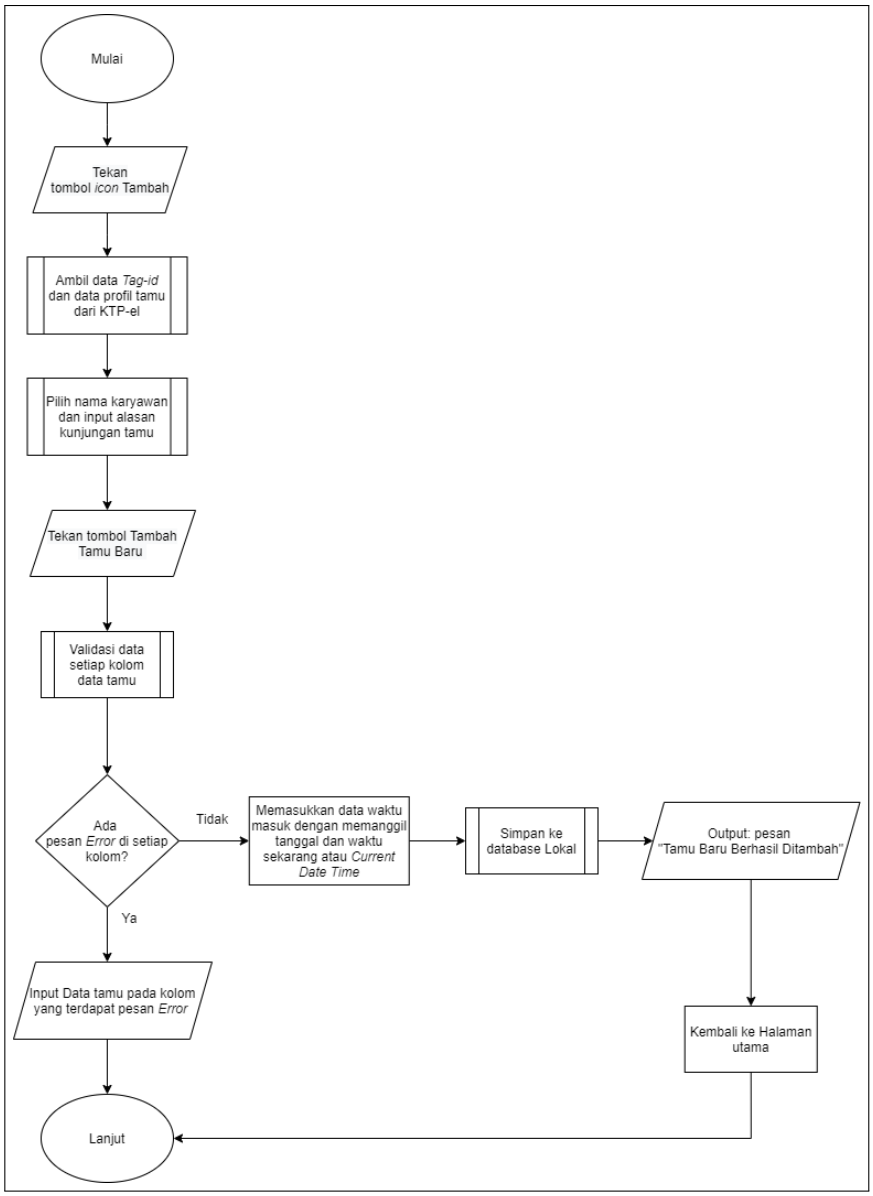


Gambar 3.4. Flowchart halaman Admin *Login*

B Flowchart Tambah Tamu Baru

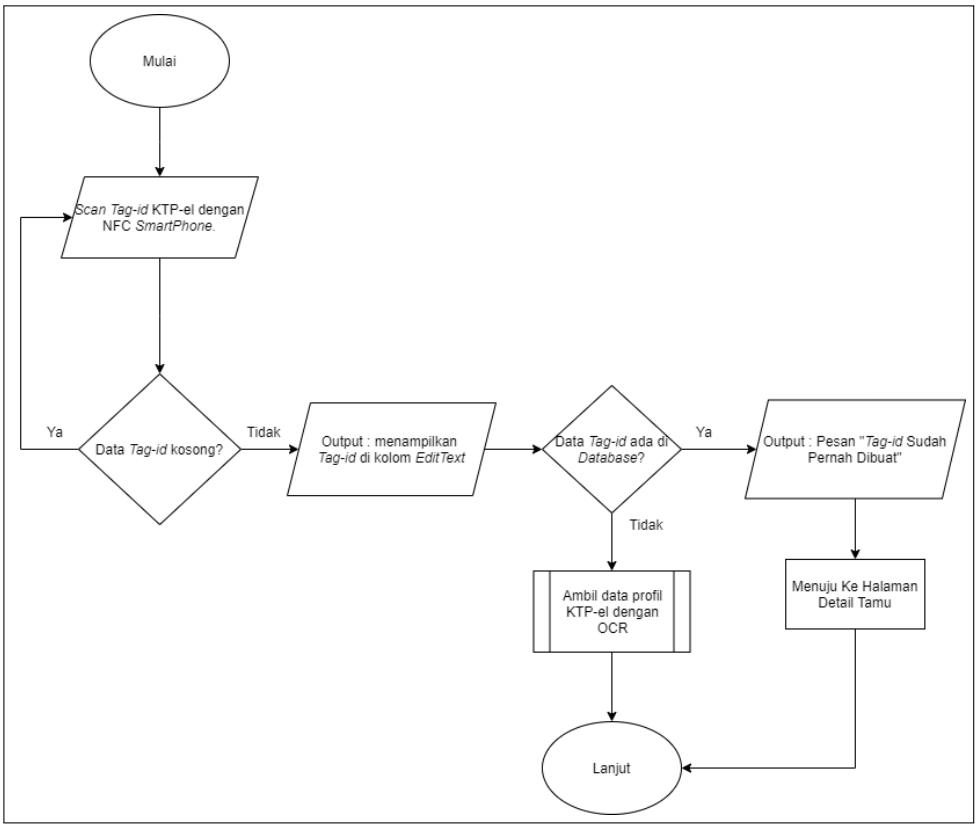
Proses tambah tamu dimulai dengan mengklik tombol *icon* tambah di ujung kanan bawah untuk menuju ke halaman tambah tamu. Proses dilanjutkan dengan mengambil *Tag-id* dan data profil tamu dari KTP-el, kemudian memilih nama residen yang ingin dikunjungi dan mengisi alasan kunjungan tamu. Setelah semua data terisi, maka bisa menekan tombol Tambah Tamu Baru. Selanjutnya proses validasi terjadi, proses ini untuk melakukan pengecekan apakah setiap kolom *input* data tidak ada yang kosong atau *null*. Apabila ketika validasi muncul *error* pada salah

satu kolom data, maka proses penyimpanan data tamu akan berhenti, dan meminta untuk mengisi data yang terkena *error* atau data yang kosong, tetapi bila validasi sudah selesai dan tidak ada *error*, maka sistem akan mengambil waktu masuk tamu baru berdasarkan tanggal dan waktu sekarang atau *current date time*. Lalu setelah mendapatkan waktu masuk, maka semua data tamu baru akan masuk ke database lokal SQLite. Proses tambah tamu akan selesai setelah muncul pesan ”Tamu Baru Berhasil Ditambah,” dan kembali ke halaman utama aplikasi. *Flowchart* tambah tamu baru bisa dilihat pada Gambar 3.5.



Gambar 3.5. Flowchart tambah tamu

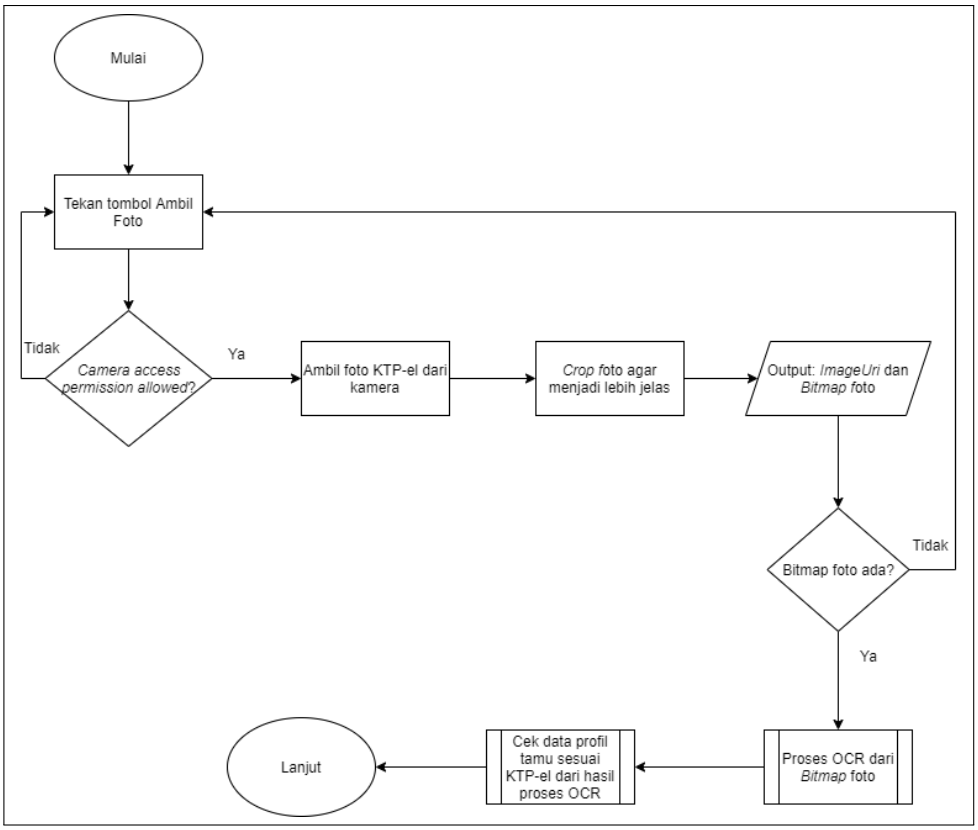
Pada proses pengambilan *Tag-id* dan data profil tamu dari KTP-el terdapat teknologi yang dipakai. Untuk *Tag-id* dengan menggunakan teknologi NFC yang terdapat pada *smartphone*. Proses pengambilan itu dengan cara melakukan *scan* KTP elektronik pada bagian NFC *smartphone*. Dari proses *scan* itu didapatkan *Tag-id* yang berupa bilangan desimal, dan heksadesimal. Karena *Tag-id* harus merupakan data yang unik, maka data *Tag-id* yang dipakai yaitu dalam bentuk heksadesimal. Sebelum data *Tag-id* dimasukkan ke kolom *EditText*, dilakukan proses pengecekan apakah data *Tag-id* ini sudah pernah ada atau dimasukkan ke dalam database. Jika sudah pernah ada, maka akan muncul pesan "Tag-id Sudah Pernah Dimasukkan", kemudian akan pindah ke halaman detail tamu yang memiliki *Tag-id*. Alur proses pengambilan *Tag-id* bisa dilihat pada Gambar 3.6.



Gambar 3.6. Flowchart Pengambilan *Tag-id* KTP-el

Setelah *Tag-id* diambil maka proses berikutnya yaitu pengambilan data profil tamu dari KTP-el dengan cara melakukan pengambilan foto KTP-el dan kemu-

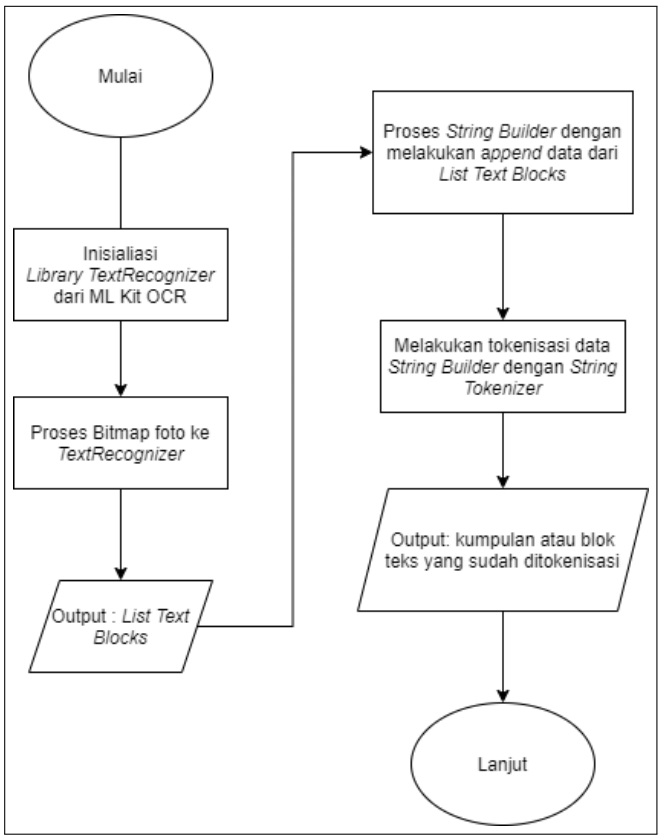
dian dilakukan *crop* atau pemotongan foto agar mendapatkan kualitas foto lebih baik. Dari hasil *crop* foto akan muncul *Image Uri* dan Bitmap. Bitmap sendiri merupakan sebuah gambar yang terbentuk dari berbagai titik dengan perpaduan warna yang mewakili setiap *pixel* dan kemudian ditampilkan di layar monitor (Zakaria, 2020). Selanjutnya bitmap foto akan dimasukkan kedalam proses pembacaan teks dengan OCR (*Optical Character Recognition*). Setelah proses OCR selesai, lalu dilakukan pengecekan data dari hasil OCR dengan format data profil pada KTP-el. Alur proses pengambilan foto dapat dilihat pada Gambar 3.7.



Gambar 3.7. Flowchart pengambilan Foto KTP-el

Proses OCR dimulai dengan menginisialisasi *library TextRecognizer* dari ML Kit OCR yang merupakan salah satu *tools* OCR buatan dari Firebase. Kemudian memproses Bitmap foto ke *TextRecognizer*, dari proses ini mendapatkan *output List TextBlocks* data KTP-el. *List TextBlocks* ini mewakili blok-blok teks persegi panjang yang berisi nol atau beberapa objek kata dan entitas seperti tanggal, angka,

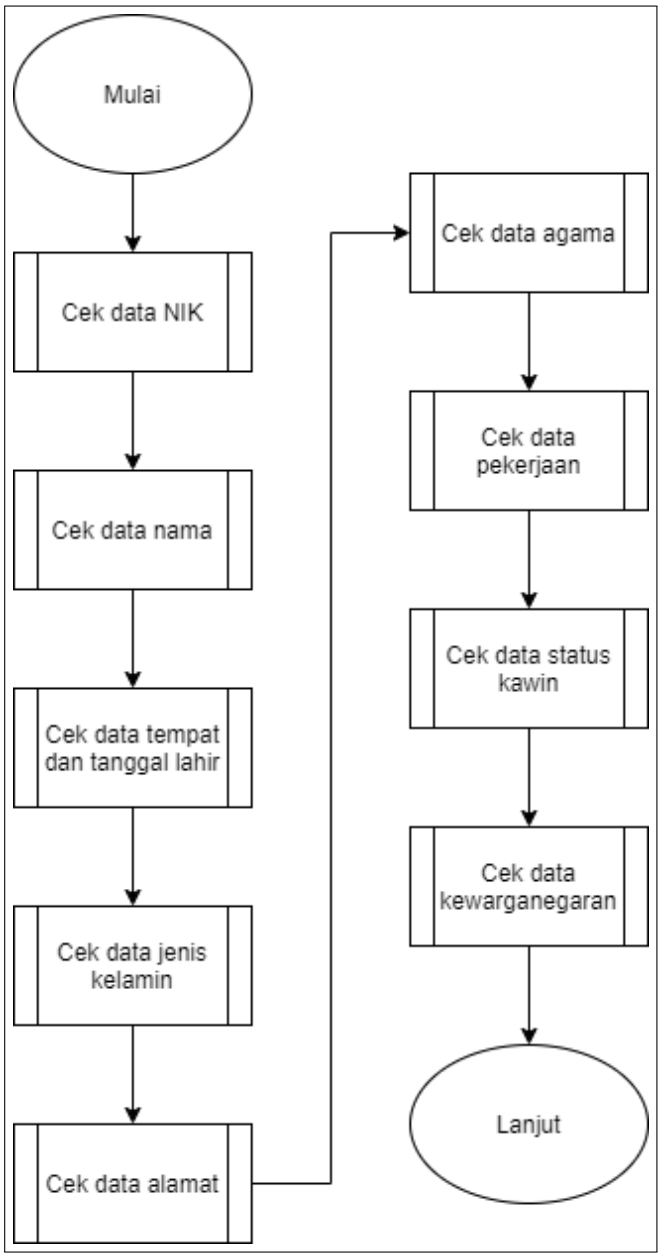
dan sebagainya. Proses selanjutnya yaitu melakukan *append* atau memasukkan data dari *List TextBlocks*, dan *append* kata pemisah yaitu *slash* atau ”/” ke *String Builder*. *String Builder* digunakan karena *List TextBlocks* yang didapatkan dalam bentuk blok teks yang tidak beraturan, dan agar dapat dengan mudah melakukan pemisahan data dari *List TextBlocks* itu. Setelah dijadikan *String Builder*, kemudian dilakukan tokenisasi dengan menggunakan *StringTokenizer* untuk membagi setiap teks yang diberikan kata pemisah yaitu seperti kata *slash*(”/”) atau spasi. *Output*-nya yaitu blok-blok teks yang sudah di tokenisasi. *Flowchart* pada proses OCR dapat dilihat pada Gambar 3.8.



Gambar 3.8. Flowchart proses OCR bitmap foto

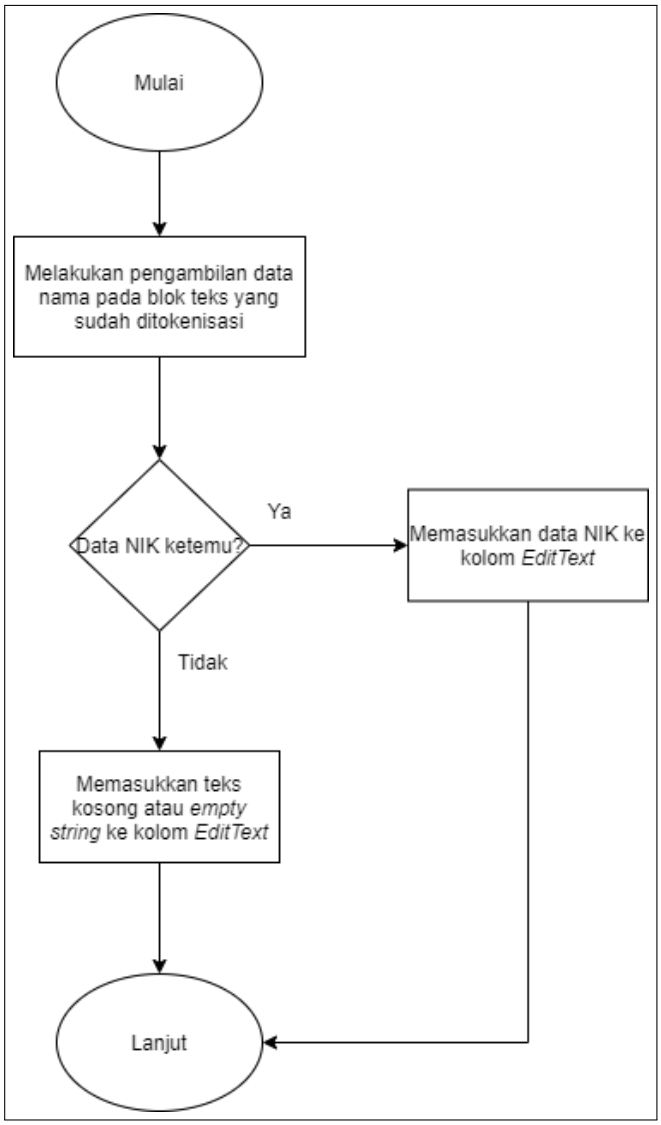
Dari hasil tokenisasi, dilakukan proses pengecekan atau pencarian data untuk menyesuaikan dengan format data profil pada KTP-el. Proses ini memiliki beberapa modul yaitu modul untuk mengecek data NIK, nama, tempat dan tanggal lahir, jenis kelamin, alamat, agama, pekerjaan, status kawin, dan kewarganegaraan

tamu. *Alur* prosesnya dapat dilihat pada Gambar 3.9.



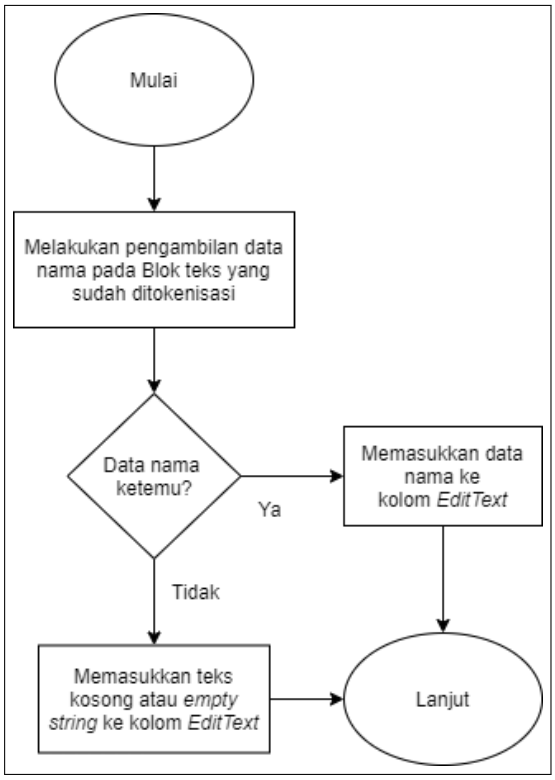
Gambar 3.9. Flowchart cek data tokenisasi dengan format KTP-el

Pengecekan data NIK dengan cara mencari potongan kata NIK pada teks yang sudah ditokenisasi, apabila ketemu, maka data tersebut dimasukkan ke dalam kolom *EditText*. Tetapi, apabila data tidak ketemu, maka dimasukkan teks kosong atau *empty string* ke dalam kolom *EditText*. Alur proses pengecekan data NIK dapat dilihat pada Gambar 3.10.



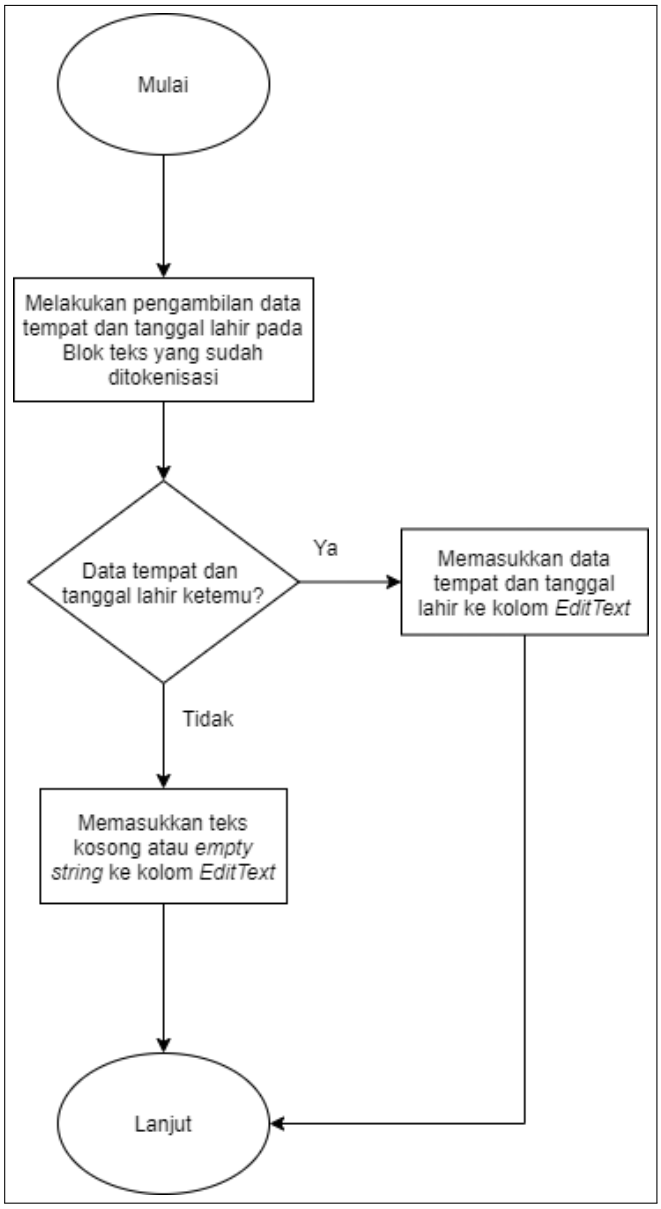
Gambar 3.10. Flowchart cek data NIK dengan hasil teks tokenisasi

Selanjutnya melakukan pengecekan data nama tamu dengan cara yang sama yaitu mencari potongan kata nama pada teks yang sudah di tokenisasi. Apabila ada, maka data tersebut dimasukkan ke dalam kolom *EditText* nama, tetapi apabila tidak ada, maka teks kosong atau *empty string* akan dimasukkan ke dalam kolom *EditText*. *Flowchart* pengecekan nama tamu dapat dilihat pada Gambar 3.11.



Gambar 3.11. Flowchart cek data nama dengan hasil teks tokenisasi

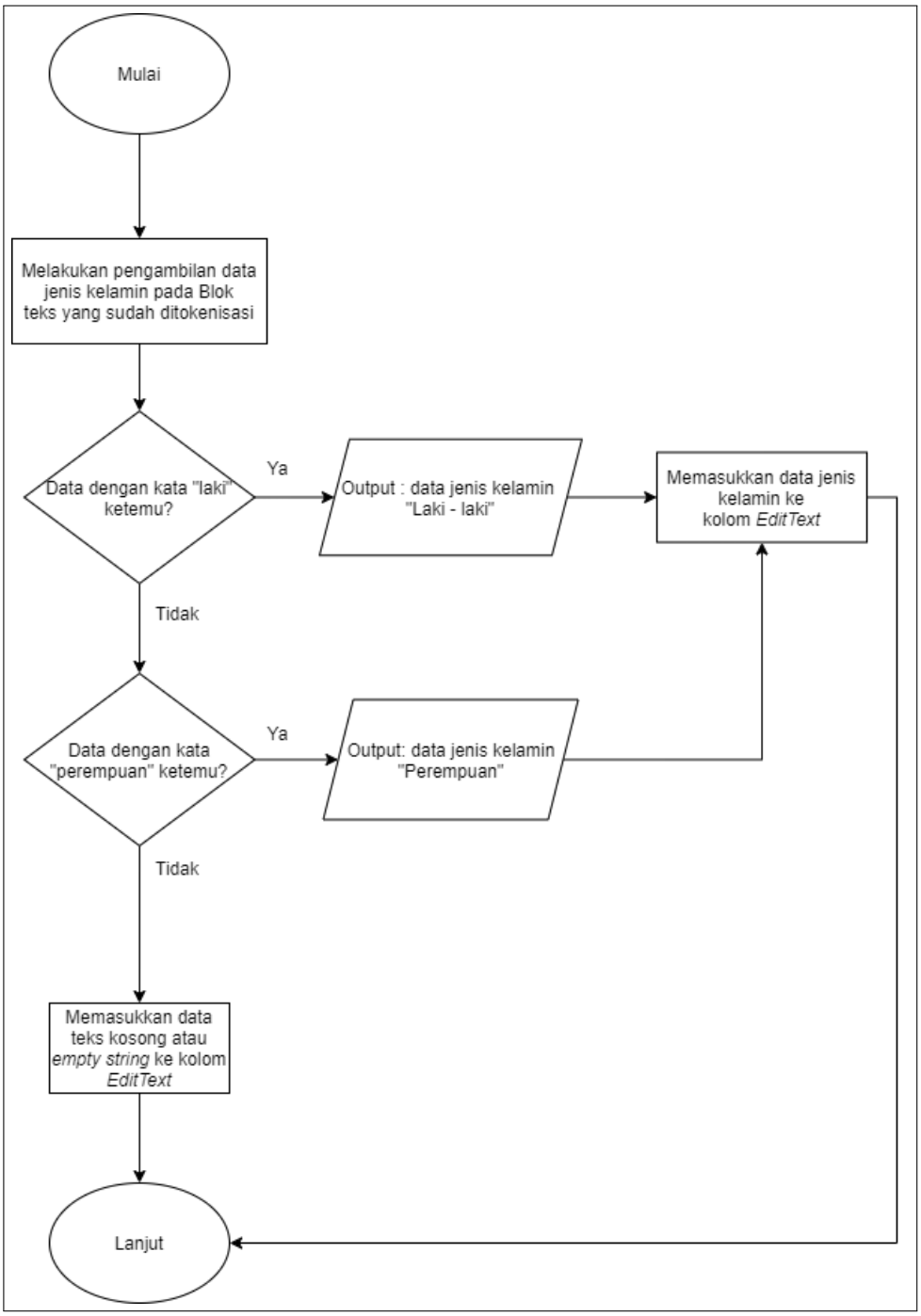
Cara pengecekan data tempat dan tanggal lahir tamu juga sama seperti pengecekan data NIK, dan nama yaitu dengan mencari potongan kata tempat dan tanggal lahir pada teks yang sudah ditokenisasi. Apabila ketemu, maka potongan kata tersebut akan masuk ke dalam kolom *EditText* tempat dan tanggal lahir, tetapi apabila tidak ketemu, maka teks kosong akan dimasukkan ke dalam *EditText*. Alur proses pengecekan tempat dan tanggal lahir dapat dilihat pada Gambar 3.12.



Gambar 3.12. Flowchart cek data tempat dan tanggal lahir dengan hasil teks tokenisasi

Untuk pengecekan data jenis kelamin tamu yaitu dengan mencari kata ”laki” dan kata ”perempuan” pada hasil teks yang sudah di tokenisasi, apabila kata ”laki” ketemu, maka data jenis kelamin akan diisi kata ”Laki - laki”, tetapi apabila kata ”perempuan” ketemu, maka data jenis kelamin akan diisi kata ”Perempuan”.Kemudian data tersebut dimasukkan ke dalam kolom *EditText* jenis kelamin. Namun apabila antara dua kata itu tidak ada atau tidak ketemu, maka pada kolom *EditText* jenis kelamin dimasukkan teks kosong. Alur proses pengecekan data jenis

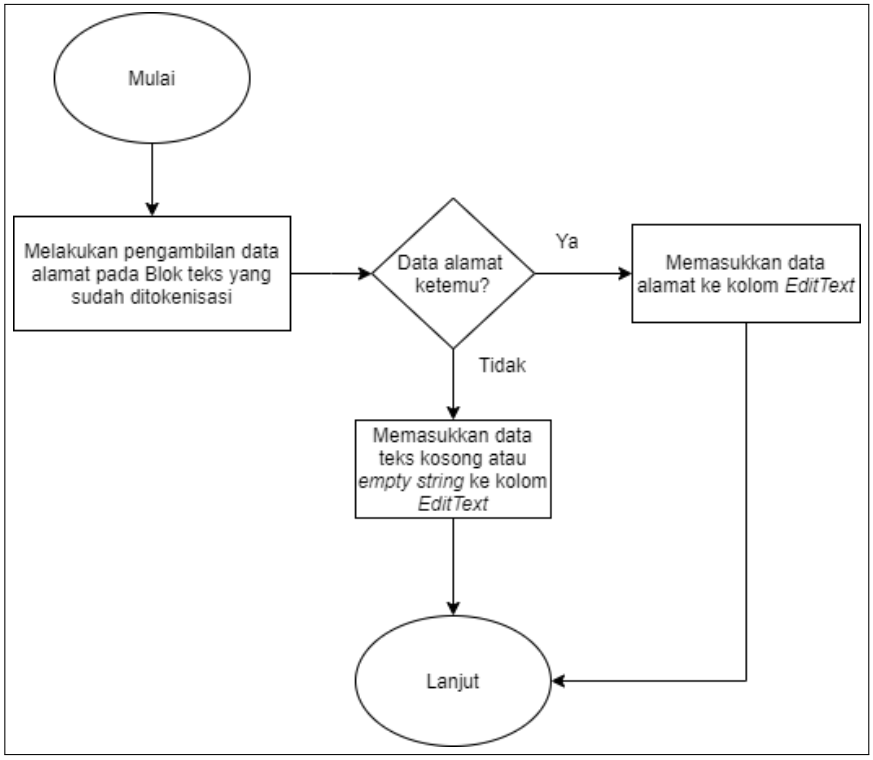
kelamin ini dapat dilihat pada Gambar 3.13.



Gambar 3.13. Flowchart cek data jenis kelamin dengan hasil teks tokenisasi

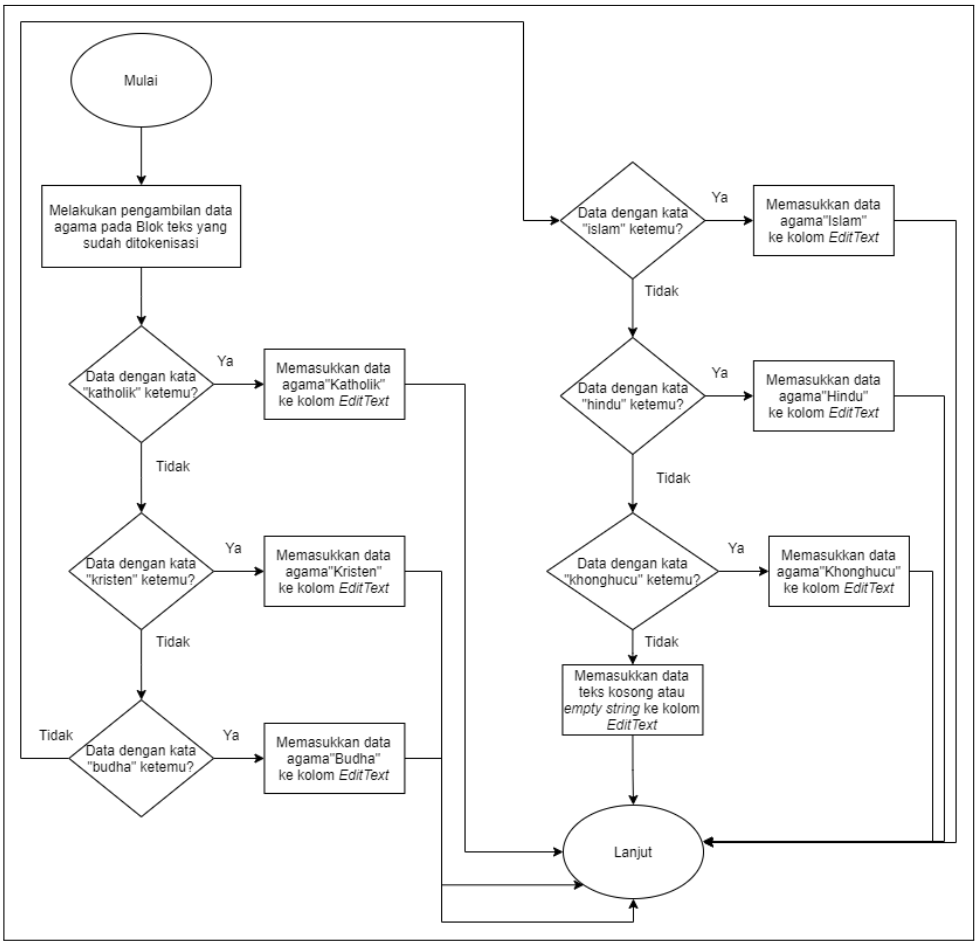
Berikutnya yaitu pengecekan data alamat tamu, caranya yaitu mencari potongan kata pada hasil teks tokenisasi. Jika kata tersebut didapatkan, maka kata tersebut akan masuk ke dalam *EditText* alamat, tetapi jika tidak, maka teks kosong

akan dimasukkan ke dalam *EditText*. Alur prosesnya dapat dilihat pada Gambar 3.14.



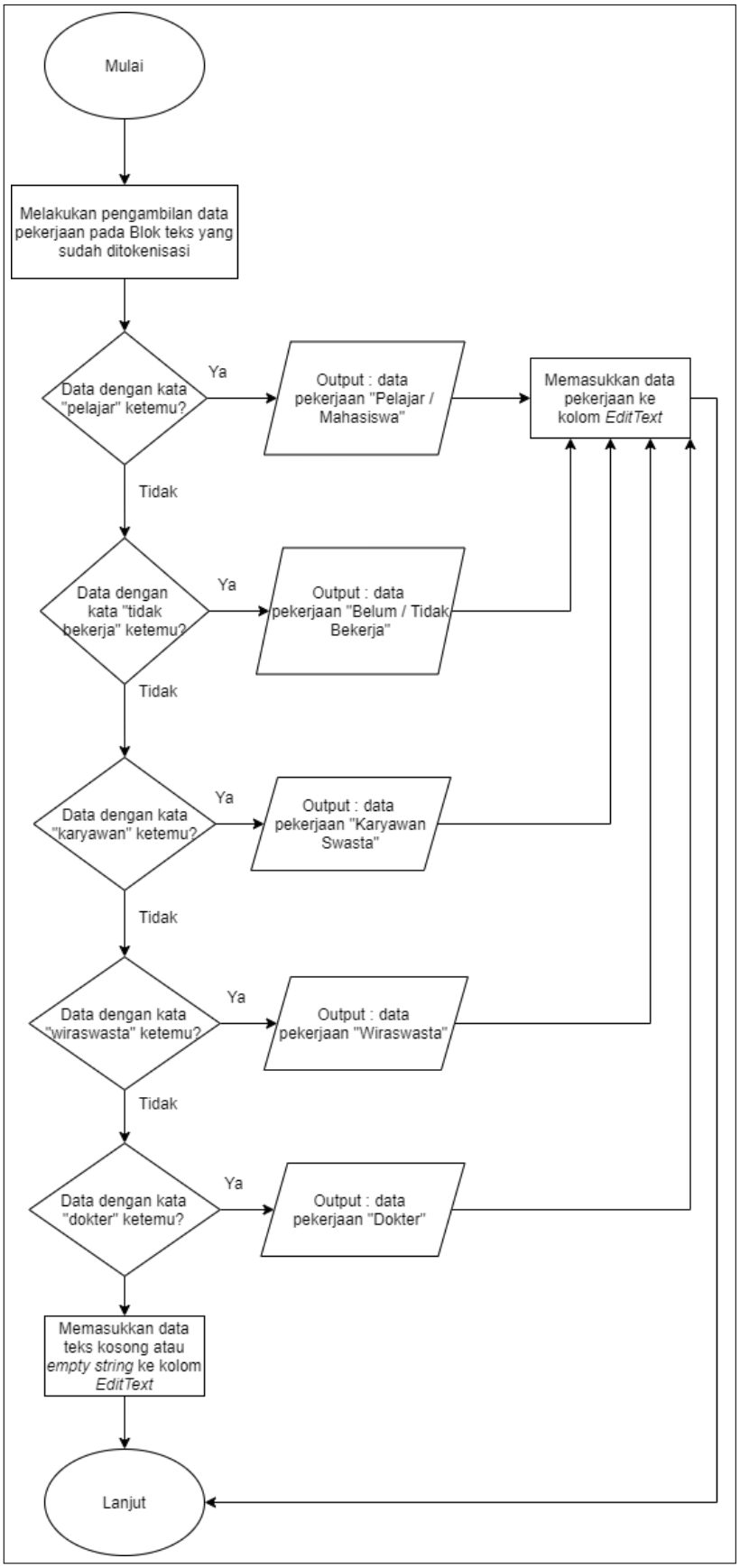
Gambar 3.14. Flowchart cek data alamat dengan hasil teks tokenisasi

Pada *flowchart* pengecekan agama tamu, terdapat pencarian kata *output* untuk data agama, tetapi hanya bisa satu kata saja. Kata yang dicari yaitu ”katholik”, ”kristen”, ”budha”, ”islam”, ”hindu”, ”khonghucu”. Jika salah satu kata muncul atau ketemu pada blok teks tokenisasi, maka kata itu akan menjadi data agama, dan kemudian dimasukkan ke kolom *EditText* agama . Namun, bila tidak ada salah satu kata yang muncul dari kata yang dicari, maka akan dimasukkan teks kosong atau *empty string* pada *EditText*. Pada Gambar 3.15, dapat dilihat alur proses pengecekan agama tamu.



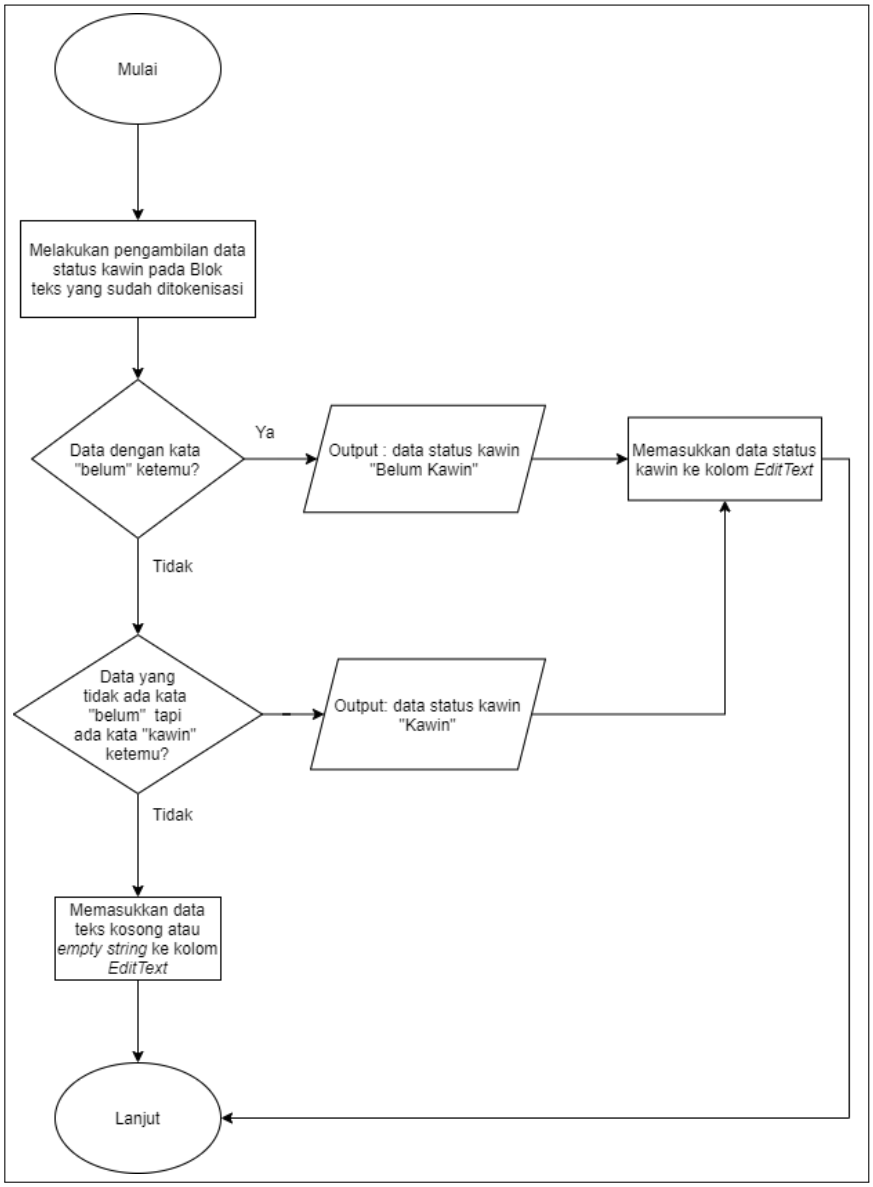
Gambar 3.15. Flowchart cek data agama dengan hasil teks tokenisasi

Berikutnya pada pengecekan pekerjaan, terdapat pilihan kata yang dapat dicari pada blok teks tokenisasi yaitu kata ”pelajar”, ”tidak bekerja”, ”karyawan”, ”wiraswasta”, dan ”dokter”. Bila salah satu kata ada atau ketemu dalam blok teks tokenisasi, maka kata itu akan menjadi data pekerjaan. Lalu dimasukkan ke dalam kolom *EditText* pekerjaan, tetapi apabila dari pilihan kata tersebut tidak ketemu, maka akan dimasukkan teks kosong ke dalam *EditText*. *Flowchart* pengecekan pekerjaan dapat dilihat pada Gambar 3.16.



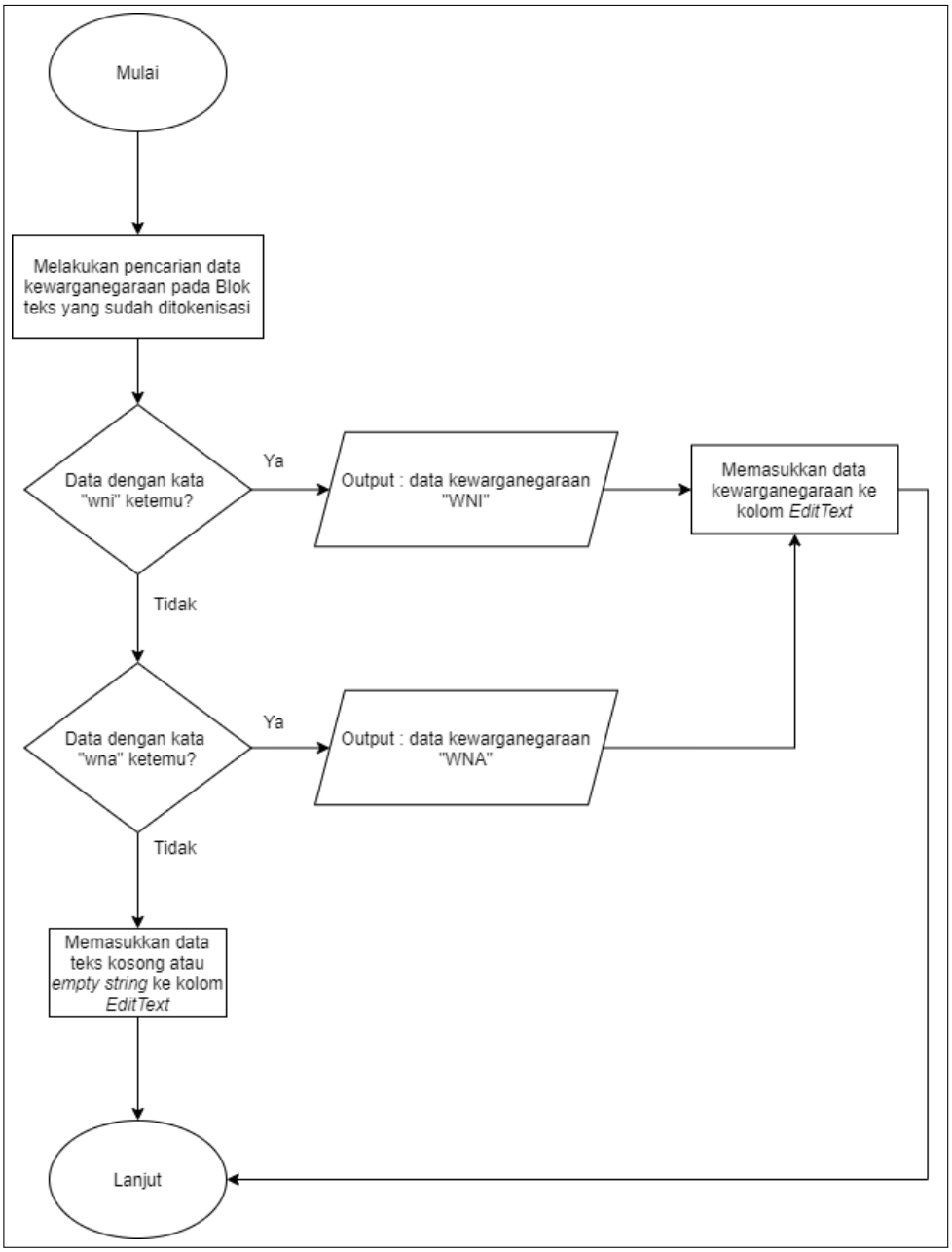
Gambar 3.16. Flowchart cek data pekerjaan dengan hasil teks tokenisasi

Untuk alur proses pengecekan status kawin tamu dengan cara mencari kata ”belum” atau mencari kata ”kawin” yang tidak ada kata ”belum” pada blok teks tokenisasi. Apabila kata ”belum” ketemu, maka kata ”Belum Kawin” akan dimasukkan ke dalam *EditText* status kawin, tetapi apabila kata ”belum” tidak ketemu, dan kata ”kawin” ketemu, maka kata ”Kawin” akan dimasukkan ke dalam *EditText* status kawin. Jika antara dua kata itu tidak ketemu, maka teks kosong akan dimasukkan ke dalam *EditText*. Pada Gambar 3.17, dapat dilihat *flowchart* cek data status kawin.



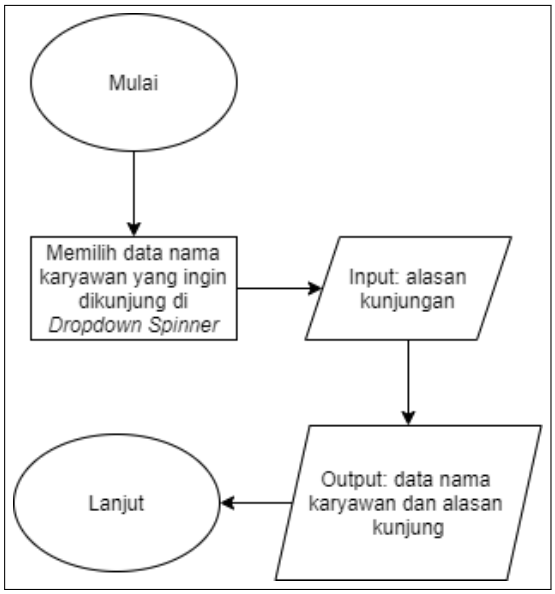
Gambar 3.17. Flowchart cek data status kawin dengan hasil teks tokenisasi

Pengecekan data kewarganegaraan yaitu dengan cara mencari kata ”wni” atau kata ”wna” pada blok teks tokenisasi. Jika kata ”wni” ketemu, maka kata ”WNI” akan dimasukkan ke dalam *EditText* kewarganegaraan, tetapi apabila kata ”wna” ketemu, maka kata ”WNA” akan dimasukkan ke dalam *EditText* kewarganegaraan. Apabila salah satu dari kedua kata itu tidak ketemu, maka dimasukkan kata kosong atau *empty string* ke dalam *EditText*. *Flowchart* untuk cek kewarganegaraan dapat dilihat pada Gambar 3.18.



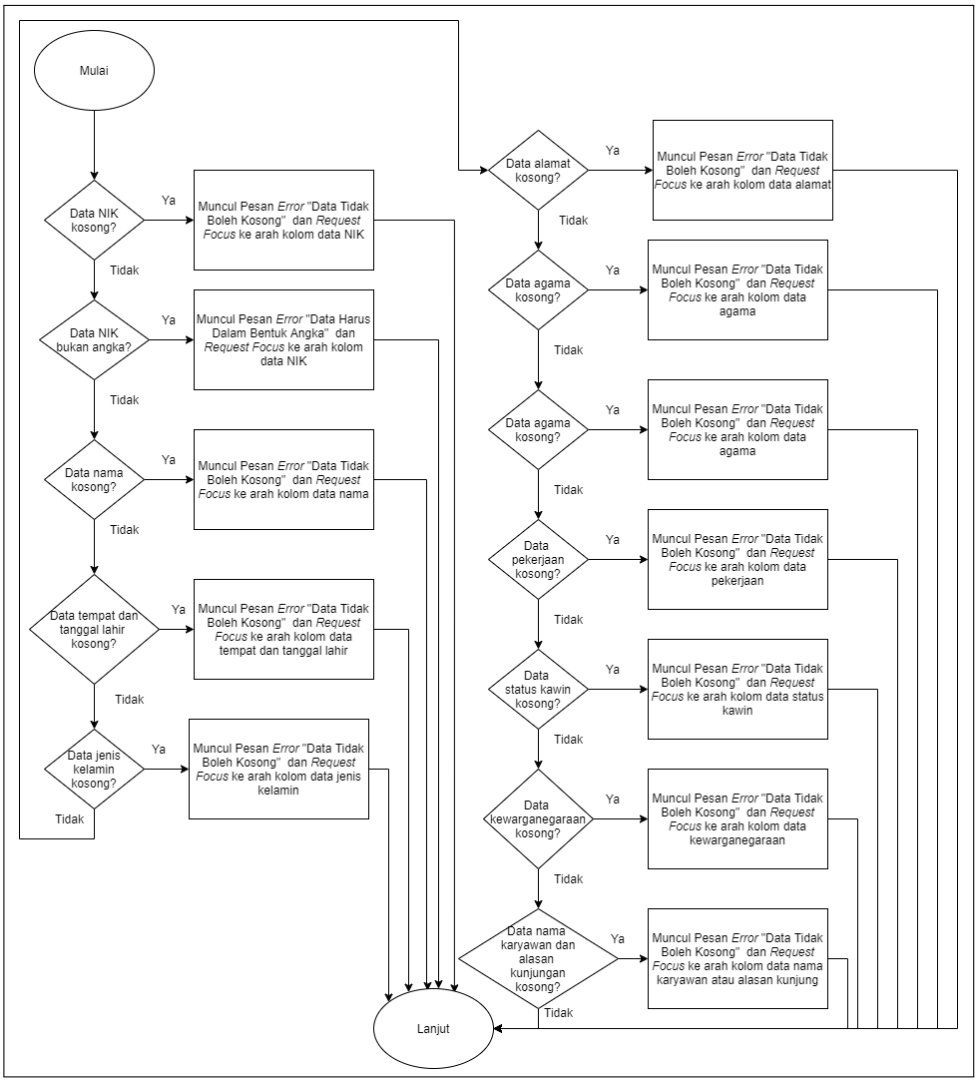
Gambar 3.18. Flowchart cek data Kewarganegaraan dengan hasil teks tokenisasi

Setelah semua data profil tamu sudah dimasukkan ke dalam *EditText*, selanjutnya memilih nama karyawan yang dapat diambil dari *Dropdown Spinner*, dan mengisi data alasan kunjungan. Alur proses dapat dilihat di Gambar 3.19.



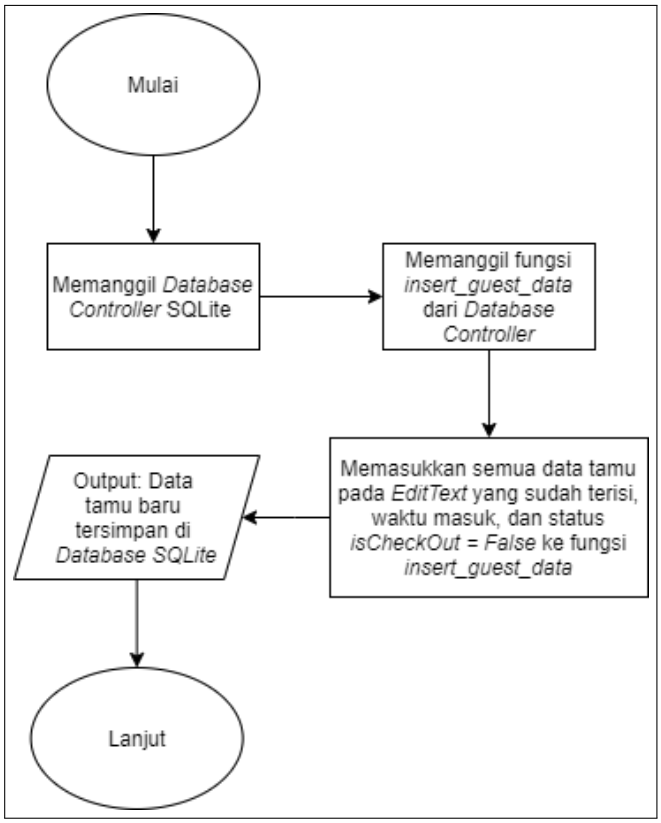
Gambar 3.19. Flowchart pengisian nama karyawan dan alasan kunjungan

Untuk validasi data tamu, proses validasi dilakukan pada setiap kolom *EditText*. Jika salah satu kolom *EditText* NIK, nama, tempat dan tanggal lahir, jenis kelamin, alamat, agama, pekerjaan, status kawin, kewarganegaraan ada yang kosong atau *empty string*, maka akan muncul pesan *error* ”Data Tidak Boleh Kosong” di bawah kolom *EditText* tersebut. Untuk NIK, bila datanya bukan angka, maka akan muncul pesan *error* yaitu ”Data Harus Dalam Bentuk Angka”. Setelah pesan *error* muncul, lalu akan melakukan *Request Focus* atau diarahkan ke kolom *EditText* yang muncul pesan *error*. *Flowchart* validasi dapat dilihat pada Gambar 3.20.



Gambar 3.20. Flowchart validasi data tambah tamu

Setelah data selesai validasi, dan tidak muncul pesan *error*, proses selanjutnya yaitu penyimpanan ke database. *Flowchart* prosesnya dapat dilihat pada Gambar 3.20. Proses diawali dengan menginisialisasi *Database Controller* dari SQLite, kemudian memanggil fungsi *insertguestdata*, dan memasukkan semua data dari *EditText*, waktu masuk, dan status `is_Checkout` sama dengan False ke dalam fungsi *insertguestdata*. Dengan fungsi ini, semua data untuk tamu baru tersimpan di dalam database, dan proses tambah tamu baru selesai.

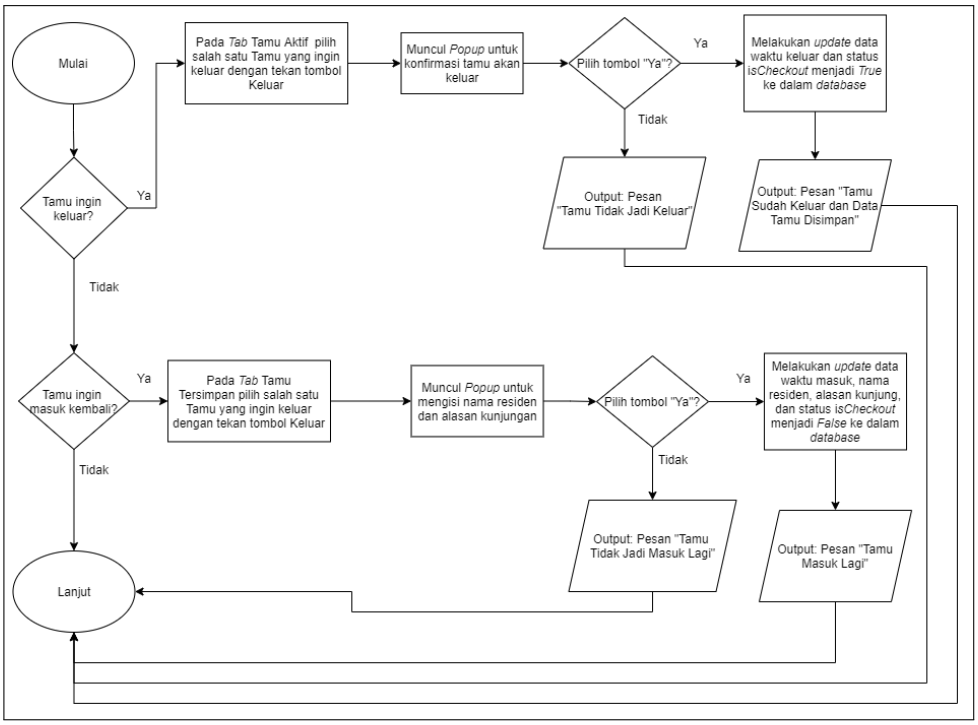


Gambar 3.21. Flowchart penyimpanan tamu baru ke database

C Flowchart Status Tamu Aktif Atau Tersimpan

Tamu yang baru dibuat akan menjadi tamu aktif atau *isCheckout* dipasang *False*. Proses perubahan status tamu dari aktif menjadi tersimpan, yaitu dengan menekan tombol Keluar pada salah satu data tamu di daftar tamu pada *tab* atau halaman Tamu Aktif. Setelah tombol Keluar ditekan, maka muncul *popup* untuk mengkonfirmasi tamu keluar. Apabila di *popup*, tombol Ya ditekan, proses yang terjadi adalah *update* data waktu keluar, dan *isCheckout* menjadi True untuk data tamu di database. Selanjutnya akan muncul pesan ”Tamu Sudah Keluar dan Data Tamu Disimpan”, dan data tamu akan berpindah ke daftar tamu di *tab* Tamu Tersimpan. Apabila tamu ingin masuk kembali, maka bisa dengan menekan tombol Masuk di *tab* Tamu Tersimpan. Setelah menekan tombol Masuk, akan muncul *popup* yang meminta untuk memilih nama karyawan, dan mengisi alasan kunjungan. Kemudian menekan tombol Ya, lalu dilakukan *update* data waktu masuk, dan *isCheck-*

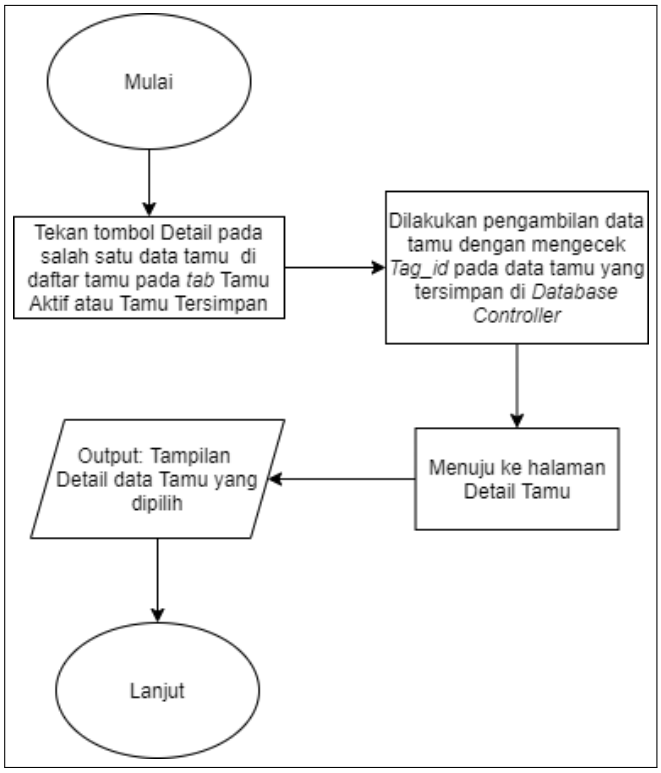
out menjadi False ke dalam data tamu di database. Kemudian akan muncul pesan ”Tamu Masuk Kembali”, dan data tamu berpindah ke daftar tamu di *tab* Tamu Aktif. *Flowchart* perubahan status tamu dapat dilihat pada Gambar 3.22.



Gambar 3.22. Flowchart perubahan status tamu aktif dan tersimpan

D Flowchart Detail Tamu

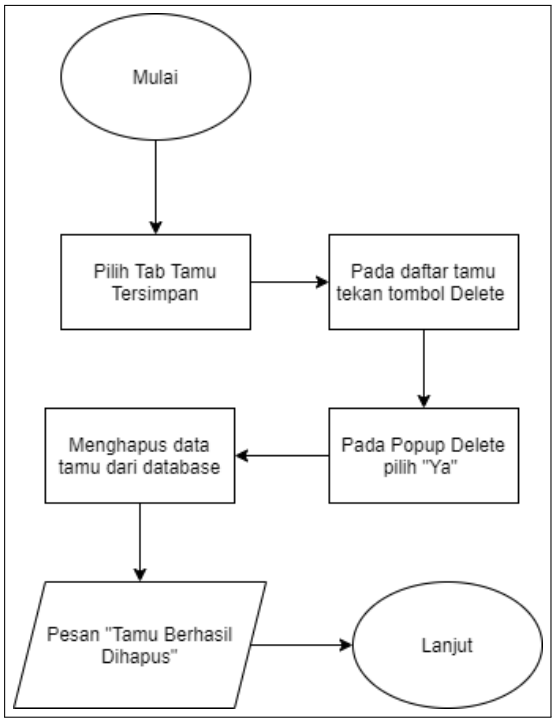
Flowchart detail tamu diawali dengan menekan tombol Detail pada daftar tamu, baik itu *tab* Tamu Aktif atau Tamu Tersimpan. Setelah menekan tombol Detail, akan melakukan proses pengambilan Detail Tamu sendiri yaitu dengan mengecek data *Tag-id* dari salah satu tamu apakah ada di database. Lalu bila ada, semua data profil tamu diambil berdasarkan data *Tag-id*. Kemudian menuju ke halaman Detail Tamu. Alur proses detail tamu ada pada Gambar 3.23.



Gambar 3.23. Flowchart detail tamu

E Flowchart Hapus Tamu

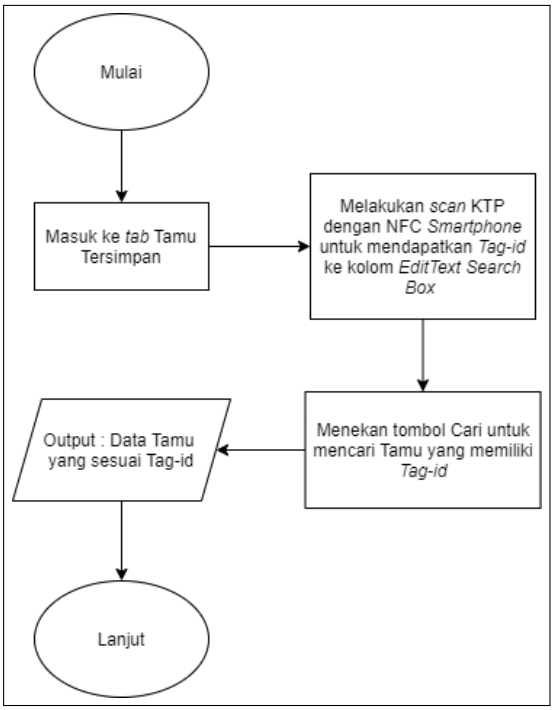
Flowchart hapus data tamu pada Gambar 3.24 ini dimulai dengan menekan tombol Hapus pada daftar tamu di *tab* Tamu Tersimpan. Selanjutnya akan muncul *popup* untuk mengkonfirmasi apakah data tamu ingin dihapus. Jika dipilih atau ditekan tombol Ya, maka data tamu yang dipilih akan dihapus dari database. Kemudian akan muncul pesan ”Tamu Berhasil Dihapus.”



Gambar 3.24. Flowchart hapus tamu

F Flowchart Cari Tamu

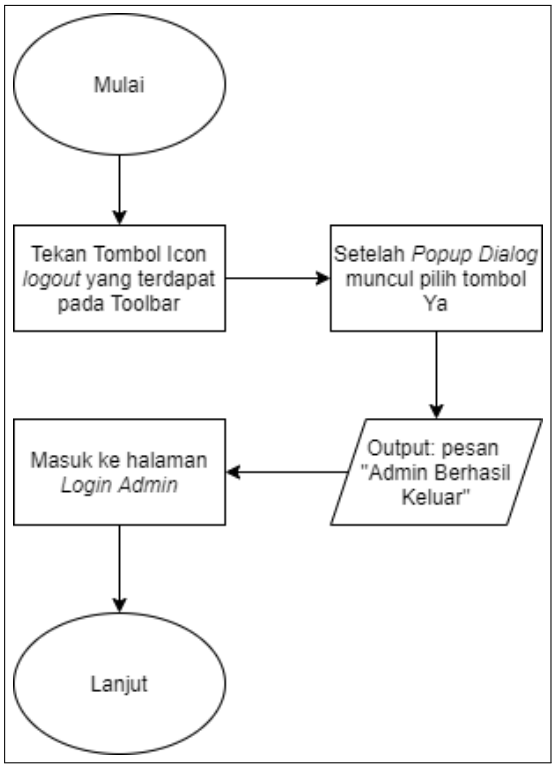
Proses cari tamu diawali dengan masuk ke *tab* Tamu Tersimpan, kemudian melakukan *scan* NFC KTP-el untuk mendapatkan *Tag-id*. Setelah berhasil mendapatkan *Tag-id*, kemudian menekan tombol Cari. Proses akan dilanjutkan ke pencarian data tamu berdasarkan *Tag-id* di dalam database. Apabila data tamu ketemu, maka akan menampilkan data tamu tersebut di daftar tamu. Proses pencarian tamu dapat dilihat pada Gambar 3.25.



Gambar 3.25. Flowchart cari tamu

G Flowchart Admin Logout

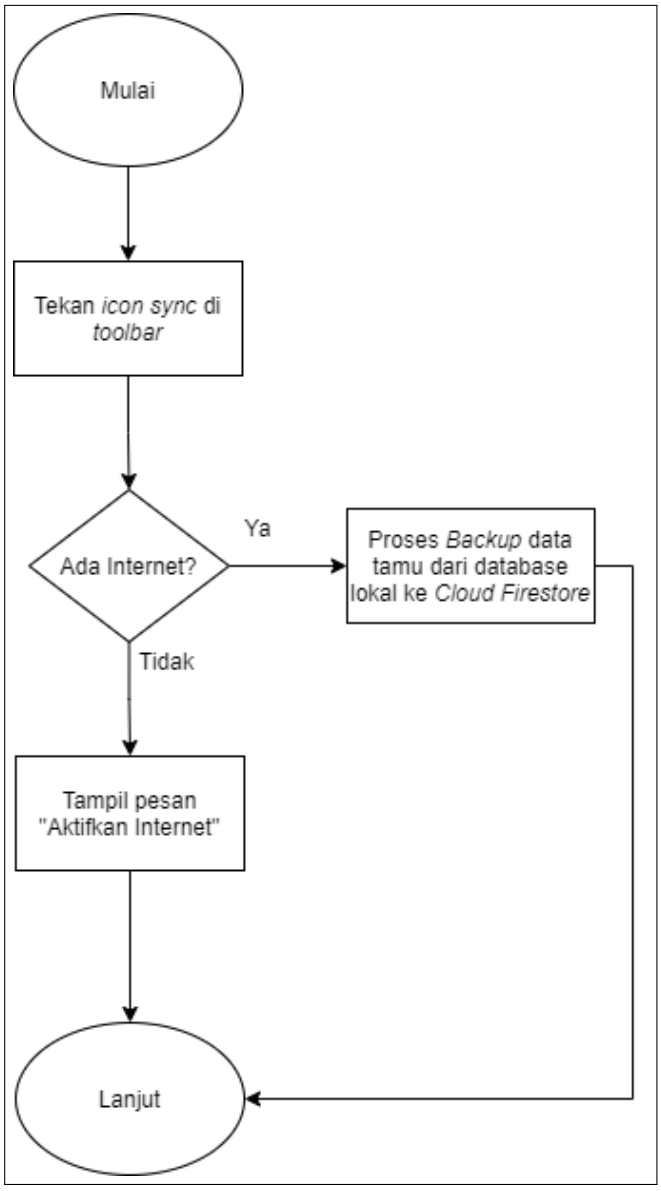
Alur proses admin melakukan *logout* atau keluar yaitu dengan menekan icon *logout* yang terletak di *toolbar*. Setelah ditekan, akan muncul *popup* untuk mengkonfirmasi apakah admin ingin *logout*. Apabila tombol Ya ditekan, maka muncul pesan ”Admin Berhasil Keluar”, dan kembali ke halaman *login*. *Flowchart* admin logout dapat dilihat pada Gambar 3.26.



Gambar 3.26. Flowchart admin *logout*

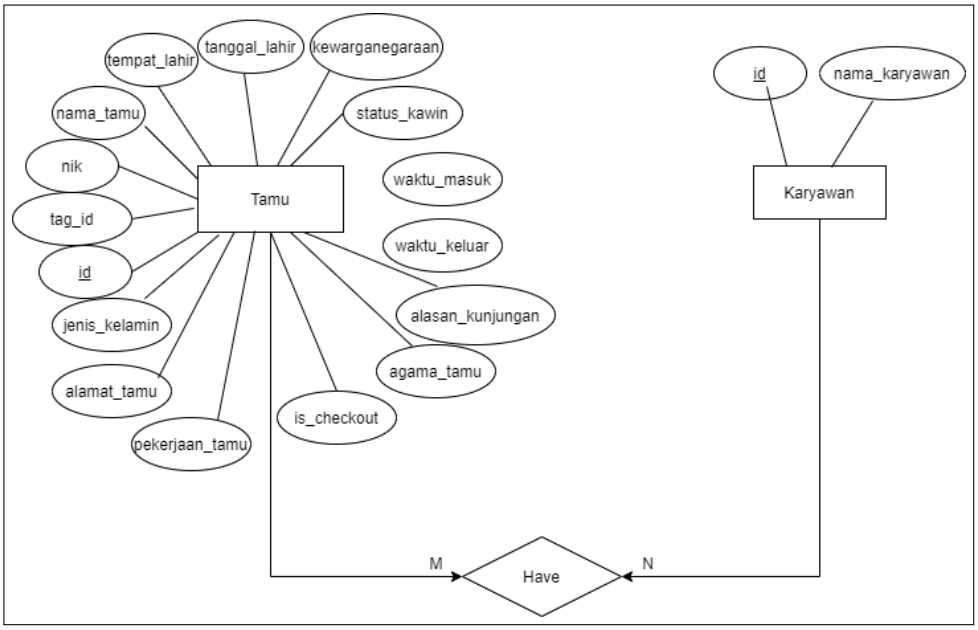
H Flowchart Backup Data Tamu ke Cloud Firestore

Alur proses dalam melakukan *backup* data tamu ke *Firestore* yaitu dimulai dengan menekan *icon sync* di *toolbar*, kemudian sistem akan mengecek apakah terkoneksi ke internet. Jika terkoneksi dengan internet, maka akan menjalankan proses *backup* data tamu yang tersimpan di database lokal ke *Cloud Firestore*, tetapi jika tidak terkoneksi, maka akan menampilkan pesan ”Aktifkan Internet”. Tampilan *flowchart* terkait dapat dilihat pada Gambar 3.27.



Gambar 3.27. Flowchart *backup* data tamu ke *Cloud Firestore*

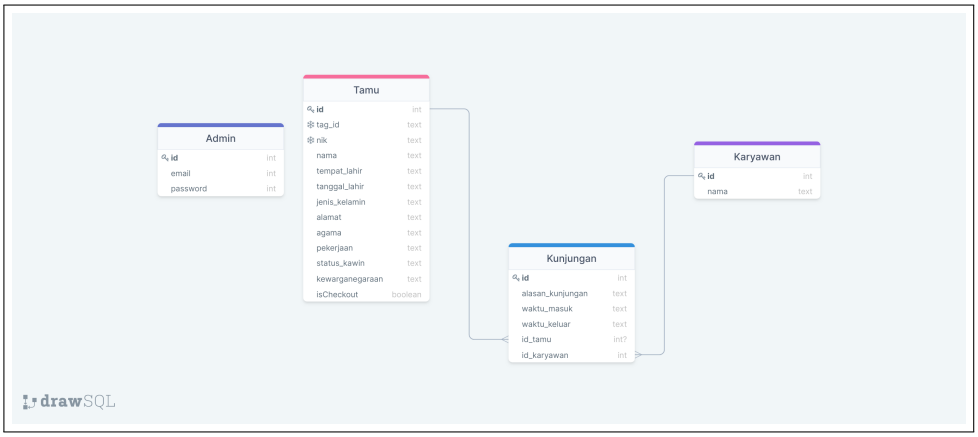
3.3.3 Entity Relationship Diagram



Gambar 3.28. *Entity Relationship Diagram* aplikasi

Pada Gambar 3.28 ini merupakan *Entity Relationship Diagram* untuk aplikasi buku tamu. Dapat dilihat hubungan *many to many* yang terjadi antara dua entitas yaitu entitas Tamu, dan entitas Karyawan. Entitas tamu sendiri memiliki beberapa atribut yang diperlukan di dalam aplikasi yaitu *id*, *Tag-id*, nama,tempat lahir, tanggal lahir, jenis kelamin, alamat, agama, pekerjaan, status kawin, kewargenargaan tamu, nama karyawan yang dikunjung, alasan kunjungan, waktu masuk, waktu keluar dan `is_Checkout` yang digunakan sebagai status tamu. Sedangkan pada entitas Karyawan terdapat atribut *id*, dan nama. Hubungan ini akan terjadi dalam aplikasi ketika proses penambahan tamu baru dilakukan.

3.3.4 Database Schema



Gambar 3.29. *Database Schema* aplikasi

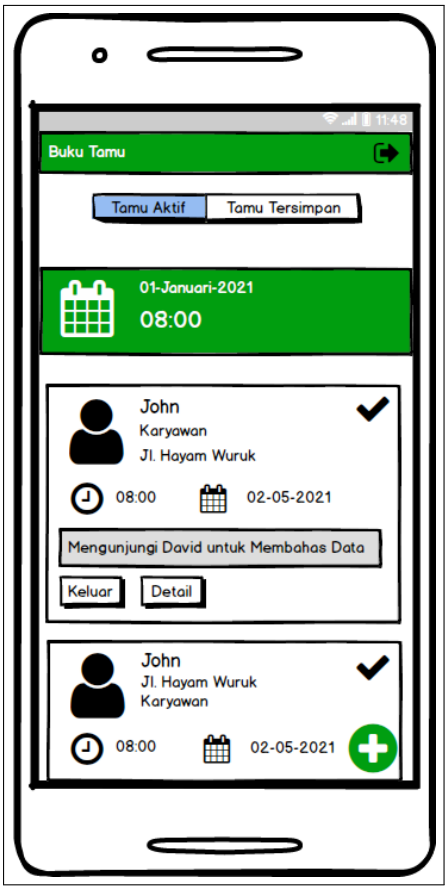
Pada Gambar 3.29 dapat dilihat bentuk dari *Database Schema* yang menggambarkan tabel atau entity yang digunakan saat pengerjaan aplikasi. Terdapat tabel admin, tamu, karyawan, dan juga tabel Kunjungan di mana terdapat hubungan antara tamu, dan karyawan.

3.3.5 Rancangan Antar Muka

Gambar 3.30. Rancangan antar muka halaman Admin *Login*

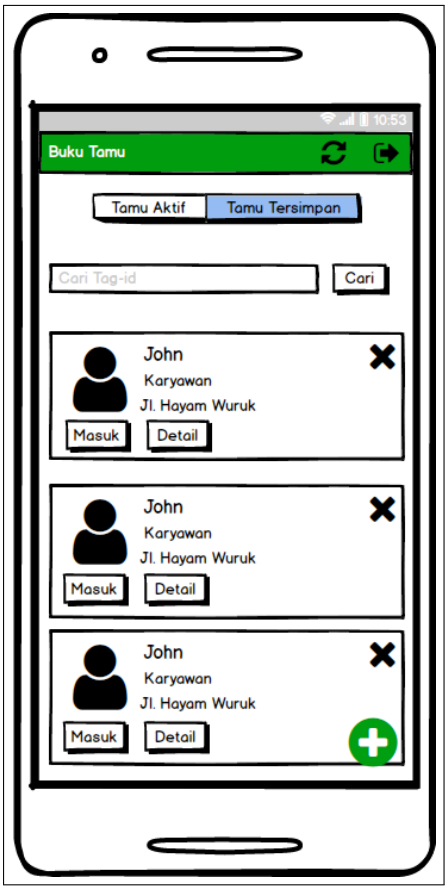
Pada Gambar 3.30, terdapat halaman admin *login* yang isinya *input field* untuk *email* dan *password*. Apabila *email* dan *password* sudah diisi, kemudian menekan tombol Masuk, maka admin berhasil masuk.

Rancangan antar muka pada halaman utama aplikasi terbagi menjadi dua yaitu *tab* Tamu Aktif, dan *tab* Tamu Tersimpan. Pada *tab* Tamu Aktif terdapat tanggal dan waktu sekarang, dan terdapat daftar tamu aktif yang dapat dilihat dalam bentuk *card*. Setiap *card* terdapat informasi data tamu yaitu nama, alamat, pekerjaan, jam dan tanggal waktu masuk, nama karyawan yang dikunjung, dan juga alasan kunjungan. Selain itu terdapat tombol Keluar, dan Detail. Tampilan rancangan antar muka dapat dilihat pada Gambar 3.30.



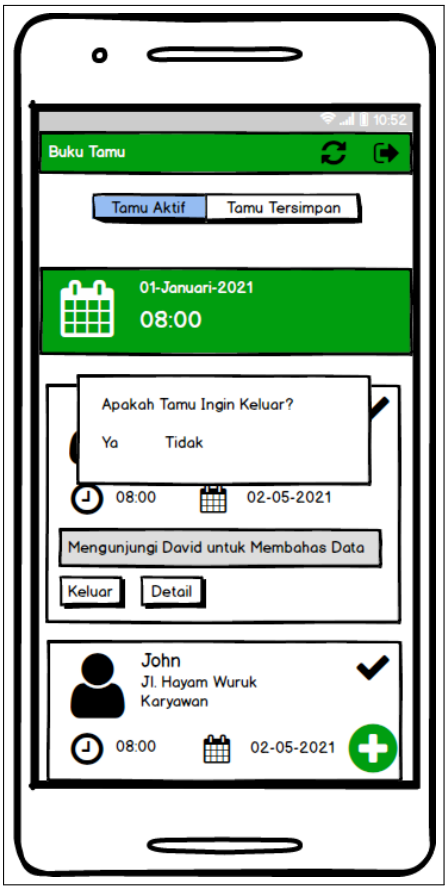
Gambar 3.31. Rancangan antar muka halaman utama pada *tab* Tamu Aktif

Rancangan antar muka *tab* Tamu Tersimpan dapat dilihat pada Gambar 3.32. Terdapat bagian *field EditText* untuk mencari *Tag-id* dengan NFC, dan tombol Cari. Kemudian dibawahnya merupakan daftar tamu yang tersimpan yang hanya menampilkan nama, pekerjaan, dan alamat tamu. Di daftar tersebut ditampilkan juga tombol Masuk dan Detail.



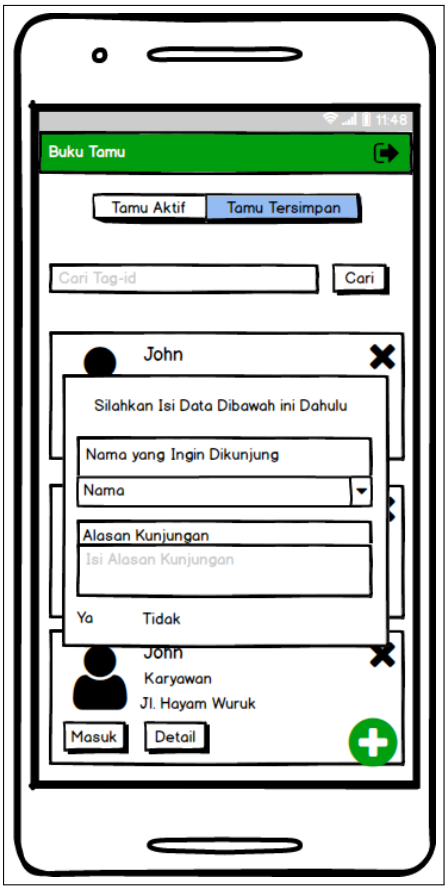
Gambar 3.32. Rancangan antar muka halaman utama pada *tab* Tamu Tersimpan

Pada daftar tamu di *tab* Tamu Aktif, apabila tombol Keluar ditekan, maka muncul *popup* konfirmasi untuk tamu yang akan keluar seperti Gambar 3.33.



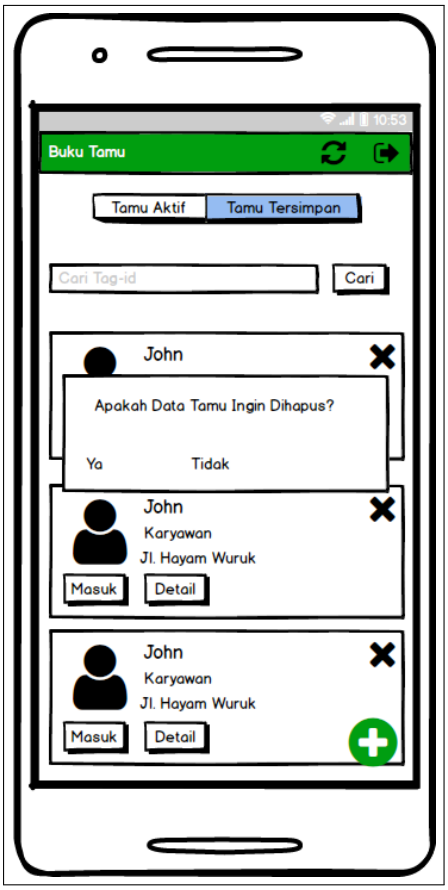
Gambar 3.33. Rancangan antar muka *popup* konfirmasi tamu keluar

Sedangkan di *tab* Tamu Tersimpan, pada daftar tamu apabila tombol Masuk ditekan akan muncul *popup* konfirmasi yang menyuruh untuk mengisi nama yang ingin dikunjung, dan alasan kunjungan. Rancangan antar muka *popup* tersebut dapat dilihat pada Gambar 3.34.



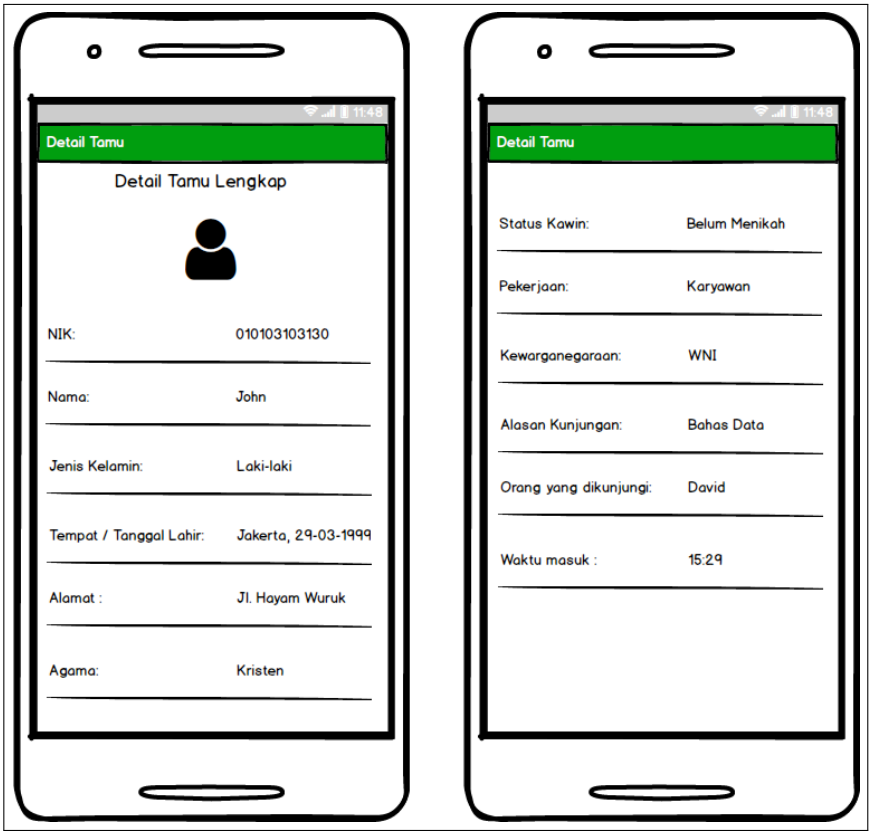
Gambar 3.34. Rancangan antar muka *popup* konfirmasi tamu masuk

Selain itu pada *tab* Tamu Tersimpan, terdapat *icon* silang. Apabila ditekan akan muncul *popup* konfirmasi untuk menghapus data tamu. Rancangannya dapat dilihat pada Gambar 3.35.



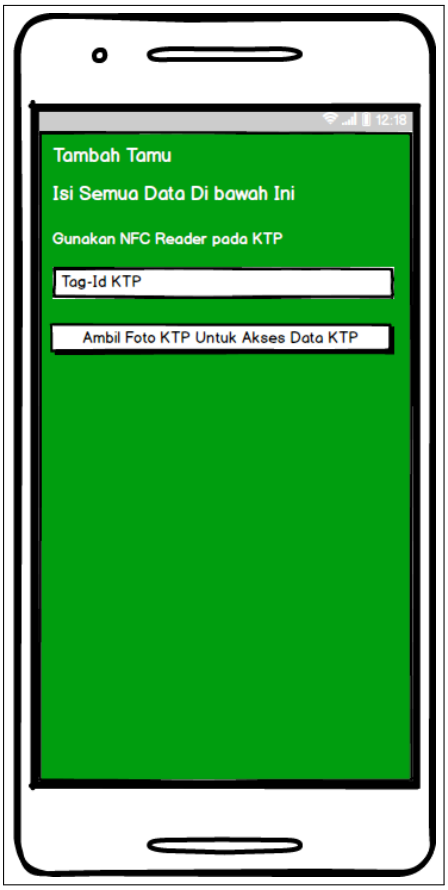
Gambar 3.35. Rancangan antar muka *popup* konfirmasi hapus data tamu

Pada Gambar 3.36 merupakan rancangan antar muka halaman Detail Tamu yang berisi *field* data NIK, nama, tempat dan tanggal lahir, jenis kelamin, alamat, agama, status perkawinan, pekerjaan, kewarganegaraan tamu, orang yang ingin dikunjungi, dan alasan kunjungan tamu secara lengkap.



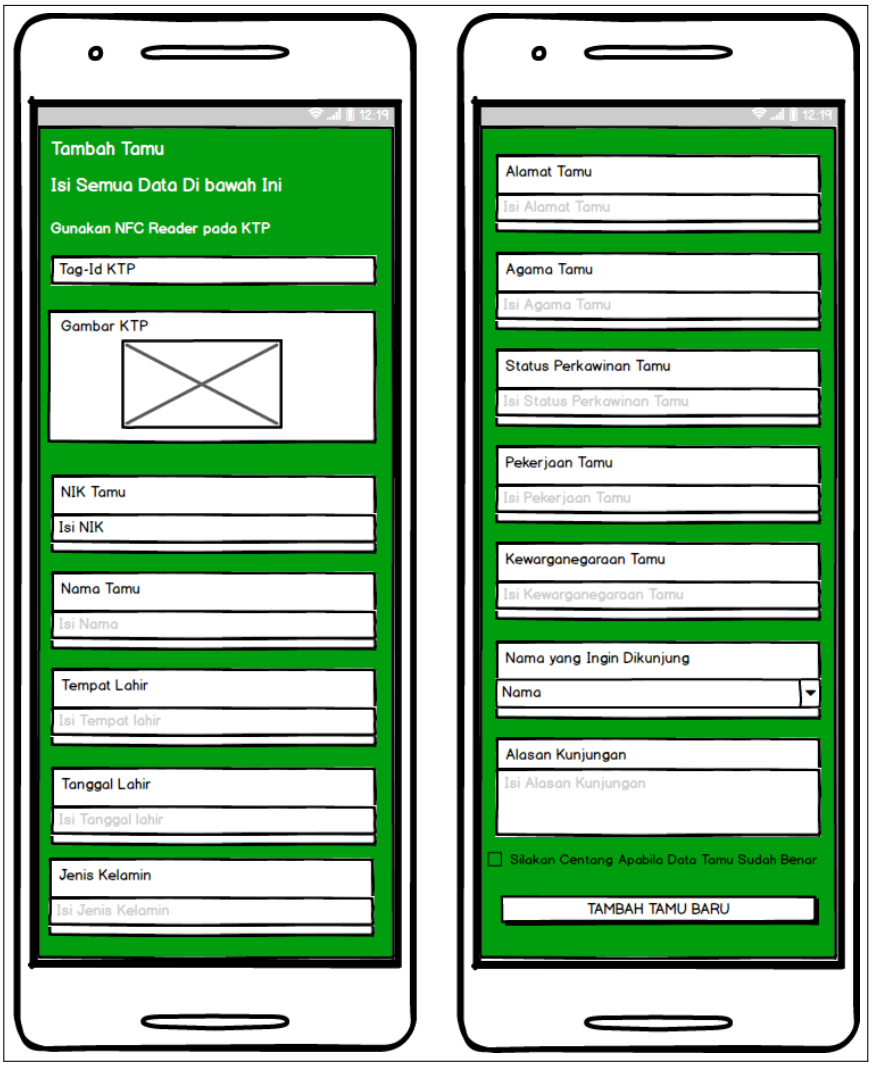
Gambar 3.36. Rancangan antar muka halaman Detail Tamu

Untuk rancangan antar muka Tambah Tamu Baru, terdapat *field EditText* untuk memasukkan *Tag-id* KTP yang datanya didapatkan dengan melakukan *scan* KTP-el dengan NFC *Smartphone*, dan tombol Ambil Foto KTP-el. Rancangannya dapat dilihat pada Gambar 3.37.



Gambar 3.37. Rancangan antar muka halaman Tambah Tamu baru

Kemudian apabila foto KTP-el sudah diambil, maka muncul beberapa *field* yaitu bagian foto, *field EditText* NIK tamu, nama tamu, tempat dan tanggal lahir tamu, jenis kelamin tamu , alamat tamu, agama tamu, status perkawinan tamu, pekerjaan tamu, kewarganegaraan tamu, *field Dropdown Spinner* untuk nama yang ingin dikunjung, dan *field* untuk alasan kunjungan. Selain itu terdapat *checkbox* untuk memastikan data di setiap *field EditText* sudah terisi dengan benar, dan tombol Tambah Tamu Baru. Rancangan antar muka halaman tambah tamu setelah ambil foto KTP-el dapat dilihat pada Gambar 3.38.



Gambar 3.38. Rancangan antar muka halaman Tambah Tamu baru setelah ambil foto KTP-el