BUKU PETUNJUK PENGGUNAAN

APLIKASI MOTHER CARING

Disusun oleh:

Dr. Anggorowati, S.Kp., M.Kep., Sp.Mat.

Ns. Jumiati Riskiyani Dwi Nandia, S.Kep., M.Kep.

Fatikhu Yatuni Asmara, S.Kp., M.Sc.

Fakultas Kedokteran
Program Studi Magister Keperawatan
Universitas Diponegoro

Semarang

2021

Aplikasi Mother Caring

A. Deskripsi Singkat

Aplikasi Mother Caring ini merupakan aplikasi ibu nifas dan keluarga sebagai media edukasi digital terkait perawatan diri masa nifas. Aplikasi ini dirancang secara lengkap sesuai tahap perkembangan masa nifas dan sesuai kebutuhan ibu nifas dengan fitur konten informasi yang menarik. Materi dalam aplikasi ini disadur dari beberapa referensi jurnal penelitian serta buku pendampingan ibu nifas (Buku KIA).

Aplikasi ini dapat diunduh melalui PlayStore: https://s.id/MotherCaring

B. Petunjuk Penggunaan Aplikasi

1. Splash screen tampilan awal aplikasi

2. Halaman beranda aplikasi

3. Halaman beranda aplikasi (lanjutan)

4. Halaman menu 1 : Konsep nifas

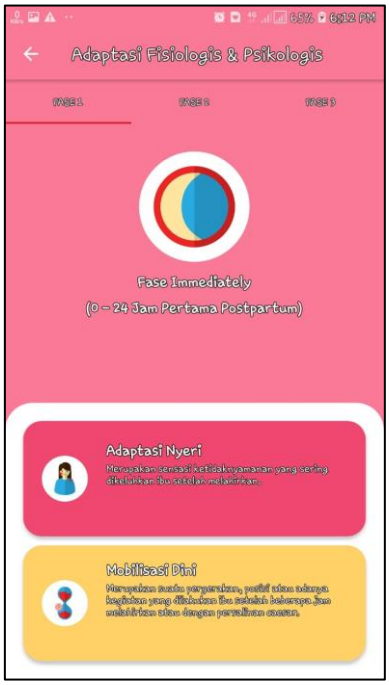
Apa Itu Nifas ?

Nifas merupakan suatu masa segera setelah melahirkan yang diperlukan untuk proses kembalinya alat reproduksi pada keadaan sebelum hamil. Periode nifas merupakan waktu yang sangat rentan bagi seorang wanita setelah melahirkan karena akan melewati peristiwa baru menjadi seorang ibu, memiliki anak, atau bertambahnya anggota baru dalam keluarga.

Pada umumnya masyarakat lebih mengenal periode ini dengan sebutan masa nifas dimana masa tersebut merujuk pada status ibu pada 6 minggu pertama atau 42 hari setelah melahirkan. Waktu nifas selama 42 hari tersebut merupakan batasan maksimum ibu setelah melahirkan disebut ibu nifas. Dalam menjalani masa nifas, ibu nifas

5. Halaman menu 2 : Perubahan fisik dan psikologis

6. Halaman menu 3 : Perawatan diri nifas



7. Halaman menu 4 : Pendampingan keluarga

Pendampingan Keluarga

Proses melahirkan yang menguras tenaga dan emosi membuat sebagian bunda merasa kelelahan,belum lagi proses setelah melahirkan, mulai dari menyusui, menina bobokan bayi, hingga mengganti popok. Tidak ada salahnya jika bunda meminta bantuan baik itu dari keluarga. Orang terdekat seperti suami dan ibu atau ibu mertua yang juga memiliki peran penting untuk mendukung pemulihan kondisi fisik dan mental setelah melahirkan. Alangkah baiknya bunda membangun komunikasi yang baik ke orang-orang terdekat bunda yaitu keluarga.

Setelah pulang dari rumah sakit atau klinik bersalin, ini merupakan waktunya bunda lebih dekat memberikan kasih sayang kepada anggota keluarga baru bunda. Tapi bukan

8. Halaman menu 5 : Tanda bahaya masa nifas

9. Halaman menu 6 : Musik relaksasi nifas

Musik Relaksasi Nifas

Ada beberapa terapi alternatif sebagai terapi relaksasi pada masa nifas, salah satu terapi musik. Terapi musik telah banyak digunakan untuk mengatasi ketegangan emosi yakni serta nyeri selama masa kehamilan hingga masa nifas. Terapi musik berperan sebagai salah satu teknik relaksasi untuk memperbaiki, memelihara, mengem bangkan mental, fisik, dan kesehatan emosi/psikologis sehingga terapi musik ini dapat dilakukan guna membantu mencegah dan mengatasi depresi nifas.

Terapi musik sering digunakan karena sangat mudah dilakukan dan terjangkau, tetapi efeknya menunjukkan betapa besar musik dalam mempengaruhi ketegangan atau kondisi rileks pada diri seseorang. Musik tersebut akan merangsang pengeluaran

10. Halaman menu 7 : Tips kebugaran masa nifas

Tips Kebugaran Masa Nifas

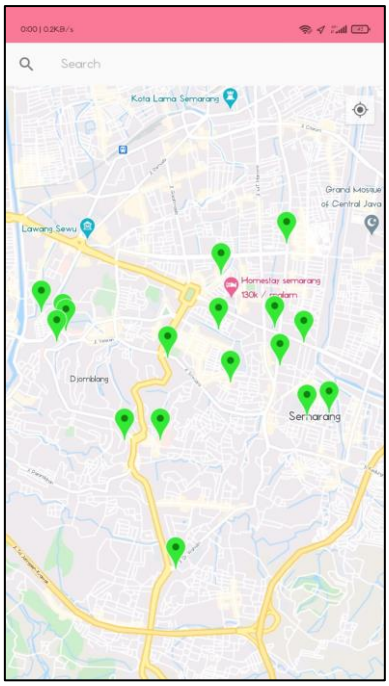
Setelah melahirkan kondisi tubuh bunda akan menurun. Selain itu sistem pencernaan juga melemah, hal tersebut terjadi karena ada ketidakseimbangan energi di dalam tubuh dan pola makan yang tidak teratur. Bahkan secara psikis dan emosional pun bunda juga akan mengalami sejumlah perubahan. Sementara itu bunda juga harus tetap menjaga kesehatan diri. Baik yang melahirkan secara normal maupun lewat operasi caesar. Di mana keduanya membutuhkan perawatan agar tubuh kembali fit.

Berikut 5 Cara Untuk Membuat Tubuh Bunda Fit Setiap Hari

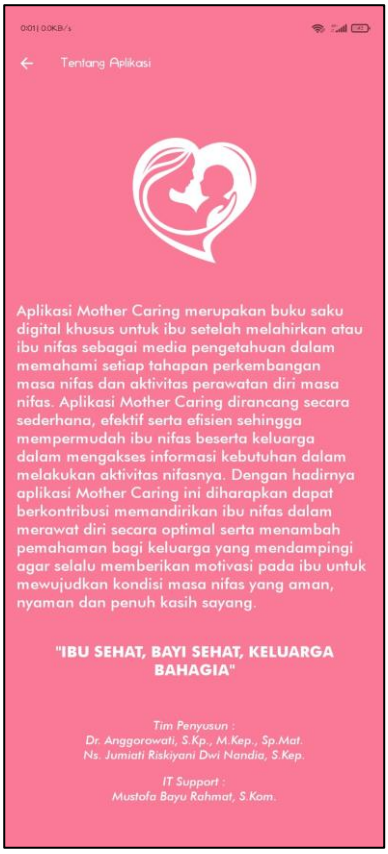
a. Cukupi kebutuhan cairan

11. Halaman menu 8 : Kontak perawat maternitas

12. Halaman menu 9 : Map pelayanan kesehatan



13. Halaman menu 10 : Tentang aplikasi



C. Tabel Use Case Aplikasi

Tabel use case merupakan penjelasan alur penggunaan aplikasi Mother Caring.

No	*Use Case*	Deskripsi
1.	Mengakses Menu Utama	Menampilkan sebelas menu utama aplikasi diantaranya yaitu menu Daftar Layanan Kesehatan, menu *Map View*, menu Konsep *Postpartum*, menu Adaptasi Fisiologis dan Psikologis, menu Materi Edukasi Tahapan *Postparum*, menu *Family Caring*, menu Deteksi Dini Komplikasi *Postpartum*, menu Musik Relaksasi *Postpartum*, menu Tips Kebugaran Masa *Postpartum* dan menu Kontak Perawat Maternitas dan menu Tentang.
2.	Mengakses Menu Daftar Layanan Kesehatan	Menampilkan daftar layanan kesehatan yang ada di daerah Purwokerto dengan fitur Pencarian.
3.	Menu *Map View*	Menampilkan peta, *button* Pencarian Lokasi, *button* Tampilkan Layanan Kesehatan, *button* Satelit.
4.	Mengakses Menu Konsep Postpartum	Menampilkan tiga sub menu penjelasan yaitu sub menu Apa Itu Postpartum, sub menu Sebutan Ibu Postpartum/Ibu Nifas dan sub menu Apa Saja Tahapan Perkembangan Pada Periode Postpartum.
5.	Mengakses Menu Adaptasi Fisiologis dan Psikologis	Menampilkan enam sub menu bagian Fisiologis yaitu sub menu Tanda Vital, sub menu Sistem Perkemihan, sub menu Sistem Endokrin, sub menu Muskulosketal, sub menu Sistem Reproduksi dan sub menu Sistem Pencernaan serta tiga sub menu bagian Psikologis yaitu sub menu Fase *Taking In*, sub menu Fase *Taking Hold* dan sub menu Fase *Letting Go*.
6.	Mengakses Menu Materi Edukasi Tahapan *Postpartum*	Menampilkan tiga bagian yaitu bagian Fase *Immediately* yang terdiri dari sub menu Adaptasi Nyeri dan sub menu Mobilisasi Dini, bagian Fase *Early Postpartum* yang terdiri dari sub menu Perawatan Payudara, sub menu Pijat Oksitoksin, sub menu ASI Ekslusif, sub menu

No	*Use Case*	Deskripsi
		Teknik Menyusui, sub menu Nutrisi Masa Menyusui, sub menu Perawatan Perineum, sub menu *Personal Hygiene,* sub menu Kebutuhan Istirahat, sub menu Senam Nifas dan sub menu Perawatan BBL, bagian Fase *Late Postpartum* yang terdiri dari sub menu Kontrasepsi, sub menu Seksualitas, sub menu Imunisasi Bayi dan sub menu Mengenal Perilaku Bayi.
7.	Mengakses Menu *Family Caring*	Menampilkan konten informasi tentang peran keluarga dalam membantu ibu nifas.
8.	Mengakses Menu Deteksi Dini Komplikasi *Postpartum*	Menampilkan lima sub menu yaitu sub menu Pendarahan Pervaginam, sub menu Infeksi Masa Nifas, sub menu Kelainan Pada Payudara, sub menu Hipertensi *Postpartum*, sub menu Anemia *Postpartum*.
9.	Mengakses Menu Musik Relaksasi *Postpartum*	Menampilkan menu yang berisi informasi penjelasan tentang fungsi musik dalam masa *postpartum* dan sub menu Pemutar Musik.
10.	Mengakses Menu Tips Kebugaran Masa *Postpartum*	Menampilkan menu yang berisi informasi tentang tips dalam menjaga tubuh tetap bugar selama masa nifas.
11.	Mengakses Menu Kontak Perawat Maternitas	Menampilkan daftar kontak perawat maternitas yang bisa dihubungi untuk berkonsultasi.
12.	Mengakses Menu Tentang	Menampilkan menu yang berisi informasi tentang aplikasi.

D. Source Code

MapsActivity

```
package com.bayu007.nifas.Menu;

import ...

public class MapsActivity extends FragmentActivity implements
        OnMapReadyCallback {

    private static final int MY_PERMISSION_CODE = 1000;
    private static final long UPDATE_INTERVAL = 500;
    private static final float DEFAULT_ZOOM = 9.5f;

    FloatingActionButton satellite;
    Button button;
    private GoogleMap mMap;
    private Location lastlocation;
    private Marker currentUserLocationMarker;
    private double latitude, longitude;
    private int ProximityRadius = 10000;
    private SupportMapFragment mapFragment;
    private LatLng sydney = new LatLng( v: -7.4272498, v1: 109.2106843);

    IGoogleAPIService mService;

    MyPlaces currentPlace;

    FusedLocationProviderClient fusedLocationProviderClient;
    LocationCallback locationCallback;
    private LocationRequest locationRequest;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_maps);

        // Obtain the SupportMapFragment and get notified when the map is ready to be used.
        mapFragment = (SupportMapFragment) getSupportFragmentManager()
                .findFragmentById(R.id.map);
        mapFragment.getMapAsync( onMapReadyCallback: this);
        mService = Common.getGoogleAPIService();

        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M)
        {
            checkUserLocationPermission();
        }

        if (!Connections.checkConnection( context: this)) {
            Toast.makeText( context: this,  text: "Kesalahan jaringan periksa koneksi anda",
                    Toast.LENGTH_SHORT).show();
            finish();
        }

        buildLocationCallBack();
        buildLocationRequest();

        fusedLocationProviderClient = LocationServices.getFusedLocationProviderClient( activity: this);
        fusedLocationProviderClient.requestLocationUpdates(locationRequest, locationCallback,
                Looper.myLooper());

        button = findViewById(R.id.fbhospital);
        satellite = findViewById(R.id.fbsatelit);

        button.setOnClickListener((v) -> { nearByPlace( placeType: "hospital"); });
        satellite.setOnClickListener((v) -> {
                if (mMap != null) {
                    int MapType = mMap.getMapType();
                    if (MapType == 1) {
```

```
                    satellite.setImageResource(R.drawable.ic_satellite_off);
                    mMap.setMapType(GoogleMap.MAP_TYPE_SATELLITE);
                } else {
                    satellite.setImageResource(R.drawable.ic_satellite_on);
                    mMap.setMapType(GoogleMap.MAP_TYPE_NORMAL);
                }
            }
        });

        setupAutoCompleteFragment();
    }

    private void setupAutoCompleteFragment() {
        PlaceAutocompleteFragment autocompleteFragment = (PlaceAutocompleteFragment)
                getFragmentManager().findFragmentById(R.id.place_autocomplete_fragment);
        autocompleteFragment.setOnPlaceSelectedListener(new PlaceSelectionListener() {
            @Override
            public void onPlaceSelected(Place place) {
                sydney = place.getLatLng();
                mapFragment.getMapAsync( onMapReadyCallback: MapsActivity.this);

                String url = getUrl(sydney.latitude, sydney.longitude,  nearbyPlace: "");
                Object[] DataTransfer = new Object[2];
                DataTransfer[0] = mMap;
                DataTransfer[1] = url;
                Log.d( tag: "onClick", url);
                GetNearbyPlacesData getNearbyPlacesData = new GetNearbyPlacesData();
                getNearbyPlacesData.execute(DataTransfer);
            }

            @Override
            public void onError(Status status) { Log.e( tag: "Error", status.getStatusMessage()); }
        });
    }

    @Override
    protected void onStop() {
        fusedLocationProviderClient.removeLocationUpdates(locationCallback);
        super.onStop();
    }

    private void buildLocationRequest()
    {
        locationRequest = new LocationRequest();
        locationRequest.setInterval(1100);
        locationRequest.setFastestInterval(1100);
        locationRequest.setSmallestDisplacement(10f);
        locationRequest.setPriority(LocationRequest.PRIORITY_BALANCED_POWER_ACCURACY);
    }

    private void buildLocationCallBack()
    {
        locationCallback = (LocationCallback) onLocationResult(locationResult) -> {
                super.onLocationResult(locationResult);
                lastlocation = locationResult.getLastLocation();

                if (currentUserLocationMarker != null)
                    currentUserLocationMarker.remove();

                latitude = lastlocation.getLatitude();
                longitude = lastlocation.getLongitude();

                LatLng latLng = new LatLng(latitude, longitude);

                MarkerOptions markerOptions = new MarkerOptions();
                markerOptions.position(latLng);
```

```
                markerOptions.snippet("Lokasi Saya");
                markerOptions.icon(BitmapDescriptorFactory
                        .defaultMarker(BitmapDescriptorFactory.HUE_ROSE));

                currentUserLocationMarker = mMap.addMarker(markerOptions);

                mMap.moveCamera(CameraUpdateFactory.newLatLng(latLng));
                mMap.animateCamera(CameraUpdateFactory.zoomBy( v: 10));
            };
        }

        private void nearByPlace(final String placeType)
        {
            mMap.clear();
            String url = getUrl(latitude, longitude, placeType);

            mService.getNearByPlaces(url)
                    .enqueue(new Callback<MyPlaces>() {
                        @Override
                        public void onResponse(Call<MyPlaces> call, Response<MyPlaces> response) {

                            currentPlace = response.body();

                            if (response.isSuccessful())
                            {
                                for (int i=0;i<response.body().getResults().length;i++)
                                {
                                    MarkerOptions markerOptions = new MarkerOptions();
                                    Results googlePlace = response.body().getResults()[i];
                                    double lat = Double.parseDouble
                                            (googlePlace.getGeometry().getLocation().getLat());
                                    double lng = Double.parseDouble
                                            (googlePlace.getGeometry().getLocation().getLng());
                                    String placeName = googlePlace.getName();
                                    String vicinity = googlePlace.getVicinity();
                                    LatLng latLng = new LatLng(lat, lng);
                                    markerOptions.position(latLng);
                                    markerOptions.title(placeName);
                                    markerOptions.icon(BitmapDescriptorFactory
                                            .defaultMarker(BitmapDescriptorFactory.HUE_GREEN));
                                    markerOptions.snippet(String.valueOf(i));
                                    mMap.addMarker(markerOptions);
                                    mMap.moveCamera(CameraUpdateFactory.newLatLng(latLng));
                                    mMap.animateCamera(CameraUpdateFactory.zoomBy( v: 14));
                                }
                            }
                        }

                        @Override
                        public void onFailure(Call<MyPlaces> call, Throwable t) {

                        }
                    });
        }

        private String getUrl(double latitude, double longitude, String nearbyPlace)
        {
            StringBuilder googleURL = new
                    StringBuilder("https://maps.googleapis.com/maps/api/place/nearbysearch/json?");
            googleURL.append("location=" + latitude + "," + longitude);
            googleURL.append("&radius=" + ProximityRadius);
            googleURL.append("&type=" + nearbyPlace);
            googleURL.append("&sensor=true");
            googleURL.append("&key=" + "AIzaSyAWF9Oansk9suI-g84Fq0mclWdrfjBYePA");

            Log.d( tag: "getUrl", googleURL.toString());
```

```
265
266             return (googleURL.toString());
267         }
268
269         @Override
270         public void onMapReady(GoogleMap googleMap) {
271             mMap = googleMap;
272
273             mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(sydney, DEFAULT_ZOOM));
274             mMap.getUiSettings().setZoomControlsEnabled(false);
275
276             if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
277                 if (ContextCompat.checkSelfPermission( context: this,
278                         Manifest.permission.ACCESS_FINE_LOCATION) == PackageManager.PERMISSION_GRANTED)
279                 {
280
281                     mMap.setMyLocationEnabled(true);
282                 }
283             }
284             else
285             {
286
287                 //mMap.setMyLocationEnabled(true);
288             }
289
290             mMap.setOnMarkerClickListener((marker) -> {
293                     if (marker.getSnippet() != null) {
294                         Common.currentResult = currentPlace.getResults()
295                                 [Integer.parseInt(marker.getSnippet())];
296                         startActivity(new Intent( packageContext: MapsActivity.this, ViewPlace.class));
297                     }
298                     return true;
299             });
```

```
301         }
302
303         private boolean checkUserLocationPermission()     {
304             if (ContextCompat.checkSelfPermission( context: this,
305                     Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED)
306             {
307                 if (ActivityCompat.shouldShowRequestPermissionRationale( activity: this,
308                         Manifest.permission.ACCESS_FINE_LOCATION))
309                     ActivityCompat.requestPermissions( activity: this, new String[]{
310
311                             Manifest.permission.ACCESS_FINE_LOCATION
312                     }, MY_PERMISSION_CODE);
313                 else
314                     ActivityCompat.requestPermissions( activity: this, new String[]{
315
316                             Manifest.permission.ACCESS_FINE_LOCATION
317                     }, MY_PERMISSION_CODE);
318                 return false;
319             }
320             else
321                 return true;
322         }
323
324         @Override
325         public void onRequestPermissionsResult(int requestCode,
326                                                @NonNull String[] permissions,
327                                                @NonNull int[] grantResults)
328         {
329             switch (requestCode)
330             {
331                 case MY_PERMISSION_CODE:
332                     if (grantResults.length > 0 && grantResults[0] == PackageManager.PERMISSION_GRANTED)
333                     {
```

```
                    if (ContextCompat.checkSelfPermission( context: this,
                            Manifest.permission.ACCESS_FINE_LOCATION) ==
                            PackageManager.PERMISSION_GRANTED)
                    {
                        mMap.setMyLocationEnabled(true);

                        buildLocationCallBack();
                        buildLocationRequest();

                        fusedLocationProviderClient = LocationServices
                                .getFusedLocationProviderClient( activity: this);
                        fusedLocationProviderClient.requestLocationUpdates
                                (locationRequest, locationCallback, Looper.myLooper());
                    }
                }break;
            }
        }

        @Override
        protected void onStart() {
            super.onStart();
            if (Connections.checkConnection( context: this)) {
                new PermissionGPS( activity: this);
            }
        }

        @Override
        protected void onRestart() {
            super.onRestart();
            if (Connections.checkConnection( context: this)) {
                new PermissionGPS( activity: this);
            }
        }

        @Override
        protected void onResume() {
            super.onResume();
            if (mMap != null) {
                mMap.clear();
            }
        }

    }
```