BAB III
PERANCANGAN DAN IMPLEMENTASI SISTEM

Pada bab ini akan membahas bagaimana perancangan dari sistem yang dibuat sehingga dapat mencari bentuk terbaik berdasarkan permasalahan yang mungkin terjadi dan kebutuhan dari sistem ini nantinya. Selain itu akan diterangkan juga bagaimana sistem ini akan berjalan.

3.1 Deskripsi Sistem Secara Umum

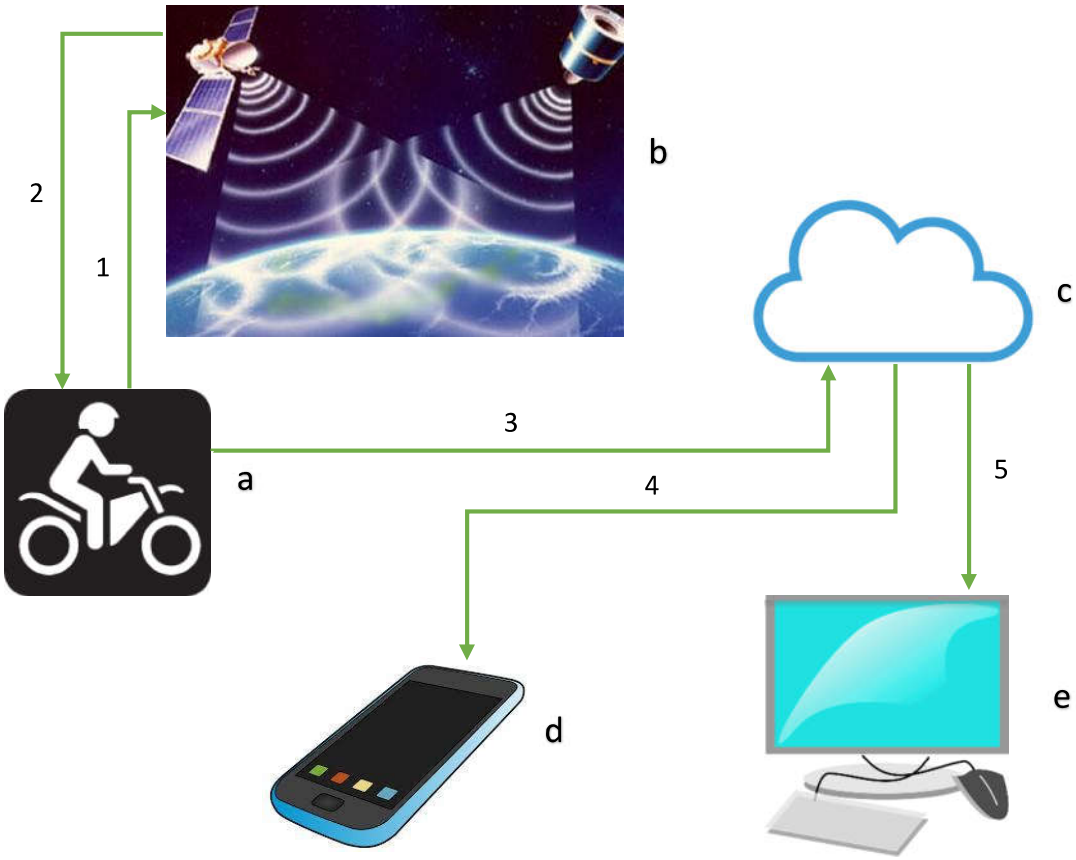
Sistem ini merupakan aplikasi *mobile* yang berjalan pada sistem operasi *Android,* yang memiliki fungsi untuk mengirimkan koordinat dan lokasi ke *Smartphone* pemilik sepeda motor (*receiver*) ketika *Smartphone* pada sepeda motor (*transmitter*) mendeteksi telah melewati radius ± 10 meter. Aplikasi juga mengirimkan koordinat, lokasi, dan identitas pemilik sepeda motor tersebut ke *database* pada halaman *web*.

3.2 Perancangan Sistem

Dalam sub bab ini akan dibahas bagaimana alur dari sistem menggunakan blok sistem beserta DFD dan ERD nya dan perancangan menu beserta fitur yang akan terdapat pada aplikasi.

3.2.1 Perancangan Alur Sistem

Alur pada perancangan sistem ini menggunakan DFD dan ERD. Secara garis besar alur sistem dijelaskan pada gambar 3.1



Gambar 3.1 Blok Sistem Aplikasi Keamanan Sepeda Motor

Keterangan blok sistem aplikasi keamanan sepeda motor :

a. *Smartphone* pada sepeda motor : Mengirimkan notifikasi dan lokasi terkini ke *smartphone* pada pemilik sepeda motor (*receiver*) dan *database* kepolisian.
b. Satelit : Mengirimkan koordinat berupa lokasi yang diminta oleh *smartphone* pada sepeda motor (*transmitter*) yang melewati jarak batas.
c. *Cloud* : Menyimpan data yang telah disimpan pada aplikasi dan sebagai perantara atau pemberi data kepada *receiver* dan *database*.
d. *Smartphone* pada pemilik sepeda motor : Mendapat notifikasi dan menerima lokasi dalam bentuk *Google Maps* dari *receiver*.
e. Personal Komputer : Menerima lokasi dan data pemilik sepeda motor yang dicuri dalam bentuk *database* secara *online*.

Berikut penjelasan dari blok sistem di atas :

1. *Transmitter* akan meminta kepada satelit koordinat dan lokasi di mana sepeda motor berada.
2. Setelah bagian pertama disetujui, maka akan dikirimkan lokasi ke *receiver* untuk menandai jarak awal sepeda motor berada.
3. Ketika *transmitter login* ke aplikasi, maka akan langsung tersimpan di *cloud* data dari *transmitter.* Selanjutnya akan mengirimkan lokasi, koordinat, dan data pemilik yang tersimpan di *cloud* pada saat registrasi kepada *receiver* dan *database* ketika kondisi *transmitter* melewati radius ± 10 meter.
4. Ketika kondisi pada poin 3 terjadi, *transmitter* akan mengirimkan notifikasi dan lokasi sepeda motor setiap 30 detik ke *receiver.*
5. Tidak hanya ke *receiver,* aplikasi juga mengirimkan lokasi beserta identitas pemilik sepeda motor ke *database*.

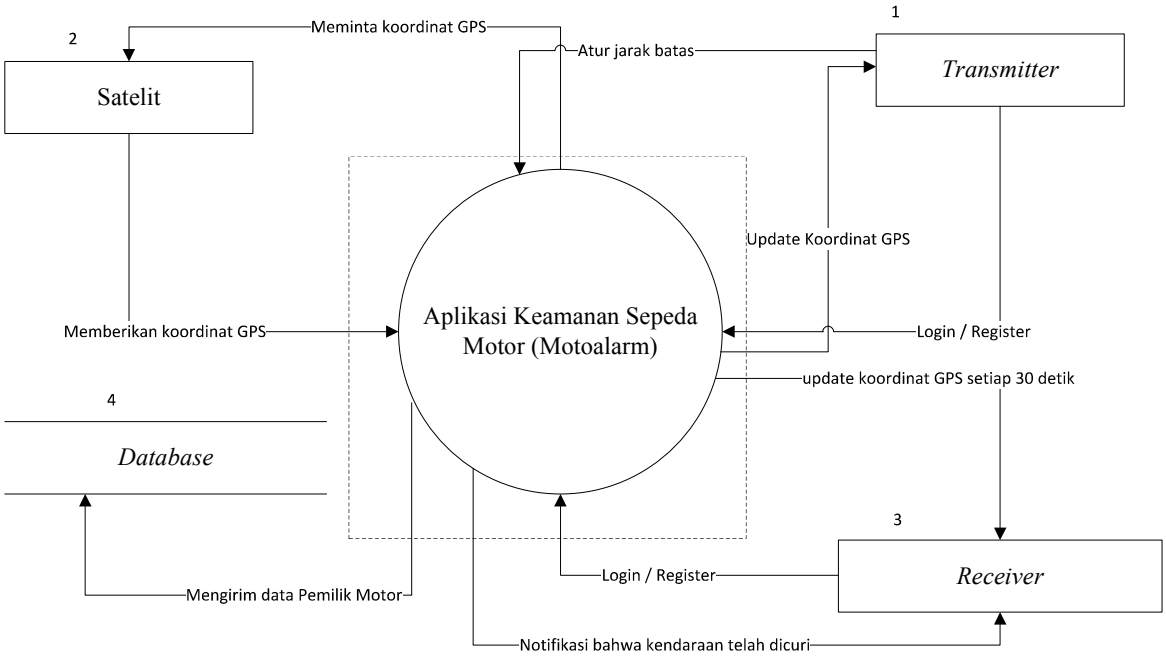
3.2.2 Perancangan Aplikasi

3.2.2.1 Data Context Diagram (DCD)

Seperti dijelaskan pada sub bab 3.1 bahwa Aplikasi *MotoAlarm* ini akan memiliki fungsi :

1. Mengirimkan notifikasi bahwa kendaraan telah melewati radius yang ditentukan bahwa sepeda motor terindikasi dicuri.
2. Jika sepeda motor dicuri, maka *transmitter* akan mengirimkan ke *receiver* lokasi dari sepeda motor yang dicuri setiap 30 detik.
3. *Transmitter* juga mengirimkan koordinat dan lokasi serta identitas pemilik sepeda motor ke *database.*

Berikut alur dari sistem yang digambarkan dalam DCD seperti pada gambar 3.2 :

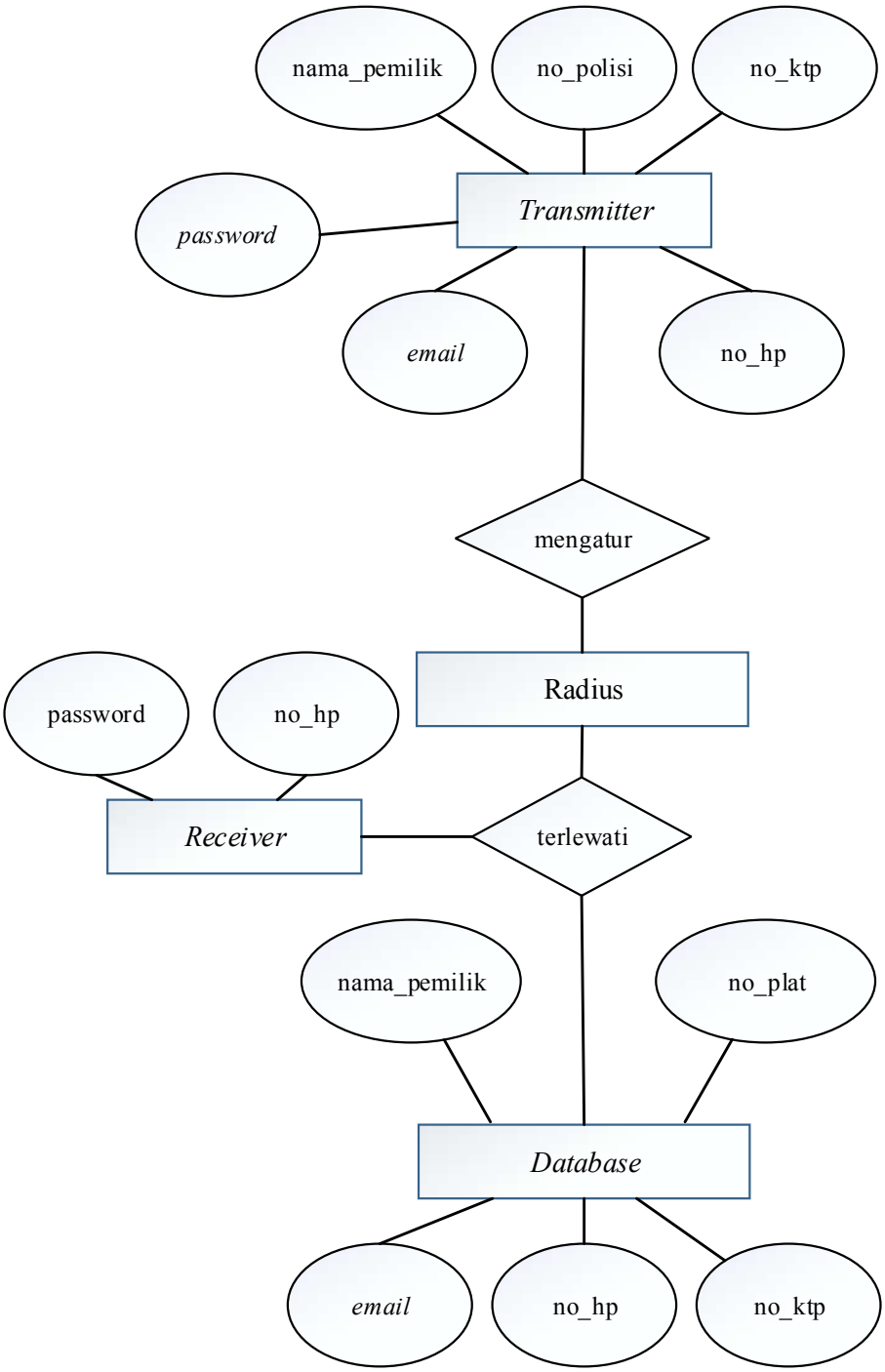


Gambar 3.2 Data Context Diagram Aplikasi Motoalarm

Berikut penjelasan dari DCD pada gambar 3.2 :

1. Pada poin pertama, *transmitter* harus *register* ke aplikasi sehingga dapat *login* dan menggunakan aplikasi. Ketika aplikasi mendeteksi bahwa radius yang ditentukan terlewati, maka *transmitter* meminta kepada aplikasi lokasi *transmitter* berada.
2. Setelah adanya permintaan dari *transmitter,* maka aplikasi meminta satelit untuk mengirimkan lokasi dari *transmitter* setiap 30 detik.
3. *Receiver* sebelumnya juga harus *register* ke aplikasi dan *login* untuk menggunakan aplikasi ini. *Receiver* akan menerima pemberitahuan setiap 30 detik dari lokasi *transmitter* ketika dicuri.
4. *Database* akan menyimpan koordinat, lokasi, dan data pemilik.

3.2.2.2 Entity Relationship Diagram (ERD)



Gambar 3.3 Entity Relation Diagram Aplikasi MotoAlarm

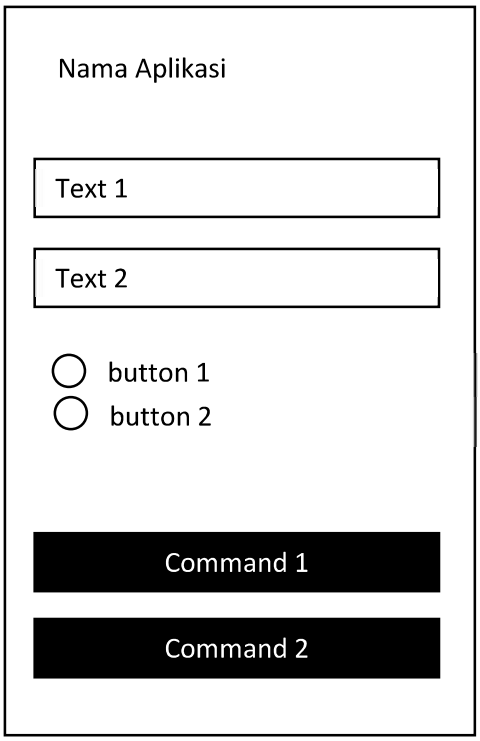
Berdasarkan ERD pada gambar 3.3, dapat dijelaskan bahwa t*ransmitter* memiliki data : nomor ID (KTP/SIM), nomor plat sepeda motor, nama pemilik sepeda motor, nomor telepon, *email*, dan *password*.

Kemudian *transmitter* mengatur radius. Jika terlewati, akan mengirimkan notifikasi dan mengirim lokasi sepeda motor setiap 30 detik ke *receiver* dan *database*. *Database* akan mendapatkan data dari *transmitter* yaitu nomor ID (KTP/SIM), nomor plat sepeda motor, nama pemilik sepeda motor, nomor telepon, email, dan lokasi.

3.3 Implementasi

3.3.1 Tampilan Aplikasi

a. *Form Login* Aplikasi



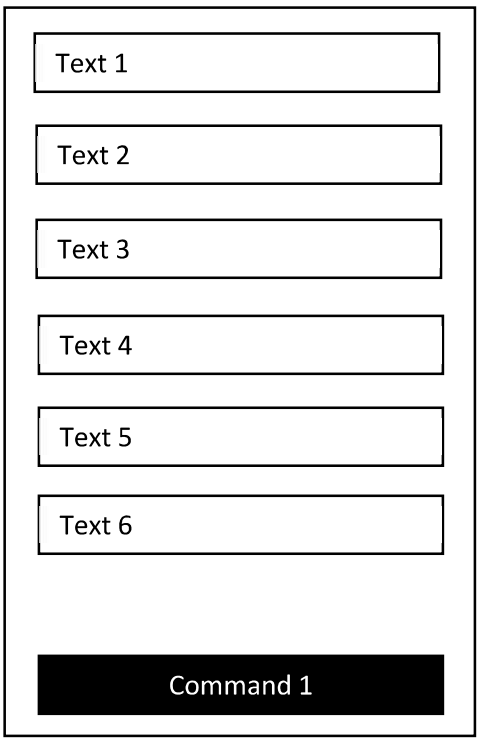
Gambar 3.4 Form Login

Dalam *Form Login* ini, *Text* 1 merupakan nomor telepon dan *Text* 2 merupakan *password*. *Button* merupakan tombol pilihan. *Button* 1 dan *button* 2 ditentukan sebagai *transmitter* dan *receiver*. *Command* 1 digunakan untuk masuk ke aplikasi (*sign in*). Sedangkan *command* 2 digunakan untuk masuk ke *form* registrasi aplikasi.

Ketika nomor telepon dan password diisi, namun belum mendaftar di *form* registrasi, maka ketika menekan *command* 1 akan muncul keterangan berupa nada dering, getaran, dan *push*

notification yang berisi "*nomor HP* atau *password* salah". Begitu juga ketika nomor telepon atau *password* tidak terisi, maka akan muncul keterangan berisi "harus diisi"

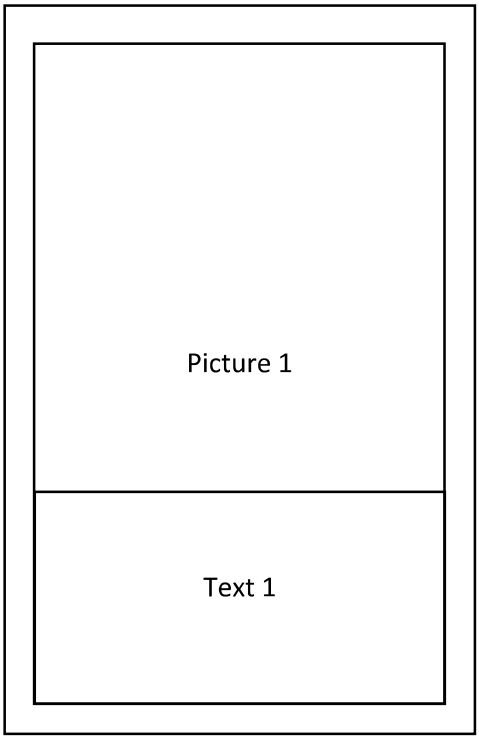
b. *Form* Registrasi Identitas Sepeda Motor



Gambar 3.5 Tampilan Form Registrasi Identitas Sepeda Motor

Ketika menekan command 2 pada gambar 3.4, maka akan langsung dibawa ke *form* registrasi seperti gambar 3.5. *Text* 1 hingga *Text* 6 merupakan inputan meliputi, nama pemilik sepeda motor, nomor kendaraan, nomor KTP/ SIM, nomor telepon, *email*, dan *password*. *Command* 1 merupakan tombol yang ketika ditekan akan menyimpan data-data yang diisi dan muncul *push notification* berisi "data berhasil disimpan"

c. *Form* ketika sepeda motor melewati jarak batas



Gambar 3.6 Tampilan ketika sepeda motor melewati jarak batas

Pada gambar 3.6, *Picture* 1 merupakan gambaran peta atau visual dari *Google Maps* dari lokasi *transmitter* ketika telah masuk ke aplikasi. Ketika sepeda motor melewati jarak batas, maka akan muncul *Text* 1 merupakan keterangan-keterangan tentang koordinat, alamat atau nama lokasi, pembaruan waktu dan jarak, dan identitas pemilik sepeda motor, seperti nomor polisi, nomor HP, nomor KTP, dan alamat *email*. Selain itu juga akan tertampil *push notification* berisi "Jarak batas terlewati, sepeda motor berpotensi dicuri" setiap 30 detik. Namun *push notification* hanya tampil pada saat masuk ke mode *receiver*.

3.3.2 Pengaturan Notifikasi pada Aplikasi

Pada saat pengujian pengiriman notifikasi, ada aturan pemberian notifikasi pada aplikasi yang akan dibuat. Radius yang telah ditentukan pada aplikasi adalah ± 10 meter. Jika melewati batas tersebut nantinya akan memunculkan notifikasi, yaitu berupa nada dering, getaran, maupun *push notification* pada *receiver* bahwa sepeda motor terindikasi dicuri seperti yang digambarkan pada tabel 3.1.

Tabel 3.1 Aturan pemberian notifikasi

Rentang Jarak (meter)	Notifikasi
0 – 10	Tidak
> 10	Ya

3.3.3 Skenario Pengujian

Dalam pengujian aplikasi, ada beberapa skenario yang akan dilakukan, yaitu :

1. Sepeda motor yang dijadikan obyek penelitian diasumsikan dipakai oleh pencuri ataupun akan dibawa di perjalanan (outdoor) sehingga tidak memperdulikan dimana posisi parkir (baik di *basement* atau lapangan terbuka/*outdoor*).
2. Seperti dalam batasan masalah pada Bab I, akses yang digunakan adalah *multiuser,* artinya aplikasi dapat dipakai oleh setiap orang jika memiliki aplikasi dan tahu *username* dan *password* yang digunakan.
3. Informasi yang dipakai hanya menggunakan akses paket data seluler.