

BAB IV

HASIL DAN PEMBAHASAN

4.1 Pengembangan Sistem

Pengembangan *business logic* yang dilakukan dalam pembuatan implementasi aplikasi *monitoring* pertumbuhan dan perkembangan bayi ini menggunakan *software Netbeans* dengan bahasa *PHP* dan *Android Studio* dengan bahasa *java*.

4.2 Pembuatan Coding PHP

Coding PHP pada aplikasi *monitoring* pertumbuhan dan perkembangan bayi berfungsi sebagai *server side* atau jembatan antara *android studio* dan *Database MySQL*.

4.2.1 Script koneksi.PHP

```
<?php
$server      = "localhost";
$user        = "root"; //
$password    = ""; //
$database    = "skripsi";

$conn = mysql_connect($server, $user, $password) or die ("Koneksi gagal!");
mysql_select_db($database) or die ("Database belum siap!");
?>
```

Gambar 4.1 Coding Koneksi.PHP

Gambar 4.1 menampilkan *coding* koneksi. Dimana pada *coding* ini *PHP* menjadi pondasi pembuka dan penutup koneksi antara *android studio* dan *Database*.

4.2.2 Script login.PHP

```

<?php
include 'koneksi.php';
class usr()
$username = $_POST["username"];
$password = $_POST["password"];
if ((empty($username)) || (empty($password))) {
    $response = new usr();
    $response->success = 0;
    $response->message = "Kolom tidak boleh kosong";
    die(json_encode($response));
}
$query = mysql_query("SELECT * FROM registrasi WHERE username='$username' AND password='$password'");
$rows = mysql_fetch_array($query);

if (!empty($rows)) {
    $response = new usr();
    $response->success = 1;
    $response->message = "Selamat datang ". $rows['username'];
    $response->idMember = $rows['idMember'];
    $response->username = $rows['username'];
    die(json_encode($response));
} else {
    $response = new usr();
    $response->success = 0;
    $response->message = "Username atau password salah";
    die(json_encode($response));
}

```

Gambar 4.2 Coding Login.PHP

Gambar 4.2 menampilkan *coding* yang digunakan *PHP* untuk memanggil *username* dan *password* melalui *query* “SELECT * FROM” pada tabel registrasi. Jika data *user* benar maka *PHP* akan menampilkan pesan “Selamat Datang”. Ketika data *user* tidak terdaftar maka *PHP* akan menampilkan pesan “Username atau password salah”.

4.2.3 Script register.PHP

```

<?php
include 'koneksi.php';
class usr()
$username = $_POST["username"];
$password = $_POST["password"];
$confirm_password = $_POST["confirm_password"];
if ((empty($username)) {
    $response = new usr();
    $response->success = 0;
    $response->message = "Kolom username tidak boleh kosong";
    die(json_encode($response));
} else if ((empty($password)) {
    $response = new usr();
    $response->success = 0;
    $response->message = "Kolom password tidak boleh kosong";
    die(json_encode($response));
} else if ((empty($confirm_password)) || ($password != $confirm_password)) {
    $response = new usr();
    $response->success = 0;
    $response->message = "Konfirmasi password tidak sama";
    die(json_encode($response));
} else {
    if ((empty($username)) && $password == $confirm_password) {
        $num_rows = mysql_num_rows(mysql_query("SELECT * FROM registrasi WHERE username='".$username."'"));
        if ($num_rows == 0) {
            $query = mysql_query("INSERT INTO registrasi (idMember, username, password) VALUES(0, '".$username."', '".$password."'");
        }
    }
}

```

Gambar 4.3 Coding Register.PHP

Gambar 4.3 menampilkan *coding PHP* yang akan digunakan pada proses registrasi *user* ketika membuat akun pada aplikasi. Pada *Script* ini *PHP* menggunakan *query* “SELECT * FROM” dan “INSERT INTO”.

Registrasi *user* membutuhkan data *username*, *password*, dan konfirmasi *password*. Pada awal *coding* tersebut terdapat beberapa validasi data yaitu *PHP* memeriksa apakah *user* sudah melengkapi kolom data seperti *username*, *password*, dan konfirmasi *password*.

Kemudian setelah *user* sudah melengkapi data, *PHP* memeriksa apakah *username* sudah terdaftar atau belum pada tabel registrasi. Jika sudah ada maka *PHP* akan merespon dengan mengirimkan pesan “*Username* sudah ada”. Apabila *username* belum ada, maka *PHP* akan memasukkan *username* baru ke dalam *Database* melalui *query* “INSERT INTO” ke dalam tabel registrasi dan *PHP* akan merespon dengan pesan “Registrasi berhasil. Silahkan login”.

4.2.4 *Script select.PHP*

```

<?php
include "koneksi.php";
$username= $_GET['username'];
$query = mysql_query("SELECT * FROM catatan where username = '". $username ." ' ORDER BY judul ASC");

$json = array();

while($row = mysql_fetch_assoc($query)){
    $json[] = $row;
}

echo json_encode($json);

mysql_close($connect);
?>

```

Gambar 4.4 *Coding Select.PHP*

Gambar 4.4 menampilkan *coding* yang digunakan untuk memanggil semua data yang ada pada tabel catatan berdasarkan *user* yang *login* dan diurutkan berdasarkan tanggal terbaru melalui *query* “SELECT * FROM”.

4.2.5 Script insert.PHP

```

include "koneksi.php";

$judul = $_POST['judul'];
$isi = $_POST['isi'];
$tanggal = $_POST['tanggal'];
$username = $_POST['username'];

class emp{

if (empty($judul) || empty($isi)) {
    $response = new emp();
    $response->success = 0;
    $response->message = "Kolom isian tidak boleh kosong";
    die(json_encode($response));
} else {
    $query = mysql_query("INSERT INTO catatan (id,judul,isi,tanggal,username) VALUES (0, '".$judul."', '".$isi."', '"
    if ($query) {
        $response = new emp();
        $response->success = 1;
        $response->message = "Data berhasil di simpan";
        die(json_encode($response));
    } else{
        $response = new emp();
        $response->success = 0;
        $response->message = "Error simpan Data";
        die(json_encode($response));
    }
}
}

```

Gambar 4.5 Coding Insert.PHP

Pada gambar 4.5 menampilkan *coding PHP* yang mempunyai *query* “INSERT INTO” untuk memasukkan data judul, isi, tanggal, dan *username* ke dalam tabel catatan. Jika berhasil maka *PHP* akan menampilkan pesan “Data Berhasil di simpan” dan jika gagal maka *PHP* akan menampilkan pesan “Error Simpan Data”.

4.2.6 Script Update.PHP

```

include "koneksi.php";

$id = $_POST['id'];
$judul = $_POST['judul'];
$isi = $_POST['isi'];
$tanggal = $_POST['tanggal'];
class emp{

if (empty($id) || empty($judul) || empty($isi)) {
    $response = new emp();
    $response->success = 0;
    $response->message = "Kolom isian tidak boleh kosong";
    die(json_encode($response));
} else {
    $query = mysql_query("UPDATE catatan SET judul='".$judul."', isi='".$isi."',
    . " tanggal='".$tanggal.'" WHERE id='".$id.'"");
    if ($query) {
        $response = new emp();
        $response->success = 1;
        $response->message = "Data berhasil diperbarui ";
        die(json_encode($response));
    } else{
        $response = new emp();
        $response->success = 0;
        $response->message = "Error update Data";
        die(json_encode($response));
    }
}
}

```

Gambar 4.6 Coding Update.PHP

Gambar 4.6 menampilkan *coding PHP* yang mempunyai *query* “SELECT * FROM” untuk mengedit data catatan yang ada berdasarkan idCatatan. Ketika berhasil mengedit data maka *PHP* akan menampilkan pesan “Data berhasil diperbarui” dan ketika gagal mengedit maka *PHP* akan menampilkan pesan “Error update data”.

4.2.7 Script Delete.PHP

```

k?php
include "koneksi.php";

$id    = $_POST['id'];

class emp{}

if (empty($id)) {
    $response = new emp();
    $response->success = 0;
    $response->message = "Error hapus Data";
    die(json_encode($response));
} else {
    $query = mysql_query("DELETE FROM catatan WHERE id='".$id."'");

    if ($query) {
        $response = new emp();
        $response->success = 1;
        $response->message = "Data berhasil di hapus";
        die(json_encode($response));
    } else{
        $response = new emp();
        $response->success = 0;
        $response->message = "Error hapus Data";
        die(json_encode($response));
    }
}
?>

```

Gambar 4.7 Coding Delete.PHP

Gambar 4.7 menampilkan *coding PHP* yang mempunyai *query* “DELETE * FROM” untuk menghapus data catatan berdasarkan idCatatan. Ketika berhasil menghapus data maka *PHP* akan menampilkan pesan “Data berhasil dihapus” dan ketika gagal menghapus data maka *PHP* akan menampilkan pesan “Error update data”.

4.2.8 Script Filterresep.PHP

```

<?php
include 'koneksi.php';
$namaresep = $_GET['namaresep'];
$query
    = mysql_query("SELECT * FROM resep where namaresep like '%$namaresep%' AND usia < 8")
    or die(mysql_error());
$json = array();

while($row = mysql_fetch_assoc($query)){
    $json[] = $row;
}

echo json_encode($json);

mysql_close($connect);
?>

```

Gambar 4.8 Coding Filterresep.PHP

Pada gambar 4.8 PHP mempunyai *query* “SELECT * FROM” untuk menampilkan semua resep dan difilter berdasarkan usia kurang dari 8 bulan. Coding filterresep juga digunakan untuk fitur pencarian pada aplikasi yaitu difilter berdasarkan nama resep.

4.2.9 Script Filterresep2.PHP

```

<?php
include 'koneksi.php';
$namaresep = $_GET['namaresep'];
$query
    = mysql_query("SELECT * FROM resep where namaresep like '%$namaresep%' AND usia > 6")
    or die(mysql_error());
$json = array();

while($row = mysql_fetch_assoc($query)){
    $json[] = $row;
}

echo json_encode($json);

mysql_close($connect);
?>

```

Gambar 4.9 Coding Filterresep2.PHP

Pada gambar 4.9 PHP mempunyai *query* “SELECT * FROM” untuk menampilkan semua resep dan usia lebih dari 6 bulan. Coding filterresep2 juga digunakan untuk fitur pencarian pada aplikasi yaitu difilter berdasarkan nama resep.

4.2.10 Script Detailresep.PHP

```

<?php
include "koneksi.php";
$idResep = $_POST['idResep'];
$query = mysql_query("SELECT * FROM resep WHERE idResep='".$idResep."'");
while ($row = mysql_fetch_array($query)){
    $char = "";

    $json = '{
        "idResep": "'.str_replace($char, '', strip_tags($row['idResep'])).'",
        "NamaResep": "'.str_replace($char, '', strip_tags($row['NamaResep'])).'",
        "Bahan": "'.str_replace($char, '', strip_tags($row['Bahan'])).'",
        "CaraBuat": "'.str_replace($char, '', strip_tags($row['CaraBuat'])).'",
        "Gizi": "'.str_replace($char, '', strip_tags($row['Gizi'])).'",
        "gambar": "'. $row['gambar'] .'"';
    }

    echo $json;
    mysql_close($connect);

```

Gambar 4.10 Coding Detailresep.PHP

Gambar 4.10 menampilkan *coding PHP* yang mempunyai *query* “SELECT * FROM” untuk menampilkan detail resep seperti bahan, cara buat, dan gizi pada satu resep berdasarkan idResep.

4.2.11 Script Filtertips.PHP

```

<?php
include 'koneksi.php';
$jenisperawatan = $_GET['jenisperawatan'];
$query = mysql_query("SELECT * FROM tips where jenisperawatan like '%$jenisperawatan%' ")
or die(mysql_error());
$json = array();

while($row = mysql_fetch_assoc($query)){
    $json[] = $row;
}

echo json_encode($json);

mysql_close($connect);
?>

```

Gambar 4.11 Coding Filtertips.PHP

Gambar 4.11 menampilkan *coding PHP* yang mempunyai *query* “SELECT * FROM” untuk menampilkan semua data tips pada tabel tips dan di filter berdasarkan jenis perawatan.

4.2.12 Script Detailtips.PHP

```

<?php
include "koneksi.php";
$idperawatan = $_POST['idperawatan'];
$query = mysql_query("SELECT * FROM tips WHERE idperawatan='".$idperawatan."'");
while ($row = mysql_fetch_array($query)) {
    $char = '';

    $json = '{
        "idperawatan": "'.str_replace($char, '', strip_tags($row['idperawatan'])).'",
        "jenisperawatan": "'.str_replace($char, '', strip_tags($row['jenisperawatan'])).'",
        "keterangan": "'.str_replace($char, '', strip_tags($row['keterangan'])).'",
        "gambar": "'. $row['gambar']. "'";
    }

    echo $json;
    mysql_close($connect);
?>

```

Gambar 4.12 Coding Detailtips.PHP

Gambar 4.12 menampilkan *coding PHP* yang mempunyai *query* “SELECT * FROM” untuk menampilkan detail tips seperti jenis perawatan, keterangan, dan gambar berdasarkan idperawatan.

4.2.13 Script FilterPerkembangan.PHP

```

<?php
include 'koneksi.php';
$usia = $_GET['usia'];
$query = mysql_query("SELECT * FROM bayiperkembangan where usia like '%$usia%' ")
or die(mysql_error());
$json = array();

while($row = mysql_fetch_assoc($query)){
    $json[] = $row;
}

echo json_encode($json);
mysql_close($connect);
?>

```

Gambar 4.13 Coding Filterperkembangan.PHP

Gambar 4.13 menampilkan *coding PHP* yang mempunyai *query* “SELECT * FROM” untuk mengambil seluruh data dari tabel bayiperkembangan. *Coding* ini juga berguna untuk fitur pencarian pada aplikasi yaitu difilter berdasarkan usia.

4.2.14 Script DetailPerkembangan.PHP

```

<?php
include "koneksi.php";
$idPerkembangan = $_POST['idPerkembangan'];
$query = mysql_query("SELECT * FROM bayiperkembangan WHERE idPerkembangan='".$idPerkembangan."'");
while ($row = mysql_fetch_array($query)) {
    $char = "";

    $json = '{
        "idPerkembangan": "'.str_replace($char, '', strip_tags($row['idPerkembangan'])).'",
        "usia": "'.str_replace($char, '', strip_tags($row['usia'])).'",
        "gerakankasar": "'.str_replace($char, '', strip_tags($row['gerakankasar'])).'",
        "gerakanhalus": "'.str_replace($char, '', strip_tags($row['gerakanhalus'])).'",
        "komunikasi": "'.str_replace($char, '', strip_tags($row['komunikasi'])).'",
        "sosial": "'.str_replace($char, '', strip_tags($row['sosial'])).'",
        "stimulasikasar": "'.str_replace($char, '', strip_tags($row['stimulasikasar'])).'",
        "stimulasihalus": "'.str_replace($char, '', strip_tags($row['stimulasihalus'])).'",
        "gambar": "'. $row['gambar']. '";
    }

    echo $json;
    mysql_close($connect);
}

```

Gambar 4.14 Coding Detailperkembangan.PHP

Gambar 4.19 menampilkan *coding PHP* yang mempunyai *query* “SELECT * FROM” untuk menampilkan detail perkembangan bayi berdasarkan *idPerkembangan*.

4.2.15 Script InsertFavorit.PHP

```

if ((empty($idMember)) {
    $response = new usr();
    $response->success = 0;
    $response->message = "Kolom username tidak boleh kosong";
    die(json_encode($response));
} else if ((empty($idResep)) {
    $response = new usr();
    $response->success = 0;
    $response->message = "Kolom password tidak boleh kosong";
    die(json_encode($response));
}
else {
    $num_rows = mysql_num_rows(mysql_query("SELECT * FROM tabelfavorit WHERE idMember='".$idMember."'"));
    if ($num_rows == 0) {
        if (!empty($idMember) && !empty($idResep)) {
            $query = mysql_query("INSERT INTO tabelfavorit (idFavorit, idMember, idResep, favorit) V");
            if ($query) {
                $response = new usr();
                $response->success = 1;
                $response->message = "Berhasil masuk daftar favorit";
                die(json_encode($response));
            } else {
                $response = new usr();
                $response->success = 0;
                // $response->message = "Username sudah ada";
                die(json_encode($response));
            }
        }
    }
}

```

Gambar 4.15 Coding Insertfavorit.PHP

Gambar 4.12 menampilkan *coding PHP* yang mempunyai *query* “INSERT INTO” ke dalam tabel favorit. Sebelum memasukkan data *idFavorit*, *idMember*, dan *idResep*, *PHP* melakukan fungsi *query* “SELECT * FROM” untuk memeriksa apakah *idMember* dan *idResep* yang sama sudah ada. Jika sudah ada maka proses

pemasukkan data ke dalam tabel favorit gagal dan apabila tidak ada maka proses pemasukkan data berhasil.

4.2.16 Script Deletefavorit.PHP

```

$idMember = $_GET['idMember'];
$idResep = $_GET['idResep'];
$idFavorit = $_GET['idFavorit'];

class emp()
{
    if (empty($idMember)) {
        $response = new emp();
        $response->success = 0;
        $response->message = "Error hapus Data";
        die(json_encode($response));
    } else {
        $query = mysql_query("DELETE FROM tabelfavorit WHERE idMember='".$idMember.'" AND idResep='".$idResep.'"");
        if ($query) {
            $response = new emp();
            $response->success = 1;
            $response->message = "Favorit di hapus";
            die(json_encode($response));
        } else {
            $response = new emp();
            $response->success = 0;
            $response->message = "Error hapus Data";
            die(json_encode($response));
        }
    }
}

```

Gambar 4.16 Coding Deletefavorit.PHP

Gambar 4.13 menampilkan *coding PHP* yang mempunyai *query* “DELETE * FROM” pada tabel favorit untuk menghapus resep favorit berdasarkan idMember dan idResep. Jika berhasil maka *PHP* akan menampilkan pesan “Favorit Dihapus”.

4.2.17 Script Selectfavorit.PHP

```

<?php
include "koneksi.php";
$idMember= $_GET['idMember'];
$query = mysql_query("SELECT resep.idResep, resep.NamaResep, resep.Bahan, resep.CaraBuat,
resep.Gizi, resep.gambar
FROM tabelfavorit
INNER JOIN resep ON tabelfavorit.idResep=resep.idResep
INNER JOIN registrasi ON tabelfavorit.idMember=registrasi.idMember WHERE tabelfavorit.idMember='".$idMember.'"");

$json = array();
while($row = mysql_fetch_assoc($query)){
    $json[] = $row;
}

echo json_encode($json);
mysql_close($connect);

```

Gambar 4.17 Coding Selectfavorit.PHP

Gambar 4.14 menampilkan *coding PHP* yang mempunyai *query* “*SELECT * FROM*” untuk memanggil semua data resep berdasarkan *idMember* dan *idResep* pada tabel favorit.

4.2.18 Script *Insertbiodatabayi.PHP*

```

class usr()
{
    $namabayi = $_POST["namabayi"];
    $username = $_POST["username"];
    if (empty($namabayi)) {
        $response = new usr();
        $response->success = 0;
        die(json_encode($response));
    }
    else {
        if (empty($username)) {
            $query = mysql_query("INSERT INTO biodatabayi (id,namabayi, username) VALUES"
                . "[0, '$namabayi', '$username']");
        }
        if ($query) {
            $response = new usr();
            $response->success = 1;
            $response->message = "Berhasil Memasukkan Data";
            die(json_encode($response));
        }
        else {
            $response = new usr();
            $response->success = 0;
            $response->message = "nama bayi sudah ada";
            die(json_encode($response));
        }
    }
}

mysql_close();

```

Gambar 4.18 Coding *Insertbiodatabayi.PHP*

Gambar 4.15 menampilkan *coding PHP* yang mempunyai *query* “*INSERT INTO*” untuk memasukkan nama bayi dan *username* ke dalam tabel biodata bayi. Jika berhasil maka *PHP* akan menampilkan pesan “Data Berhasil di simpan” dan jika gagal maka *PHP* akan menampilkan pesan “Error Simpan Data”.

4.2.19 Script *GetSpinner.PHP*

```

<?php

include "koneksi.php";
$username = $_GET['username'];
$query = mysql_query("SELECT * FROM biodatabayi where username= '$username.'");

$json = array();

while($row = mysql_fetch_assoc($query)) {
    $json[] = $row;
}

echo json_encode($json);

```

Gambar 4.19 Coding *Getspinner.PHP*

Gambar 4.16 menampilkan *coding PHP* yang mempunyai *query* “SELECT * FROM” untuk menampilkan nama bayi dari tabel biodatabayi berdasarkan *username*.

4.2.20 Script InsertGrafik.PHP

```
<?php
include 'koneksi.php';
class usr{
    $namabayi= $_POST["namabayi"];
    $umur = $_POST["umur"];
    $berat= $_POST["berat"];
    $username= $_POST["username"];

    if ((empty($namabayi)) {
        $response = new usr();
        $response->success = 0;
        $response->message = "Nama bayi tidak boleh kosong";
        die(json_encode($response));
    } else if ((empty($umur)) {
        $response = new usr();
        $response->success = 0;
        $response->message = "Umur tidak boleh kosong";
        die(json_encode($response));
    }
    else {
        $num_rows = mysql_num_rows(mysql_query("SELECT * FROM grafikp WHERE namabayi='".$namabayi.'" "
        . "AND umur='".$umur.'" AND username='".$username.'""));
        if ($num_rows == 0){
            if (!empty($namabayi) && !empty($umur) ){
                $query = mysql_query("INSERT INTO grafikp (id,namabayi, umur, berat, username) "
                . "VALUES(0,'" . $namabayi . "','" . $umur . "','" . $berat . "','" . $username .'"");
                if ($query){
                    $response = new usr();
                    $response->success = 1;
                    $response->message = "Berhasil memasukkan data";
                    die(json_encode($response));
                }
            }
        }
    }
}
```

Gambar 4.20 Coding Insertgrafik.PHP

Gambar 4.17 menampilkan *coding PHP* yang mempunyai *query* “SELECT * FROM” yang berfungsi untuk memeriksa apakah sudah ada umur dan *username* yang sama pada tabel grafikp. Jika ada maka data tidak berhasil masuk ke dalam tabel grafikp dan jika belum ada maka *PHP* akan melakukan fungsi *query* “INSERT INTO” untuk memasukkan data umur, berat, *username*, dan nama bayi ke dalam tabel grafikp.

4.2.21 Script Detailgrafik.PHP

```

<?php
include "koneksi.php";

$namabayi= $_GET['namabayi'];
$username= $_GET['username'];
$query = mysql_query("SELECT umur,berat FROM grafikp where namabayi = '". $namabayi ." '
. "AND username = '". $username ." ' GROUP BY umur");
$data="";
$count = mysql_num_rows($query); $count2=0;
while ($row = mysql_fetch_array($query))
{
    $count2=$count2+1;
    if($count == $count2)
    {
        $data .= '{"umur": "'. $row['umur'] ." ", "berat": "'. $row['berat'] ." '";
    }
    else {
        $data .= '{"umur": "'. $row['umur'] ." ", "berat": "'. $row['berat'] ." '";
    }
}
$json = "[".$data."]";
echo $json;
mysql_close($connect);
?>

```

Gambar 4.21 Coding Detailgrafik.PHP

Gambar 4.20 menampilkan *coding PHP* yang terdapat *query* “SELECT * FROM” untuk menampilkan detail data bayi dari tabel grafik tiap bulannya berdasarkan nama bayi dan *username*.

4.3 Implementasi pada Android Studio

Coding implementasi pada aplikasi *monitoring* pertumbuhan dan perkembangan bayi terbagi menjadi dua yaitu antarmuka dan fungsi sistem. Implementasi antarmuka dilakukan pada folder *layout*, *drawable*, dan *styles*. Sedangkan implementasi *coding* fungsi dilakukan pada folder *Java*. Adapun hasil implementasi yang telah dilakukan sebagai berikut:

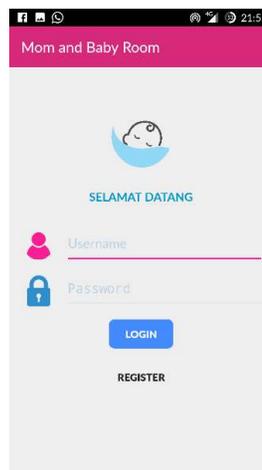
4.3.1 Activity Splashscreen



Gambar 4.22 Tampilan *Splashscreen*

Pada gambar 4.22 menampilkan tampilan awal aplikasi *monitoring* pertumbuhan dan perkembangan bayi.

4.3.2 Activity Login



Gambar 4.23 Tampilan Activity Login

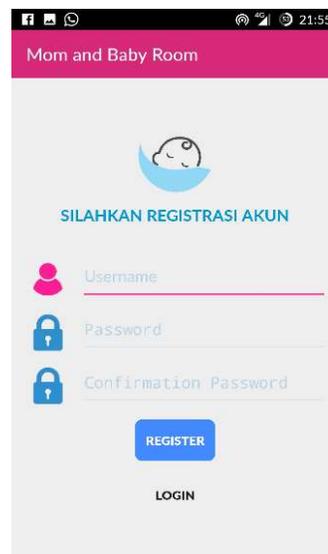
1. Gambar 4.23 menampilkan *Activity login*. Pada *Activity* ini *user* harus mengisi kolom *username* dan *password*. Kemudian *user* harus mengklik tombol login agar dapat mengakses menu dalam aplikasi.



Gambar 4.24 Notifikasi validasi

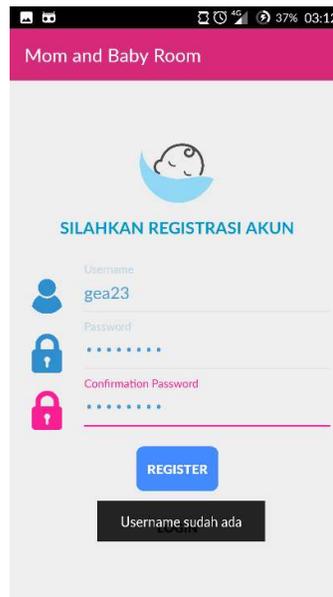
2. Pastikan *user* sudah memiliki *username* dan *password*. Jika *username* dan *password* benar maka aplikasi akan menampilkan *Activity* menu utama dan jika tidak maka aplikasi akan menampilkan notifikasi “*username* atau *password* salah” seperti pada gambar 4.24 sehingga *user* harus mendaftar terlebih dahulu.

4.3.3 *Activity* Registrasi



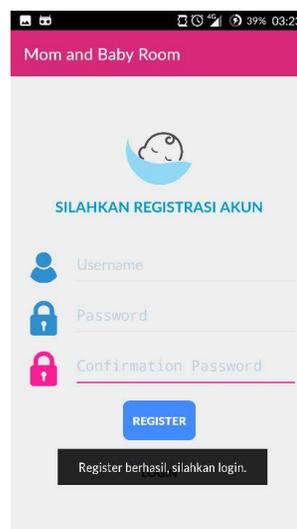
Gambar 4.25 Activity Registrasi

1. Gambar 4.25 menampilkan *Activity* registrasi akun. *User* harus mengisi kolom nama, *username*, *password*, dan konfirmasi *password*. *User* memilih tombol register untuk mendaftar.



Gambar 4.26 Notifikasi Validasi

2. Gambar 4.26 menampilkan validasi *username* ketika *username* sudah terdaftar maka aplikasi menampilkan notifikasi “*Username* sudah ada”.



Gambar 4.27 Notifikasi Register Berhasil

3. Gambar 4.27 menampilkan notifikasi “Register berhasil. Silahkan Login” ketika *user* berhasil mendaftar akun.

4.3.4 *Activity* Menu Utama



Gambar 4.28 *Activity* Menu Utama

Gambar 4.28 menampilkan *activity* menu utama aplikasi *monitoring* pertumbuhan dan perkembangan bayi. Pada *activity* ini terdapat banyak tombol menu seperti catatan, resep, kalkulator, informasi, grafik, dan favorit. *User* akan diarahkan ke *activity* yang dituju ketika menekan salah satu tombol menu di *activity* menu utama.

4.3.5 Activity Catatan



Gambar 4.29 Default Activity Catatan

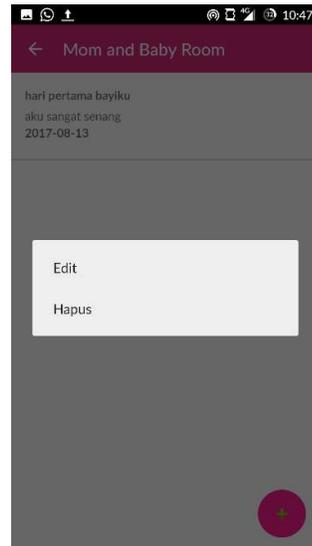


Gambar 4.30 Tampilan Form Catatan

1. Gambar 4.29 menampilkan *default* tampilan menu catatan ketika tidak ada catatan yang disimpan. *User* dapat menambahkan catatan dengan menekan tombol *add* pada menu catatan.
2. Gambar 4.30 menampilkan *form* catatan ketika *user* menekan tombol *add* untuk menambahkan catatan. Pada *form* ini terdapat kolom judul, isi, dan tanggal. *User* dapat menyimpan catatan dengan menekan tombol “Simpan”. atau dapat membatalkan *Activity* catatan akan menampilkan catatan pada *listview* dan *user* dapat mengedit atau menghapus catatan dengan cara menekan lama catatan yang ingin diedit atau dihapus.



Gambar 4.31 Tampilan *Listview* catatan



Gambar 4.32 Menu Edit dan Hapus

3. Gambar 4.31 menampilkan *listview* catatan yang sudah disimpan *user*.
4. Gambar 4.32 menampilkan tombol “Edit” dan “Hapus” ketika *user* menekan lama catatan yang ingin diedit atau dihapus.

4.3.6 *Activity* Menu Pilih Resep



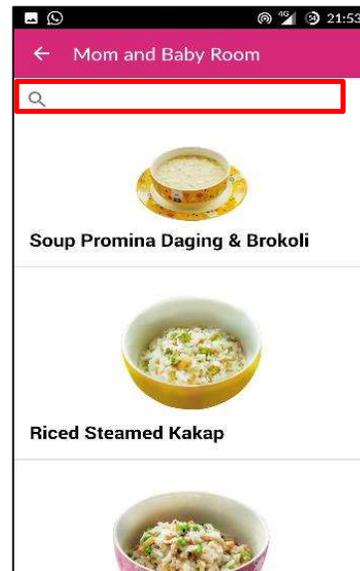
Gambar 4.33 Menu Pilih Resep

Gambar 4.33 menampilkan *Activity* pilih resep dimana *user* dapat memilih menu resep MPASI berdasarkan umur bayi. Ketika *user* memilih tombol 6 sampai 7 bulan maka *user* akan diarahkan ke *Activity* menu 6 sampai 7 bulan dan ketika *user* memilih tombol 8 sampai 12 maka *user* akan diarahkan ke *Activity* 8 sampai 12 bulan.

4.3.7 *Activity* Menu Resep



Gambar 4.34 Menu resep 6 sampai 7 Bulan



Gambar 4.35 Menu resep 8 sampai 12 Bulan

Gambar 4.34 dan Gambar 4.35 menampilkan *activity* menu resep MPASI 6 sampai 7 bulan dan menu resep 8 sampai 12 bulan. *User* dapat melihat menu-menu resep yang ada dengan cara *scrolling*. Pada *Activity* ini juga terdapat tombol search yang berfungsi ketika *user* ingin mencari satu nama resep. Ketika kata kunci yang dicari ada maka akan muncul nama resep tersebut dan apabila nama resep yang dicari tidak ada maka akan muncul tulisan “Data tidak ditemukan”. Gambar pencarian dapat dilihat pada Gambar 4.35 dan Gambar ketika kata kunci tidak ditemukan dapat dilihat pada Gambar 4.36



Gambar 4.36 Filter Resep



Gambar 4.37 Data Tidak Ditemukan

User dapat melihat detail resep dengan memilih daftar resep yang ada. Setelah memilih resep *user* dapat menjadikan resep sebagai resep favorit. Fitur favorit berguna bagi *user* agar tidak perlu lagi mencari menu resep yang ingin dicoba. Gambar tampilan detail resep dan favorit dapat dilihat pada Gambar 4.38 dan Gambar 4.39.



Gambar 4.38 Detail Resep



Gambar 4.39 Pilih Tombol Favorit

4.3.8 Activity Menu Pilih Kalkulator



Gambar 4.40 Menu Kalkulator Bayi

Gambar 4.40 menampilkan *Activity* kalkulator bayi dimana *user* dapat memilih kalkulator berdasarkan jenis kelamin bayi. Ketika *user* memilih tombol bayi perempuan maka *user* akan diarahkan ke *Activity* menu kalkulator bayi perempuan dan apabila *user* memilih tombol bayi laki-laki maka *user* akan diarahkan ke *Activity* kalkulator bayi laki-laki.

4.3.9 Activity Menu Kalkulator Bayi Perempuan dan Laki-laki

Gambar 4.41 Form Hitung Berat Badan

Gambar 4.42 Hasil Perhitungan

1. Gambar 4.41 menampilkan *Activity* kalkulator bayi. Pada *Activity* ini *user* harus memasukkan data umur dan berat bayi jika hendak melihat berat normal bayi tiap bulannya. Ketika *user* menekan tombol hitung maka akan muncul *dialogform* yang menampilkan hasil perhitungan.
2. Gambar 4.42 menampilkan *dialogform* ketika *user* menekan tombol “Hitung”.

4.3.10 *Activity* Menu Pilih Informasi



Gambar 4.43 *Activity* Menu Pilih Informasi

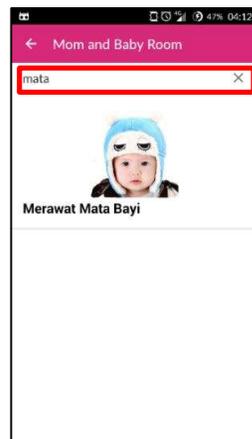
Gambar 4.43 menampilkan *Activity* pilih informasi dimana *user* dapat memilih informasi yang ingin dilihat. Ketika *user* memilih tombol perawatan maka *user* akan diarahkan ke *Activity* menu perawatan dan apabila *user* memilih menu perkembangan maka *user* akan diarahkan ke *Activity* menu perkembangan.

4.3.11 Activity Menu Perawatan



Gambar 4.44 Listview Perawatan

1. Gambar 4.44 menampilkan *Activity* perawatan bayi. Pada *Activity* ini *user* dapat melihat berbagai perawatan bayi dengan cara *scrolling*. Pada *Activity* ini juga terdapat fitur *search* dimana *user* dapat mencari satu perawatan tertentu.



Gambar 4.45 Filter Perawatan



Gambar 4.46 Data Tidak Ditemukan

2. Gambar 4.45 tampilan ketika kata kunci yang dicari ada maka aplikasi akan menampilkan perawatan yang dicari.

3. Gambar 4.46 ketika kata kunci yang dicari tidak ada maka aplikasi akan menampilkan “Data tidak ditemukan”.



Gambar 4.47 Detail Perawatan

4. Gambar 4.47 menampilkan *Activity* detail dari salah satu perawatan yang dipilih.

4.3.12 *Activity* Menu Perkembangan Bayi



Gambar 4.48 Listview Perkembangan Bayi



Gambar 4.49 Detail Perkembangan

1. Gambar 4.48 menampilkan *activity* perkembangan bayi. Pada *Activity* ini *user* dapat melihat berbagai perkembangan bayi dengan cara *scrolling*.

2. Gambar 4.49 menampilkan *activity* detail ketika *user* memilih salah satu info perkembangan bayi tertentu.

4.3.13 *Activity* Menu Info Imunisasi



Gambar 4.50 *Listview* Imunisasi



Gambar 4.51 Detail Imunisasi

1. Gambar 4.50 menampilkan *activity* info imunisasi. Pada *activity* ini, *user* dapat melihat menu imunisasi dengan cara *scrolling*. Jika *user* ingin melihat detail imunisasi berdasarkan usia maka *user* harus menekan salah satu usia.
2. Gambar 4.51 menampilkan *activity* detail info imunisasi.

4.3.14 *Activity* Menu Pilihan Grafik



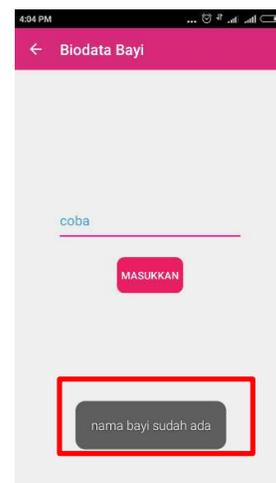
Gambar 4.52 Menu Pilih Grafik

Gambar 4.52 menampilkan *activity* menu pilih grafik. Ketika *user* memilih tombol nama bayi maka *user* akan diarahkan ke *activity* menu nama bayi dan apabila *user* memilih menu masukkan data bayi maka *user* akan diarahkan ke *activity* menu data bayi. Kemudian ketika *user* memilih tombol grafik maka *user* akan diarahkan ke *activity* grafik.

4.3.15 *Activity* Nama Bayi



Gambar 4.53 *Form* Biodata Bayi



Gambar 4.54 Validasi Nama Bayi

1. Gambar 4.53 menampilkan *activity* menu nama bayi. Pada *activity* ini *user* bisa memasukkan nama bayi yang ingin dilihat grafik pertumbuhannya setiap bulan. Jika nama bayi yang dimasukkan belum terdaftar maka akan muncul notifikasi “Berhasil Memasukkan Data”.
2. Gambar 4.54 menampilkan notifikasi bahwa nama bayi sudah terdaftar.

4.3.16 Activity Data Bayi

The screenshot shows a mobile application interface for entering baby data. At the top, there is a pink header with a back arrow and the text 'Data Bayi'. Below the header, there is a dropdown menu labeled 'Pilih nama bayi:' with the value 'coba'. Underneath, there are two input fields: 'Umur Bayi(Dalam Bulan)' and 'Berat Bayi(Dalam Kilogram)'. At the bottom, there is a pink button labeled 'MASUKKAN DATA'.

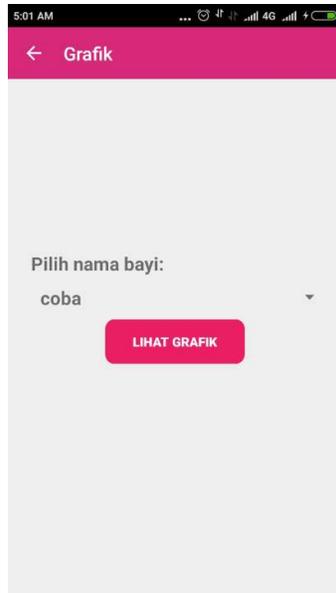
Gambar 4.55 Form Data Bayi

This screenshot shows the same 'Data Bayi' form as in Gambar 4.55, but with the age field filled with '11' and the weight field filled with '6'. A pink button labeled 'MASUKKAN DATA' is visible. Below the button, a grey notification box with a red border contains the text 'Umur yang anda masukkan sudah tercatat', indicating that the age has already been recorded.

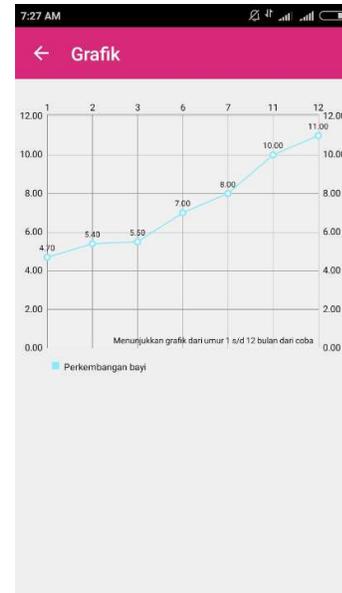
Gambar 4.56 Validasi Umur

1. Gambar 4.55 menampilkan *activity* menu data bayi. Pada *activity* ini *user* memasukkan umur dan berat bayi untuk disimpan agar dapat dilihat grafik pertumbuhannya.
2. Gambar 4.56 menampilkan notifikasi validasi bahwa umur yang dimasukkan oleh *user* sudah pernah dimasukkan.

4.3.17 Activity Menu Grafik



Gambar 4.57 Activity Grafik



Gambar 4.58 Grafik Pertumbuhan

1. Gambar 4.57 menampilkan *activity* menu grafik. Pada *activity* ini *user* dapat memilih nama bayi yang ingin dilihat perkembangannya.
2. Tampilan grafik pertumbuhan seorang bayi dapat dilihat pada Gambar 4.58.

4.3.18 Activity Menu Favorit



Gambar 4.59 Default Activity Favorit



Gambar 4.60 Tampilan Menu Favorit

1. Gambar 4.59 menampilkan *Activity* menu favorit ketika tidak ada resep yang difavoritkan.
2. Tampilan menu favorit ketika sudah ada resep yang difavoritkan bisa dilihat pada Gambar 4.60.

4.3.19 Activity Menu Tentang



Gambar 4.61 Tombol Tentang



Gambar 4.62 Form Tentang

1. Gambar 4.61 menampilkan tombol tentang dan ketika *user* ingin melihat info aplikasi maka *user* harus menekan tombol tentang.
2. Gambar 4.62 menampilkan *dialogform* tentang aplikasi monitoring pertumbuhan dan perkembangan bayi.

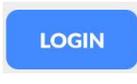
4.4 Pengujian Sistem

Pengujian sistem dilakukan untuk mengevaluasi apakah seluruh fungsi yang ada pada aplikasi sudah berjalan sesuai dengan rancangan.

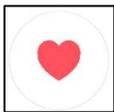
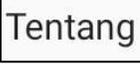
4.4.1 Pengujian *User Interface*

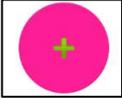
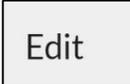
Pengujian *user interface* bertujuan untuk mengetahui fungsionalitas tiap elemen *interface* yang terdapat dalam sistem aplikasi. Elemen yang diujikan yaitu elemen *button* pada setiap *Activity* aplikasi. Hasil pengujian dapat dilihat pada tabel 4.1.

Tabel 4.1 Tabel Pengujian *User Interface*

No.	Kasus yang diuji	Test Case	Hasil yang diharapkan	Hasil yang didapat	Status
1.	Tombol daftar pada <i>Activity</i> registrasi akun		Sistem menampilkan notifikasi berhasil mendaftarkan akun.		Berhasil
2.	Tombol <i>login</i> pada <i>Activity</i> <i>login</i>		Sistem menampilkan <i>Activity</i> menu utama.		Berhasil

No.	Kasus yang diuji	Test Case	Hasil yang diharapkan	Hasil yang didapat	Status
3.	Tombol catatan pada <i>Activity</i> menu utama		Sistem menampilkan menu catatan.		Berhasil
4.	Tombol resep pada <i>Activity</i> menu utama		Sistem menampilkan menu resep.		Berhasil
5.	Tombol kalkulator pada <i>Activity</i> menu utama		Sistem menampilkan menu kalkulator.		Berhasil
6.	Tombol informasi pada <i>Activity</i> menu utama		Sistem menampilkan menu informasi.		Berhasil

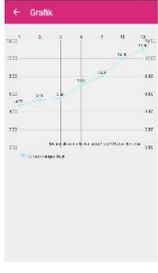
No.	Kasus yang diuji	<i>Test Case</i>	Hasil yang diharapkan	Hasil yang didapat	Status
7.	Tombol grafik pada <i>Activity</i> menu utama		Sistem menampilkan menu grafik.		Berhasil
8.	Tombol favorit pada <i>Activity</i> menu utama		Sistem menampilkan menu favorit.		Berhasil
9.	Tombol Keluar pada <i>Activity</i> menu utama		Sistem menampilkan <i>dialogform</i> konfirmasi		Berhasil
10.	Tombol tentang pada <i>Activity</i> menu utama		Sistem menampilkan <i>Activity</i> tentang.		Berhasil

No.	Kasus yang diuji	<i>Test Case</i>	Hasil yang diharapkan	Hasil yang didapat	Status
11.	Tombol <i>add</i> pada <i>Activity</i> menu catatan		Sistem menampilkan <i>Activity</i> tambah catatan.		Berhasil
12.	Tombol simpan pada <i>Activity</i> menu catatan		Sistem menampilkan notifikasi berhasil disimpan.		Berhasil
14.	Tombol edit pada <i>Activity</i> menu catatan		Sistem menampilkan notifikasi catatan berhasil diperbarui.		Berhasil
15.	Tombol hapus pada <i>Activity</i> menu catatan		Sistem menampilkan notifikasi catatan berhasil dihapus.		Berhasil

No.	Kasus yang diuji	<i>Test Case</i>	Hasil yang diharapkan	Hasil yang didapat	Status
16.	Tombol resep 6 sampai 7 bulan pada <i>Activity</i> menu resep		Sistem menampilkan resep MPASI 6 sampai 7 bulan.		Berhasil
17.	Tombol resep 8 sampai dengan 12 bulan.		Sistem menampilkan resep MPASI 8 sampai 12 bulan.		Berhasil
18.	Tombol favorit pada <i>Activity</i> detail resep.		Sistem menyimpan resep sebagai favorit.		Berhasil
19.	Tombol kalkulator bayi perempuan pada <i>Activity</i> menu kalkulator		Sistem menampilkan <i>Activity</i> kalkulator bayi perempuan.		Berhasil

No.	Kasus yang diuji	Test Case	Hasil yang diharapkan	Hasil yang didapat	Status
20.	Tombol kalkulator bayi laki-laki pada <i>Activity</i> menu kalkulator.	Bayi Laki - laki	Sistem menampilkan <i>Activity</i> kalkulator bayi laki-laki.		Berhasil
21.	Tombol hitung pada <i>Activity</i> kalkulator bayi perempuan dan laki-laki	HITUNG	Sistem menampilkan hasil perhitungan.		Berhasil
22.	Tombol perawatan pada <i>Activity</i> menu informasi bayi	Perawatan	Sistem menampilkan <i>Activity</i> menu perawatan bayi.		Berhasil
23.	Tombol perkembangan pada <i>Activity</i> menu informasi bayi	Perkembangan	Sistem menampilkan <i>Activity</i> menu perkembangan bayi.		Berhasil

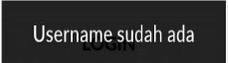
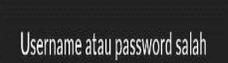
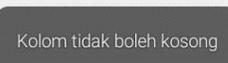
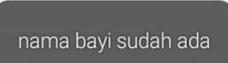
No.	Kasus yang diuji	<i>Test Case</i>	Hasil yang diharapkan	Hasil yang didapat	Status
24.	Tombol imunisasi pada <i>Activity</i> menu informasi bayi.		Sistem menampilkan <i>Activity</i> menu imunisasi bayi.		Berhasil
25.	Tombol nama bayi pada <i>Activity</i> menu grafik		Sistem menampilkan <i>Activity</i> menu nama bayi.		Berhasil
26.	Tombol data bayi pada <i>Activity</i> menu grafik		Sistem menampilkan <i>Activity</i> menu data bayi.		Berhasil
27.	Tombol grafik pada <i>Activity</i> menu grafik		Sistem menampilkan <i>Activity</i> menu grafik bayi.		Berhasil

No.	Kasus yang diuji	Test Case	Hasil yang diharapkan	Hasil yang didapat	Status
28.	Tombol masukkan pada <i>Activity</i> menu nama bayi		Sistem menampilkan notifikasi berhasil memasukkan nama bayi.		Berhasil
29.	Tombol masukkan data pada <i>Activity</i> menu data bayi		Sistem menampilkan notifikasi berhasil memasukkan data bayi.		Berhasil
30.	Tombol lihat grafik pada <i>Activity</i> menu grafik		Sistem menampilkan <i>Activity</i> grafik bayi.		Berhasil

4.4.2 Pengujian Validasi

Pengujian validasi bertujuan untuk mengetahui apakah validasi-validasi yang ada pada system sudah berfungsi dengan baik. Hasil pengujian validasi dapat dilihat pada Tabel 4.3.

Tabel 4.2 Pengujian Validasi

No	Test case	Hasil yang diharapkan	Hasil yang didapat	Status
1.	Validasi <i>username</i> sudah terdaftar	Sistem dapat menampilkan notifikasi bahwa <i>username</i> sudah terdaftar		Berhasil
2.	Validasi <i>username</i> dan <i>password</i> tidak terdaftar	Sistem dapat menampilkan notifikasi bahwa <i>username</i> dan <i>password</i> yang dimasukkan salah.		Berhasil
3.	Validasi kolom kosong	Sistem dapat memberikan info apabila kolom pengisian data kosong.		Berhasil
4.	Validasi <i>password</i> dan konfirmasi <i>password</i> tidak sama.	Sistem dapat menampilkan notifikasi bahwa kolom <i>password</i> dan kolom konfirmasi <i>password</i> tidak sama.		Berhasil
5.	Validasi ketika nama bayi dengan <i>username</i> yang <i>login</i> sudah ada.	Sistem dapat menampilkan notifikasi bahwa nama bayi sudah ada.		Berhasil

No	Test case	Hasil yang diharapkan	Hasil yang didapat	Status
6.	Validasi ketika umur bayi dengan <i>username</i> yang <i>login</i> sudah tercatat.	Sistem dapat menampilkan notifikasi bahwa umur bayi sudah tercatat.		Berhasil