

BAB III

METODE PENELITIAN

3.1 Alat dan Bahan

Alat dan bahan pendukung dalam pengerjaan aplikasi monitoring pertumbuhan dan perkembangan bayi berbasis android ini terdiri dari perangkat keras (*hardware*) dan perangkat lunak(*software*) .

3.1.1 Perangkat Lunak

Untuk mengembangkan aplikasi monitoring pertumbuhan dan perkembangan bayi, peneliti membutuhkan beberapa perangkat lunak yang mendukung dalam pengerjaan aplikasi monitoring pertumbuhan dan perkembangan bayi, yaitu:

Tabel 3.1 Perangkat Lunak

Software	Versi	Fungsi
Android Studio	1.3.2	Sebagai tempat untuk mengembangkan kode aplikasi dan pembuatan layout.
XAMPP	3.2.1	Sebagai server localhost yang akan membantu koneksi ke <i>Database MySQL</i>
Netbeans	8.0	Sebagai media untuk memanggil atau memasukkan data kedalam MySQL
Google Chrome	60.0.3112.101	Sebagai media untuk membuka <i>PHPMyAdmin MySQL</i>

3.1.2 Perangkat Keras

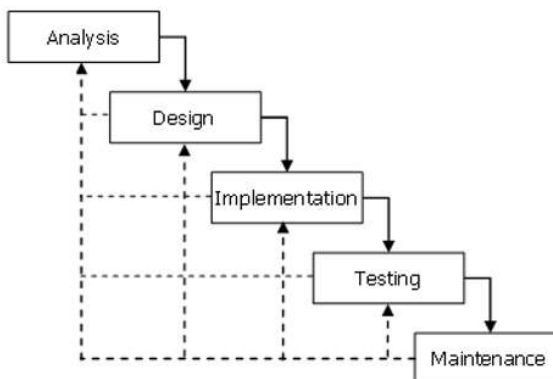
Untuk mengembangkan aplikasi monitoring pertumbuhan dan perkembangan bayi, peneliti membutuhkan beberapa perangkat keras yang mendukung dalam pengerjaan aplikasi monitoring pertumbuhan dan perkembangan bayi, yaitu:

Tabel 3.2 Perangkat Keras

Item	Spesifikasi
<i>Model</i>	Asus K46CB
<i>CPU Cores</i>	4 CPU x 1,7 Ghz
<i>Processor Type</i>	Intel(R) Core(TM) i5-3317U
<i>RAM</i>	8 Gb
<i>Memory</i>	500 Gb

3.2 Metode Pengembangan Sistem

Pengembangan aplikasi monitoring pertumbuhan dan perkembangan bayi menggunakan model SDLC (*Software Development Life Cycle*). Model SDLC yang dipakai dalam pengembangan sistem ini yaitu model *Waterfall* dari Youssef Bassil dalam jurnalnya yang berjudul “*A Simulation Model for the Waterfall Software Development Life Cycle*”. Tahap-tahap pengembangan sistem menggunakan model *Waterfall* dapat dilihat pada gambar 3.1.



Gambar 3.1 Model *Waterfall*

3.2.1 Analysis

Pada tahap ini peneliti melakukan penyebaran kuesioner terhadap calon *user* melalui *google form*. Hasil dari kuesioner digunakan untuk menganalisa data yang digunakan untuk pengembangan sistem dalam membantu memenuhi kebutuhan *user*.

3.2.2 Design

Setelah dianalisa maka peneliti membuat rancangan *interface* dan sistem berdasarkan kebutuhan fungsi *software*. Adapun rancangan *user interface* menggunakan *software Mockup* dan rancangan sistem menggunakan *flowchart* dan beberapa UML (*Unified Modelling Language*) seperti *usecase diagram* dan *class diagram*.

3.2.3 Implementation

Pada tahap ini peneliti mengubah *design* menjadi sebuah aplikasi agar fungsi *software* dapat dijalankan. Untuk mengubah desain menjadi sebuah aplikasi peneliti menggunakan *software Android Studio* dengan bahasa pemrogramana *java*. Pengembangan aplikasi ini dilakukan dari awal hingga aplikasi siap dijalankan.

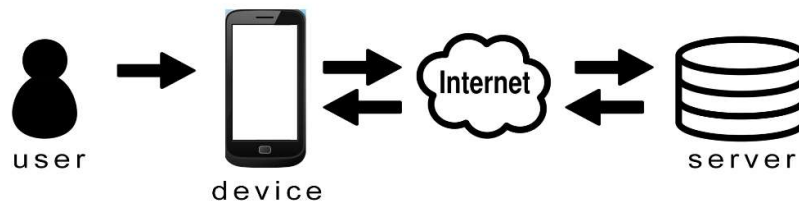
3.2.4 Testing

Tahap selanjutnya yaitu *testing*. *Testing* digunakan untuk mengetahui apakah aplikasi yang dikembangkan berjalan sesuai dengan yang diharapkan. *Testing* yang digunakan pada aplikasi *monitoring* pertumbuhan dan perkembangan bayi menggunakan metode *black box testing*.

3.2.5 Maintenance

Tahap terakhir yaitu *maintenance*. Tahap ini digunakan ketika aplikasi sudah jadi dan ingin lebih dikembangkan sekaligus pemeliharaan ketika terjadi *bug*.

3.3 Arsitektur Sistem



Gambar 3. 2 Arsitektur Sistem

Penjelasan gambar 3.2:

1. *User* memberikan perintah atau meminta aplikasi untuk menampilkan data-data yang diinginkan seperti resep makanan, info perawatan, perkembangan, dan imunisasi.
2. Aplikasi mengambil data dari database melalui koneksi internet.
3. *User* memberikan perintah atau meminta aplikasi untuk memasukkan data catatan.
4. Aplikasi memasukkan data catatan ke database melalui koneksi internet.

3.4 Analisis Kebutuhan

- a. *Activity Splash Screen* sebagai pembuka agar tampilan lebih menarik.
- b. *Activity* menu registrasi untuk mendaftar.
- c. *Activity* menu *login* untuk masuk ke dalam aplikasi
- d. *Activity* menu utama yang menampilkan tombol-tombol menu catatan, menu kalkulator gizi, informasi, resep MPASI, favorit, dan menu grafik.
- e. *Activity* menu resep MPASI 6 sampai 7 bulan untuk melihat daftar resep MPASI 6 sampai 7 bulan.
- f. *Activity* menu resep MPASI 8 sampai 12 bulan untuk melihat daftar resep MPASI 8 sampai 12 bulan.
- g. *Activity* menu kalkulator gizi bayi perempuan.
- h. *Activity* menu kalkulator gizi bayi laki laki.
- i. *Activity* menu informasi bayi.
- j. *Activity* menu perawatan bayi

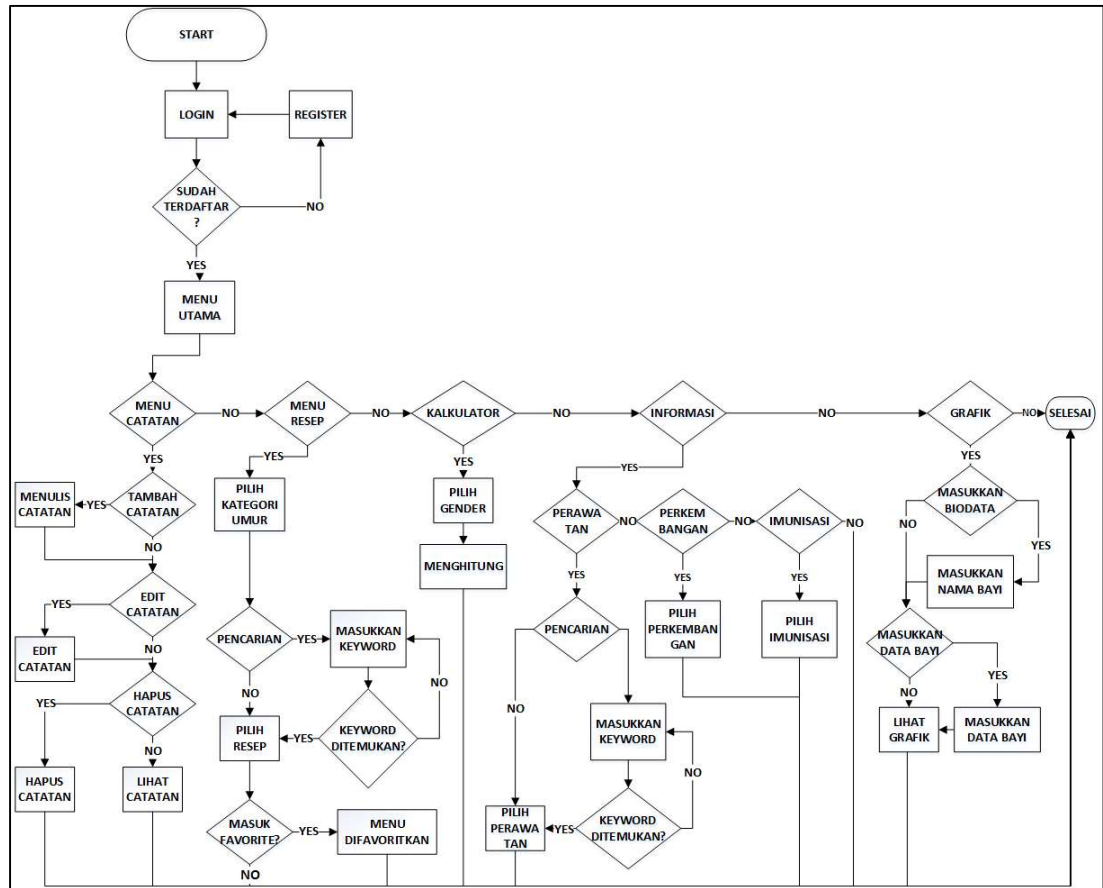
- k. *Activity* menu perkembangan bayi
- l. *Activity* catatan untuk membantu para ibu untuk mencatat hal-hal yang berkaitan dengan bayi.
- m. *Activity* menu nama bayi untuk memasukkan nama bayi.
- n. *Activity* menu data bayi untuk memasukkan umur dan berat bayi.
- o. *Activity* menu grafik untuk melihat grafik pertumbuhan dari bayi.
- p. *Activity* menu favorit untuk melihat resep yang sudah di favoritkan.

3.5 Rancangan Sistem

Metode yang digunakan dalam perancangan *logic* aplikasi *monitoring* pertumbuhan dan perkembangan bayi ini adalah *flowchart* dan *Unified Model Language* (UML). Adapun model UML yang digunakan yaitu *usecase diagram* dan *class diagram*.

3.5.1 Flowchart

Untuk membantu *logic* aplikasi ini maka dibuatlah *flowchart*. Flowchart dapat dilihat pada gambar 3.2.



Gambar 3.3 Flowchart Aplikasi

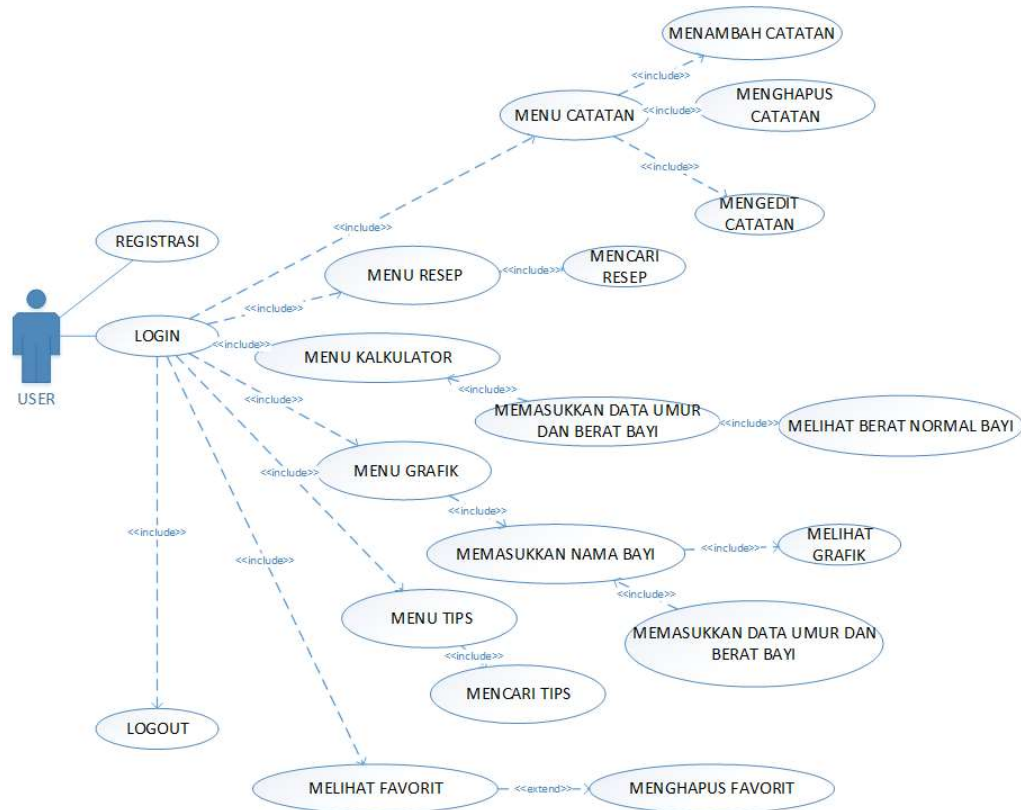
1. *User* harus *login* terlebih dahulu untuk menggunakan aplikasi. Jika *login* berhasil maka *Activity* menu utama akan tampil. Namun jika *user* belum memiliki akun maka *user* harus mendaftar terlebih dahulu di *Activity* registrasi. Setelah itu *user* bisa mencoba *login* lagi.
2. Ketika *user* selesai *login* maka *Activity* menu utama akan tampil. Pada *Activity* ini *user* bisa memilih menu-menu yang ada pada aplikasi ini.
3. Jika *user* memilih untuk membuka menu catatan maka *user* dapat menuju *Activity* menu catatan. Di dalam menu catatan ini *user* dapat melihat, menulis, mengedit, dan menghapus catatan.
4. Jika *user* memilih untuk membuka menu resep maka *user* dapat menuju ke *Activity* menu resep kemudian memilih kategori sesuai umur yang

diinginkan. Lalu *user* dapat melihat dan mencari resep yang ada. Selain itu *user* dapat menfavoritkan resep makanan.

5. Jika *user* memilih untuk membuka menu kalkulator maka *user* dapat menuju ke *Activity* menu kalkulator kemudian memilih kategori *gender* bayi. Lalu *user* dapat menghitung dan melihat berat ideal bayi.
6. Jika *user* memilih untuk membuka menu informasi maka *user* dapat menuju *Activity* menu informasi kemudian *user* dapat memilih informasi yang diinginkan seperti perawatan, perkembangan, dan imunisasi.
7. Jika *user* memilih untuk membuka menu grafik maka *user* dapat menuju *Activity* menu grafik kemudian *user* dapat memasukkan nama bayi, umur dan berat bayi serta melihat grafik pertumbuhan bayi.

3.5.2 Usecase Diagram

Berdasarkan analisis kebutuhan di atas maka dibuatlah *usecase* diagram untuk membantu pembuatan logic aplikasi. Gambar *usecase* diagram bisa dilihat pada gambar di bawah ini



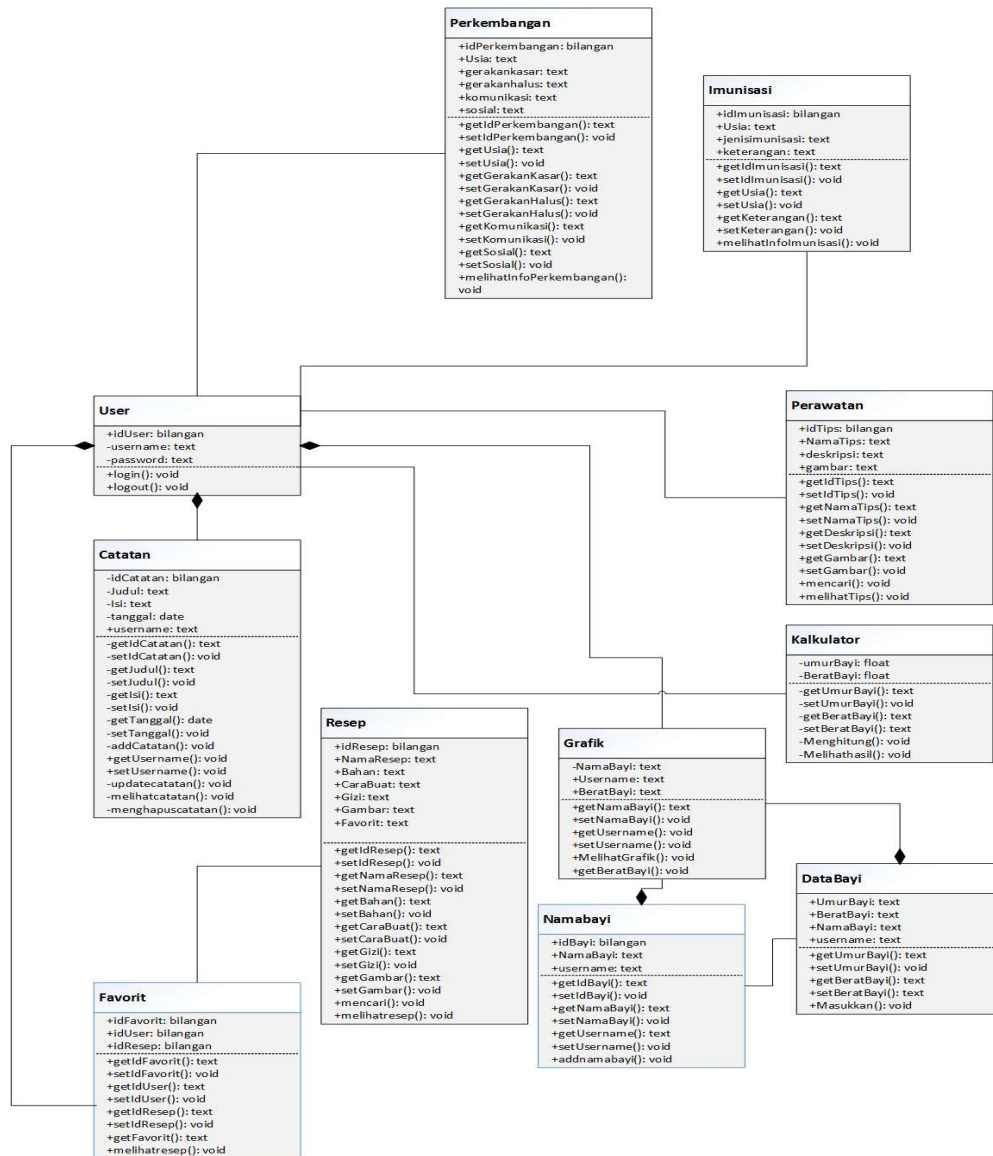
Gambar 3.4 Usecase Diagram

Adapun penjelasan dari gambar *usecase* di atas:

1. Registrasi: Memungkinkan *user* untuk melakukan registrasi akun agar bisa membuka aplikasi.
2. Login: Memungkinkan *user* untuk dapat melihat menu-menu yang ada di dalam aplikasi.
3. Melihat menu catatan: Memungkinkan *user* untuk menulis, mengedit, dan menghapus catatan.
4. Melihat Resep: Memungkinkan *user* untuk mengetahui resep MPASI dan mencari resep yang cocok untuk bayi berumur 6 sampai 12 bulan.
5. Melihat menu favorit: Memungkinkan *user* untuk melihat resep yang telah difavoritkan.
6. Melihat menu kalkulator: Memungkinkan *user* untuk memasukkan data umur dan berat agar dapat melihat berat badan ideal untuk bayi tersebut.

7. Melihat menu grafik: Memungkinkan *user* untuk memasukkan nama bayi, data dan umur bayi serta dapat melihat grafik pertumbuhan bayi tersebut.
8. Melihat menu informasi: Memungkinkan *user* untuk melihat dan mencari informasi dalam hal perawatan dan perkembangan bayi.

3.5.3 Class Diagram



Gambar 3.5 Class Diagram

Adapun penjelasan tentang *class* diagram di atas:

1. Kelas *user* memiliki fungsi *login* dan *logout* untuk mengakses menu di dalam aplikasi.
2. Kelas *catatan* memiliki fungsi menulis, melihat, mengedit, dan menghapus catatan.
3. Kelas *resep* memiliki fungsi untuk menampilkan dan mencari resep MPASI untuk usia 6 sampai 12 bulan.
4. Kelas *favorit* memiliki fungsi untuk menampilkan resep yang telah difavoritkan
5. Kelas *perawatan* memiliki fungsi untuk menampilkan dan mencari perawatan perawatan bayi.
6. Kelas *perkembangan bayi* memiliki fungsi untuk menampilkan perkembangan bayi.
7. Kelas *imunisasi* memiliki fungsi untuk menampilkan imunisasi bayi
8. Kelas *kalkulator* memiliki fungsi untuk menghitung dan menampilkan berat badan ideal bayi 1 sampai 12 bulan.
9. Kelas *nama bayi* memiliki fungsi untuk menambah nama bayi ke dalam *Database*.
10. Kelas *databayi* memiliki fungsi untuk memasukkan data nama, umur, dan berat badan bayi serta *username* ke dalam *Database*.
11. Kelas *grafik* memiliki fungsi menampilkan grafik umur dan berat badan bayi.
12. Kelas *catatan* memiliki relasi *composition* dengan kelas *user* dikarenakan catatan yang ditampilkan berdasarkan *username* yang *login*. Kelas *catatan* dapat mengakses *username* dengan *method getUsername()*.
13. Kelas *favorit* memiliki relasi *association* dengan kelas *resep* dimana kelas *favorit* dapat mengakses data resep melalui *method getIdResep()*.
14. Kelas *favorit* memiliki relasi *association* dengan kelas *user* dimana kelas *favorit* dapat mengakses *idUser* melalui *method getIdUser*.
15. Kelas *data bayi* memiliki relasi *association* dengan kelas *nama bayi* karena kelas *data bayi* dapat mengakses nama bayi melalui *method getNamaBayi()*

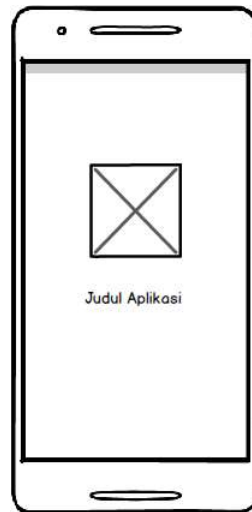
16. Kelas grafik mempunyai relasi *composition* dengan kelas nama bayi, data bayi, dan kelas *user* dikarenakan kelas grafik tidak bisa berdiri sendiri tanpa adanya nama bayi, *username*, dan data bayi. Nama bayi, *username*, dan databayi dapat diakses oleh kelas grafik melalui method *getUsername*, *getNamaBayi*, dan *getBeratBayi*, *getUmurBayi*.
17. Kelas kalkulator mempunyai relasi *association* dengan kelas *user*.
18. Kelas perawatan mempunyai relasi *association* dengan kelas *user*.
19. Kelas perkembangan mempunyai relasi *association* dengan kelas *user*.

3.6 Rancangan *User interface*

Perancangan antarmuka merupakan tampilan dari suatu perangkat lunak yang berperan sebagai media komunikasi antara perangkat lunak dan pengguna. Perancangan ini merupakan sebuah penggambaran, perencanaan dan pengaturan dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh. Perancangan antarmuka diharapkan memudahkan pengguna dalam melakukan proses interaksi terhadap sistem. Berikut adalah perancangan antar muka yang ada pada aplikasi *monitoring* pertumbuhan dan perkembangan bayi.

3.6.1 Perancangan *Splashscreen*

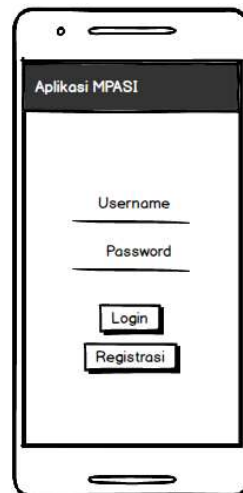
Perancangan *splashscreen* digunakan untuk tampilan *activity splashscreen* ketika *user* baru membuka aplikasi *monitoring* pertumbuhan dan perkembangan bayi. Perancangan *splashscreen* dapat dilihat pada gambar 3.5:



Gambar 3.6 *Splashscreen*

3.6.2 Perancangan *Activity Login*

Activity login dirancang dengan menampilkan 2 *edit text* dan 2 *button*. Perancangan tampilan *activity login* dapat dilihat pada gambar 3.6.

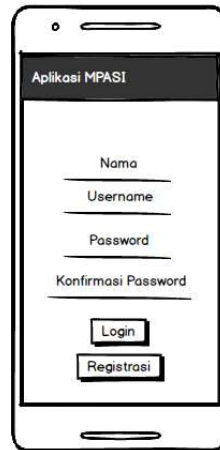


Gambar 3.7 *Activity Login*

3.6.3 Perancangan *Activity Registrasi*

Activity Registrasi dirancang dengan menampilkan 3 *edit text* yaitu *username*, *password*, dan konfirmasi *password* serta 2 *button* yaitu *login* dan

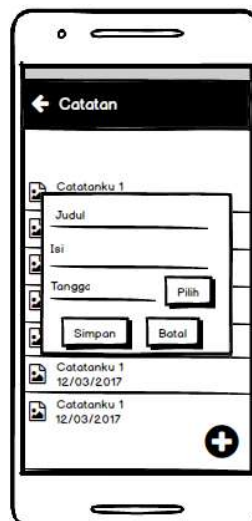
register. Perancangan antarmuka registrasi pada aplikasi *android monitoring pertumbuhan dan perkembangan bayi* dapat dilihat pada gambar 3.7.



Gambar 3.8 *Activity* Registrasi

3.6.4 Perancangan *Activity* Catatan

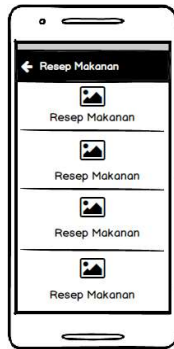
Activity catatan dirancang dengan menampilkan *dialogform* dan *listview*. Perancangan antarmuka catatan pada aplikasi *monitoring pertumbuhan dan perkembangan bayi* dapat dilihat pada gambar 3.8.



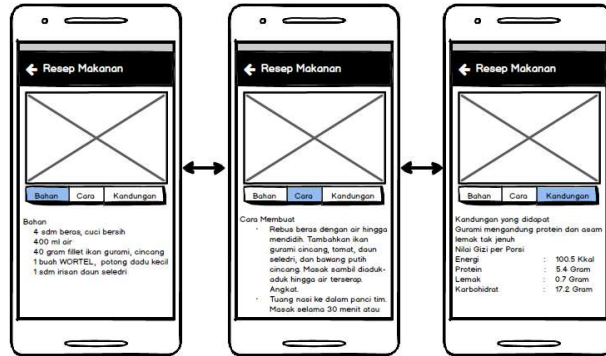
Gambar 3.9 *Activity* Catatan

3.6.5 Perancangan *Activity* Menu Resep

Perancangan antarmuka resep dan detail resep pada aplikasi *monitoring* pertumbuhan dan perkembangan bayi dapat dilihat pada gambar 3. 9 dan 3.10.



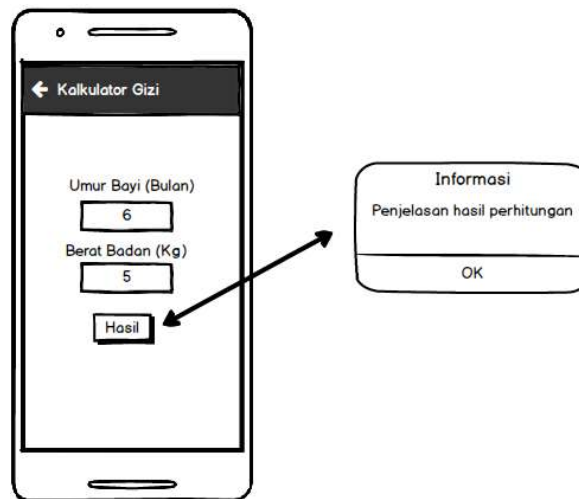
Gambar 3.10 *Listview* Resep



Gambar 3.11 Tampilan Resep

3.6.6 Perancangan *User interface Activity* Kalkulator

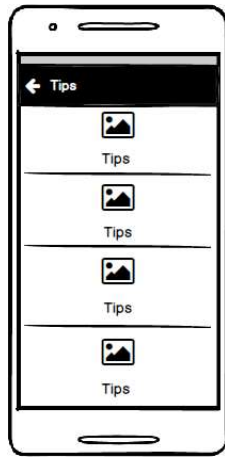
Perancangan antarmuka dari kalkulator gizi pada aplikasi *monitoring* pertumbuhan dan perkembangan bayi dapat dilihat pada gambar 3.11. Pada gambar 3.11 terdapat 2 *EditText* yang berfungsi untuk memasukan data bayi yang akan dihitung yaitu memasukan umur dalam bulan, berat badan dalam Kilogram (Kg) dan 1 *tombol* yang berfungsi untuk menghitung hasil.



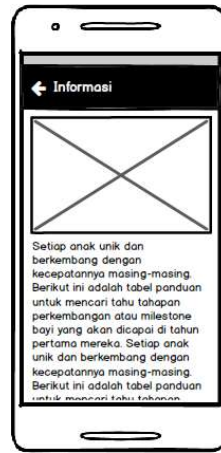
Gambar 3.12 Kalkulator Gizi

3.6.7 Perancangan *User interface Activity Info Perawatan*

Perancangan antarmuka tips pada aplikasi *monitoring* pertumbuhan dan perkembangan bayi dapat dilihat pada gambar 3.12 dan Perancangan antarmuka detail tips pada aplikasi *monitoring* pertumbuhan dan perkembangan bayi dapat dilihat pada gambar 3.13



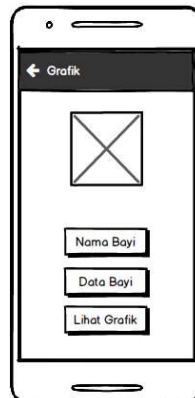
Gambar 3.13 *Listview Tips*



Gambar 3.14 *Detail Tips*

3.6.8 Perancangan *User interface Activity Grafik*

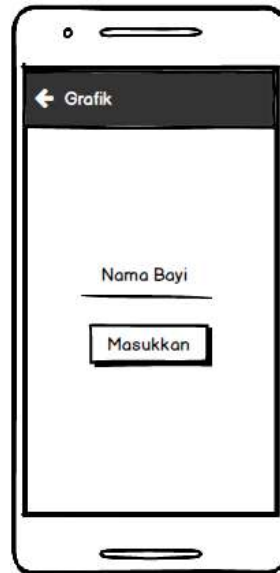
Perancangan antarmuka menu input data grafik dan menu grafik pada aplikasi *android monitoring* pertumbuhan dan perkembangan bayi dapat dilihat pada gambar 3.14.



Gambar 3.15 *Tampilan Menu Grafik*

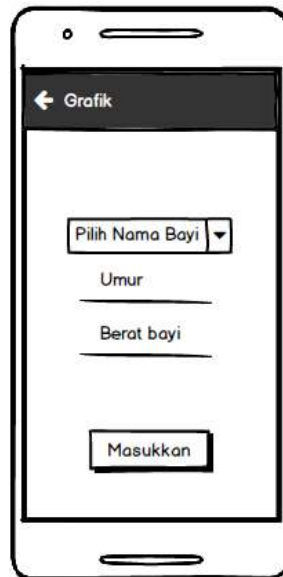
3.6.9 Perancangan *User Interface Activity* Nama Bayi

Perancangan antarmuka di tombol pilihan nama bayi pada aplikasi *android monitoring* pertumbuhan dan perkembangan bayi dapat dilihat pada gambar 3.15. Pada gambar 3.15 terdapat 1 *edit text* yang digunakan untuk mengisi nama bayi dan 1 tombol untuk memasukkan data.



Gambar 3.16 Menambah Nama Bayi

Perancangan antarmuka di tombol lihat grafik pada aplikasi *android monitoring* pertumbuhan dan perkembangan bayi dapat dilihat pada gambar 3.16. Pada gambar 3.16 terdapat 1 *spinner* yang menampilkan daftar nama bayi yang sudah ada, 2 *edit text* yang berfungsi untuk memasukkan umur dan berat badan bayi, dan 1 tombol yang berfungsi untuk menyimpan.



Gambar 3.17 Memilih nama Bayi

3.7 Metode Pengujian

Dalam mengembangkan sebuah aplikasi pengujian diperlukan untuk mengevaluasi jalannya aplikasi. Evaluasi digunakan untuk mengetahui apakah aplikasi tersebut sudah memenuhi target perencanaan dan melihat apakah masih ada kekurangannya. Pengujian dilakukan dengan *black box testing* dimana dengan metode ini, pengujian dilihat dari jalannya aplikasi secara fungsional.

Hal-hal yang akan diuji pada aplikasi *monitoring* pertumbuhan dan perkembangan bayi yaitu:

1. Aplikasi dapat memproses registrasi.
2. Aplikasi dapat memproses *login*.
3. Aplikasi dapat menampilkan *username*.
4. Aplikasi dapat menyimpan, melihat, mengedit, dan menghapus catatan yang ditulis oleh *user*.
5. Aplikasi dapat menampilkan daftar resep MPASI untuk usia 6 sampai 12 bulan.
6. Aplikasi berhasil memasukkan resep ke dalam menu favorit.
7. Aplikasi dapat menghitung berat badan ideal bayi.

8. Aplikasi dapat menampilkan informasi detail tentang berat badan bayi.
9. Aplikasi dapat menampilkan informasi tentang perawatan, perkembangan, dan imunisasi bayi.
10. Aplikasi dapat menampilkan informasi detail tentang perawatan, perkembangan, dan imunisasi bayi.
11. Aplikasi dapat menyimpan data umur dan berat badan bayi ke dalam *Database*.
12. Aplikasi dapat menampilkan grafik pertumbuhan bayi.