BAB III
METODOLOGI

3.1 Metode Perancangan *Database*

Dalam penelitian ini metode yang digunakan dalam melakukan perancangan *database* meliputi 4 tahapan yaitu:

1. Analisis dan Pengumpulan Data

   Analisis dan engumpulan data bertujuan untuk mengumpulkan dan menganalisa data atau informasi apa saja yang dibutuhkan di dalam sistem.
2. Perancangan *Database Level* Konseptual

   Perancangan *database level* konseptual bertujuan untuk mengecek kebutuhan pengguna, batasan-batasan serta hubungannya.
3. Perancangan *Database Level* logikal

   Perancangan database *level* logikal bertujuan untu memetakan rancangan konseptual ke dalam model database yang akan digunakan.
4. Perancangan *Database Level* Fisikal

   Perancangan *database level* fisikal bertujuan untuk mengimplementasikan hasil dari rancangan *level* konseptual dan *level* logikal untuk mendapatkan rancangan *database* yang akan digunakan.

3.2 Analisis dan Pengumpulan Data

3.2.1 Kuisioner

Teknik yang digunakan penulis dalam melakukan analisis dan pengumpulan data atau informasi yang dibutuhkan adalah dengan kuisioner. Dalam penelitian ini kuisioner disajikan dalam bentuk pertanyaan berupa pilihan ganda, sehingga mempermudah responden untuk mengisi kuisioner tersebut, kuisioner disebar kepada para ibu sebagai responden. Data dari hasil kuesioner dapat dilihat di lampiran 1. Sehingga data yang diperoleh dapat membantu dalam pembuatan *database* pada aplikasi *Monitoring* pertumbuhan dan perkembangan bayi.

3.2.2 Penelitian

Selanjutnya teknik yang digunakan penulis dalam melakukan analisis dan pengumpulan data atau informasi yang dibutuhkan adalah teknik penelitian, yaitu mencari informasi yang berhubungan menggunakan jurnal-jurnal, buku referensi dan sumber internet. Seperti buku resep MPASI, informasi pertumbuhan dan perkembangan bayi, *website* yang merujuk pada IDAI (Ikatan Dokter Anak Indonesia), dan sebagainya. Setelah melakukan pencarian dan pengumpulan data kemudian dilakukan analisis pada data dan informasi tersebut yang bertujuan untuk menentukan data dan informasi yang tepat untuk dimasukkan ke dalam *database.*

3.3 Perancangan *Database Level* Konseptual

Tahap selanjutnya yang dilakukan dalam melakukan perancangan sebuah *database* adalah metode perancangan *database level* konseptual, perancangan *level* konseptual ini bertujuan untuk mengecek kebutuhan-kebutuhan *user,* Batasan-batasan dan hubungannya. Perancangan skema konseptual ini biasa menggunakan model *ERD (Entity Relationshiop Diagram).*

3.3.1 *Entity* dan *Attribute*

Entity dan *attribute* dibuat terlebih dahulu sebelum membuat ERD *(Enity Relationship Diagram). entity* dan *attribute* ini didapatkan dari hasil analisis dan pengumpulan data. *entity* dan *attribute* tersebut terdiri dari *entity* grafik pertumbuhan, *entity* resep MPASI, *entity* info perkembangan bayi, *entity* info jadwa imunisasi, *entity* info perawtan, *entity* catatan, *entity* bayi, *entity member*, dan *entity* resep *favorite*.

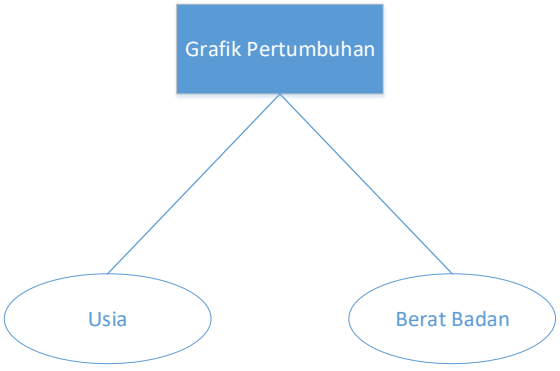
1. *Entity* Grafik Pertumbuhan

Entity grafik pertumbuhan adalah *entity* yang menunjukan data grafik pertumbuhan bayi, berikut adalah gambar yang menunjukan grafik pertumbuhan bayi:

Gambar 3.1 Data Grafik Pertumbuhan Bayi

Sumber: https://www.slideshare.net/Abuwa_Maulidya/gizi-pertumbuhan-perkembangan-balita.

Berdasarkan pada grafik pertumbuhan di atas maka diperoleh data yang dibutuhkan untuk menentukan *entity* dan *attribute* tersebut, yaitu grafik pertumbuhan sebagai nama *entity*. Dan data usia serta berat badan sebagai *attribute* nya. *entity* dan *attribute* tersebut dapa dilihat pada gambar berikut:



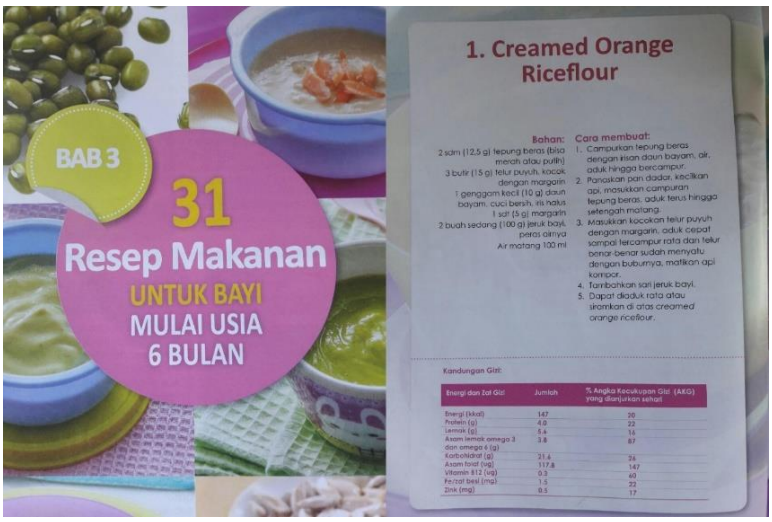
Gambar 3.2 *Entity* Grafik Pertumbuhan

Penjelasan *entity* grafik pertumbuhan pada gambar 3.2 adalah sebagai berikut:

Pada *attribute* usia perlu dimasukkan ke dalam *entity* grafik pertumbuhan Karena berfungsi untuk mengetahui usia bayi. Pada *attribute* berat badan perlu dimasukkan ke dalam *entity* grafik pertumbuhan Karena berfungsi untuk menyimpan data berat badan bayi.

2. *Entity* Resep MPASI

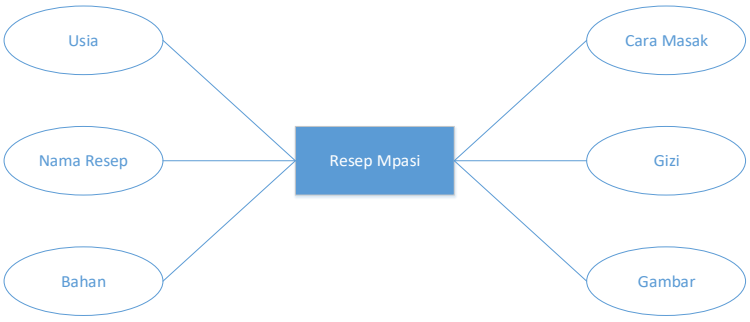
Entity resep MPASI adalah *entity* yang menunjukan data resep MPASI. Berikut adalah gambar yang menunjukan data resep MPASI:



Gambar 3.3 Data Resep MPASI

Sumber: Buku resep MPASI untuk anak 6-12 bulan

Berdasarkan pada gambar di atas, maka diperoleh data yang dibutuhkan untuk menentukan *entity* dan *attribute* tersebut. Yaitu Resep MPASI sebagai nama *entity*. Dan usia, nama resep, bahan, cara masak, gizi, beserta gambar sebagai *attribute* nya. *entity dan attribute* tersebut dapat dilihat pada gamabar berikut:



Gambar 3.4 *Entity* Resep MPASI

Penjelasan *entity* resep MPASI pada gambar 3.4 adalah sebagai berikut:

Pada *attribute* usia perlu dimasukkan kedalam *entity* resep makan Karena berfungsi untuk mengetahui resep tersebut diperuntukan bayi pada usia berapa bulan, pada *attribute* nama_makanan, bahan, cara_masak, dan gizi perlu dimasukkan ke dalam *entity* resep_makanan Karena berfungsi untuk mengetahui nama makanan, bahan-bahan, cara masak serta kandungan gizi yang terdapat pada resep tersebut, pada *attribute* gambar perlu dimasukkan ke dalam *entity* resep karena berfungsi untuk menyimpan data gambar dari resep MPASI.

3. *Entity* Info Perkemabangan Bayi

Entity info perkembangan bayi adalah *entity* yang menunjukan data perkembangan bayi, berikut adalah gambar tabel yang menunjukan data perkembangan bayi:

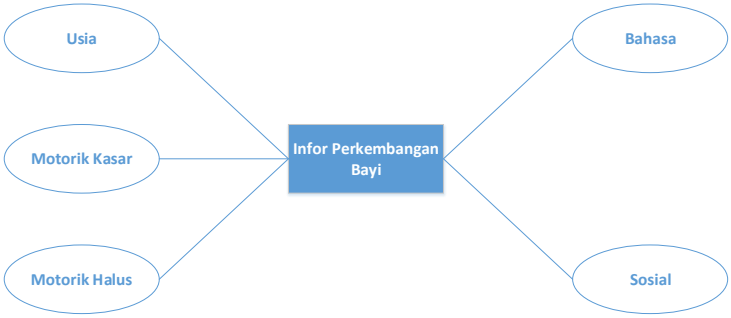
TAHAPAN PERKEMBANGAN BAYI TAHUN PERTAMA

Bulan	Motorik Kasar	Motorik Halus	Bahasa-Kognitif	Sosial
Bulan 1	Menggerak-gerakkan kepala dari sisi ke sisi	Dapat menggenggam dengan erat	Senang menatap tangan dan jemari	Menaksir gerakan dengan mata
Bulan 2	Menahan kepala dan leher saat tengkurap	Membuka dan menutup tangan	Mulai senang bermain dengan jari	Merespon dengan senyum
Bulan 3	Menyentuh dan menangkap benda	Menggenggam benda dengan tangannya	Mengoceh pelan	Menirukan ketika Anda menjulurkan lidah

Gambar 3.5 Data Perkembangan Bayi

Sumber: http://www.gambarkatakata.xyz/tahapan-perkembangan-bayi-tahun-pertama-rumah-bunda/.

Berdasarkan pada gambar tabel perkembangan bayi di atas maka diperoleh data yang dibutuhkan untuk menentukan *entity* dan *attribute* tersebut, yaitu perkembangan bayi sebagai nama *entity*, dan data umur, mototrik kasar, motorik halus, bahasa, beserta sosial sebagai *attribute* nya. *entity* dan *attribute* tersebut dapat dilihat pada gambar berikut:



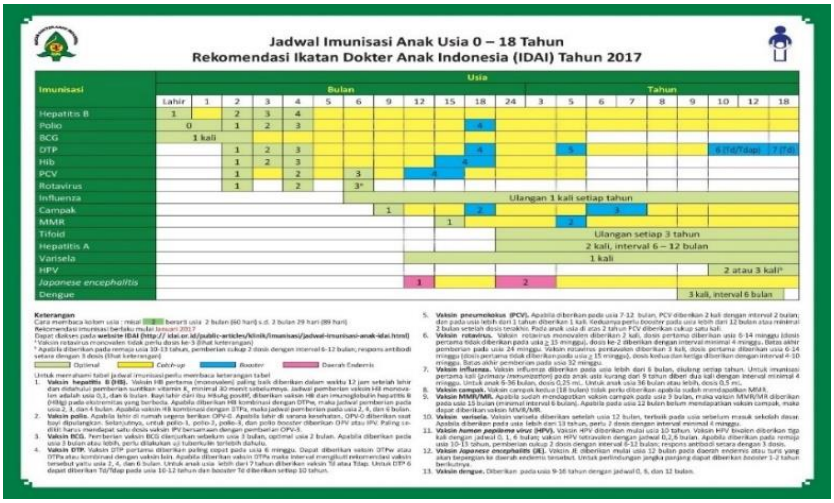
Gambar 3.6 *Entity* Perkembangan Bayi

Penjelasan *entity* info perkembangan bayi pada gambar 3.6 adalah sebagai berikut:

Pada *attribute* umur perlu dimasukkan ke dalam *Entity* perkembangan bayi Karena berfungsi untuk mengetahui umur dari bayi. Pada *attribute* motorik kasar dan motorik halus perlu dimasukkan ke dalam *Entity* perkembangan bayi yang berfungsi untuk mengetahui perkembangan motorik bayi, pada *attribute* bahasa dan sosial perlu dimasukkan ke dalam *Entity* perkembangan bayi yang berfungsi untuk mengetahui perkembangan komunikasi dan perilaku bayi berdasarkan usia.

4. *Entity* Info Jadwal Imunisasi

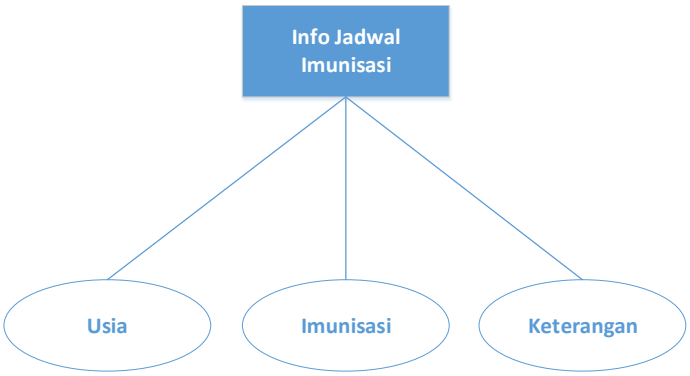
Entity info jadwal imunisasi adalah *entity* yang menunjukan informasi jadwal imunisasi, berikut adalah gambar yang menunjukan tabel informasi jadwal imunisasi:



Gambar 3.7 Data Jadwal Imunisasi

Sumber: http://www.idai.or.id/artikel/klinik/imunisasi/jadwal-imunisasi-2017.

Berdasarkan pada tabel jadwal imunisasi yang ditunjukan pada gambar di atas maka diperoleh data yang dibutuhkan untuk menentukan *entity* dan *attribute* tersebut, yaitu jadwal imunisasi sebagai nama *entity*, dan data jenis imunisasi, usia, serta keterangan sebagai *attribute* nya. *entity* dan *attribute* tersebut dapat dilihat pada gambar berikut:



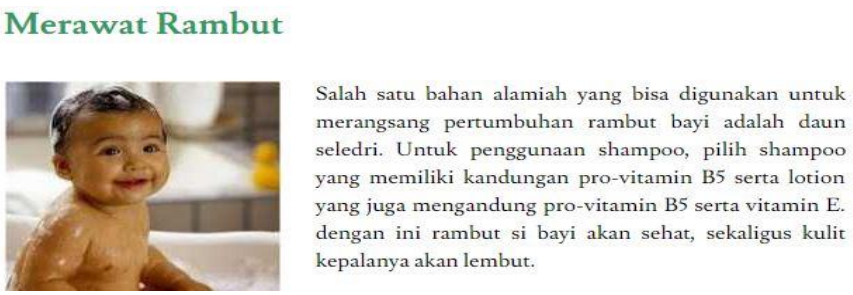
Gambar 3.8 *Entity* jadwal imunisasi

Penjelasan *entity* jadwal imunisasi pada gambar 3.8 adalah sebagai berikut:

Pada *attribute* imunisasi perlu dimasukkan ke dalam *entity* jadwal imunisasi karena berfungsi untuk mengetahui jenis imunisasi apa yang harus diberikan untuk bayi sesuai dengan usia nya, pada *attribute* usia perlu dimasukkan ke dalam *entity* jadwal imunisasi karena befungsi untuk menunjukan usia bayi. pada *attribute* keterangan perlu dimasukkan ke dalam *entity* jadwal imunisasi karena berfungsi untuk mengetahui informasi lebih terkait jenis imunisasi.

5. *Entity* Info Perawatan Bayi

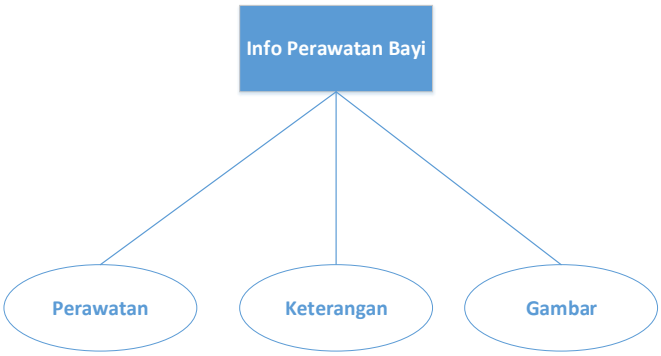
Entity Perawatan adalah *entity* yang menunjukan informasi terkait perawatan bayi, berikut adalah gambar yang menunjukan informasi perawatan bayi:



Gambar 3.9 Data Perawatan Bayi

Sumber: http://paketmadupenyuburkandunganalmabruroh.my.id/tips-merawat-bayi/yang-baru-lahir/.

Berdasarkan pada gambar di atas maka diperoleh data yang diperlukan untuk menentukan *entity* dan *attribute* tersebut, yaitu perawatan bayi sebagai nama *entity*, dan data jenis perawatan serta keterangan dan gambar sebagai *attribute* nya. *entity dan attribute* tersebut padat dilihat pada gambar berikut:



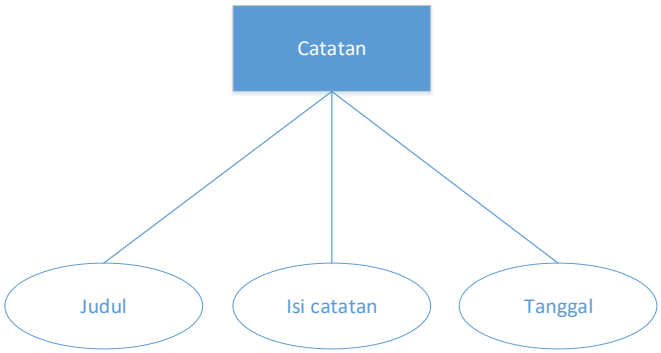
Gambar 3.10 *Entity* Perawatan Bayi

Penjelasan *entity perawatan* pada gambar 3.10 adalah sebagai berikut:

Pada *attribute* jenis perawatan perlu dimasukkan ke dalam *entity* tips Karena berfungsi untuk mengetahui perawatan apa saja yang harus dilakukan untuk merawat bayi, pada *attribute* keterangan perlu dimasukkan ke dalam *entity tips* Karena berfungsi untuk mengetahui informasi dan cara untuk melakukan perawatan tersebut. Pada *attribute* gambar perlu dimasukkan ke dalam *entity* perawatan bayi karena berfungsi untuk menyimpan gambar perawatan.

6. *Entity* Catatan

Entity Catatan adalah *entity* yang menunjukan catatan, fungsi dari catatan adalah untuk membuat catatan pribadi bagi pengguna dan juga dapat memudahkan pengguna dalam menyimpan informasi yang dibutuhkan, dalam *entity* catatan tersebut penulis menambahkan 3 *attribute* yaitu *attribute* judul, isi_catatan, dan tanggal. *Attribute* tersebut dapat dilihat pada gambar berikut:



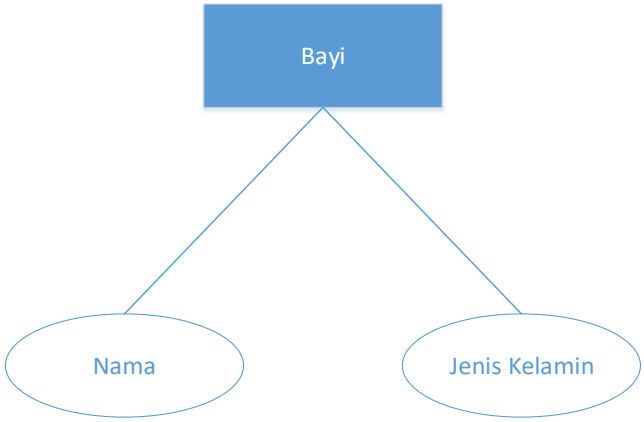
Gambar 3.11 *Entity* Catatan

Penjelasan *entity* catatan pada gambar 3.11 adalah sebagai berikut:

pada *attribute* judul perlu dimasukkan ke dalam *entity* catatan Karena berfungsi untuk memberi judul pada catatan yang dibuat. Pada *attribute* isi_catatan perlu dimasukkan ke dalam *entity* catatan Karena berfungsi sebagai tempat menyimpan catatan. Pada *attribute* tanggal perlu dimasukkan ke dalam *entity* catatan karena berfungsi untuk memberikan tanggal pada catatan.

7. *Entity* Bayi

Entity Bayi adalah *entity* yang menunjukan data bayi, *entity* bayi perlu ditambahkan karena data *bayi* ini akan digunakan pada saat membuat grafik pertumbuhan, dalam *entity* bayi penulis menambahkan 2 *attribute* yang dibutuhkan yaitu nama, dan jenis kelamin. *Attribute* tersebut dapat dilihat pada gambar berikut:



Gambar 3.12 *Entity* Bayi

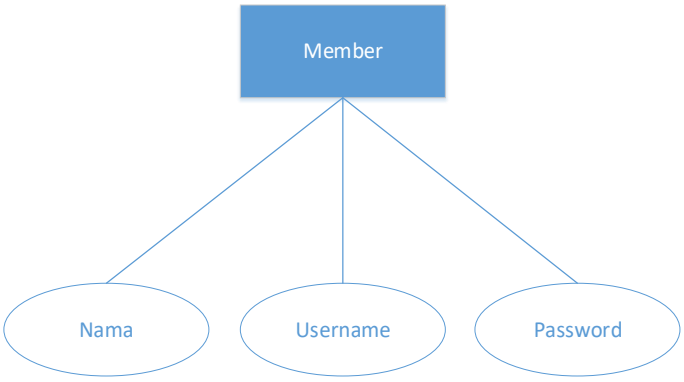
Penjelasan *entity* bayi pada gambar 3.12 adalah sebagai berikut:

pada *attribute* nama perlu dimasukkan ke dalam *Entity* bayi Karena berfungsi untuk mengetahui nama bayi, pada *attribute* jenis kelamin perlu dimasukkan ke dalam *entity* bayi Karena berfungsi untuk mengetahui jenis kelamin bayi.

8. *Entity Member*

Entity Member adalah *entity* yang menunjukan data pengguna aplikasi, *Entity member* perlu ditambahkan karena data *member* ini yang akan digunakan oleh *member* ketika mengakses aplikasi, dalam *entity member* penulis

menambahkan 3 *attribute* yang dibutuhkan yaitu nama, *username*, dan *password*. *Attribute* tersebut dapat dilihat pada gambar 3.13 berikut:



Gambar 3.13 *Entity Member*

Penjelasan *entity member* pada gambar 3.13 adalah sebagai berikut:

pada *attribute* nama perlu dimasukkan ke dalam *entity member* Karena berfungsi untuk mengetahui nama *member*. Sedangkan pada *attribute username* dan *password* perlu dimasukkan ke dalam *entity member* karena berfungsi untuk login *member*.

3.3.2 Primary Key

Setelah menentukan *entity* dan *attribute* langkah selanjutnya adalah menentukan *primary key* pada tiap *entity* tersebut, *primary key* berfungsi untuk memastikan bahwa setiap baris data pada tabel bersifat unik (berbeda antara baris satu dengan baris yang lain).

1. *Primary Key Entity* Grafik Pertumbuhan

Pada *entity* grafik pertumbuhan yang ditunjukkan pada gambar 3.2, terdapat *attribute* sebagai berikut:

- Grafik Pertumbuhan (usia, berat badan)

Berdasarkan pada daftar *attribute* di atas, tidak ada *attribute* yang bersifat unik, maka perlu ditambahkan satu *attribute* lagi dengan nama id_grafik, *attribute* id_grafik ini yang akan menjadi *primary key* dari *entity* grafik pertumbuhan.

- Grafik Pertumbuhan (**id_grafik,** usia, berat badan)

2. *Primary Key Entity* Resep MPASI

Pada *entity* resep MPASI yang ditunjukkan pada gambar 3.4, terdapat *attribute* sebagai berikut:

- Resep MPASI (usia, nama_resep, bahan, cara_masak, gizi, gambar)

Berdasarkan pada daftar *attribute* di atas, tidak ada *attribute* yang bersifat unik, maka perlu ditambahkan satu *attribute* lagi dengan nama id_resep, *attribute* id_resep ini yang akan menjadi *primary key* dari *entity* resep MPASI.

- Resep MPASI (**id_resep**, usia, nama_resep, bahan, cara_masak, gizi, gambar)

3. *Primary Key Entity* Info Perkembangan Bayi

Pada *entity* info perkembangan bayi yang ditunjukkan pada gambar 3.6, terdapat *attribute* sebagai berikut:

- Info Perkembangan Bayi (Umur, motorik kasar, motorik halus, bahasa, Sosial)

Berdasarkan pada daftar *attribute* di atas, pada *attribute* umur bersifat unik karena setiap umur mempunyai perkembangan yang berbeda. Oleh karena itu *attribute* umur akan menjadi *primary key* pada *entity* perkembangan bayi.

- Info Perkembangan Bayi (**Umur**, motorik kasar, motorik halus, bahasa, Sosial)

4. *Primary Key Entity* Info Jadwal Imunisasi

Pada *entity* info jadwal imunisasi yang ditunjukkan pada gambar 3.8, terdapat *attribute* sebagai berikut:

- Info Jadwal Imunisasi (usia, imunisasi, keterangan)

berdasarkan pada daftar *attribute* di atas, tidak ada *attribute* yang bersifat unik, maka perlu ditambahkan satu *attribute* lagi dengan nama id_jadwal, *attribute* id_jadwal ini yang akan menjadi *primary key* dari *entity* jadwal imunisasi.

- Info Jadwal Imunisasi (**id_jadwal**, usia, imunisasi, keterangan)

5. *Primary Key Entity* Info Perawatan Bayi

Pada *entity* info perawatan bayi yang ditunjukkan pada gambar 3.10, terdapat *attribute* sebagai berikut:

- Info Perawatan Bayi (jenis perawtan, keterangan, gambar)

berdasarkan pada daftar *attribute* di atas, tidak ada *attribute* yang bersifat unik, maka perlu ditambahkan satu *attribute* lagi dengan nama id_perawatan, *attribute* id_perawatan ini yang akan menjadi *primary key* dari *entity* perawatan.

- Info Perawatan Bayi (**id_perawatan**, jenis perawtan, keterangan, gambar)

6. *Primary Key Entity* Catatan

Pada *entity* catatan yang ditunjukkan pada gambar 3.11, terdapat *attribute* sebagai berikut:

- Catatan (judul, isi catatan, tanggal)

Berdasarkan pada daftar *attribute* di atas, tidak ada *attribute* yang bersifat unik, maka perlu ditambahkan satu *attribute* lagi dengan nama id_catatan, *attribute* id_catatan ini yang akan menjadi *primary key* dari *entity* catatan.

- Catatan (**id_catatan**, judul, isi catatan, tanggal)

7. *Primary Key Entity* Bayi

Pada *entity* bayi yang ditunjukkan pada gambar 3.12, terdapat *attribute* sebagai berikut:

- Bayi (nama, jenis kelamin)

berdasarkan pada daftar *attribute* di atas, tidak ada *attribute* yang bersifat unik, maka perlu ditambahkan satu *attribute* lagi dengan nama id_bayi, *attribute* id_bayi ini yang akan menjadi *primary key* dari *entity* bayi.

- Bayi (**id_bayi**. nama, jenis kelamin)

8. *Primary Key Entity Member*

Pada *entity member* yang ditunjukkan pada gambar 3.13, terdapat *attribute* sebagai berikut:

- *Member* (nama, *username*, *password*)

berdasarkan pada daftar *attribute* di atas, *attribute username* dapat dijadikan *primary key* karena *atribute username* bersifat unik atau tidak boleh ada *member* yang mempunyai *username* yang sama. maka perlu ditambahkan satu *attribute* lagi dengan nama *id_member*, *attribute id_member* ini yang akan menjadi *primary key* dari *entity member*.

- *Member* (nama, ***username***, *password*)

3.3.3 Sist**em yang Berlaku**

Setelah menentukan *entity*, *attribut*, dan *primary key*, langkah selanjutnya merancang sistem yang akan diberlakukan dengan mengidentifikasi relasi (hubungan) antar *entity*, bisa saja antar satu *entity* dengan *entity* yang lain tidak saling berhubungan, tapi bisa juga antar *entity* dengan *entity* lain saling berhubungan sesuai dengan kebutuhan.

Berikut adalah sistem yang akan diberlakukan pada perancangan *database* aplikasi *monitoring* pertumbuhan dan kecerdasan bayi, yaitu:

1. *Entity Member* dengan *Entity* bayi
   a. *Member* dapat memiliki satu atau lebih dari satu bayi.
   b. *Member* dapat menambah, merubah dan menghapus data bayi yang dimiliki.
2. *Entity* bayi dengan *Entity* grafik pertumbuhan
   a. *Member* dapat membuat satu atau lebih dari satu grafik pertumbuhan sesuai dengan jumlah bayi yang dimiliki.
   b. Jika *member* tidak memiliki bayi maka *member* tidak dapat membuat grafik pertumbuhan.

3. *Entity member* dengan *Entity* catatan
   a. *Member* dapat membuat satu atau lebih dari satu catatan
   b. *Member* dapat menambahkan judul, isi catatan, dan tanggal dalam catatan.
   c. *Member* dapat merubah dan menghapus catatan
4. *Entity Member* dengan *Entity* resep mpasi
   a. *Member* dapat memilih satu atau lebih dari satu resep mpasi menjadi resep *favorite*
5. *Entity* Info Perkembangan Bayi

Entity info perkembangan bayi tidak memiliki relasi dengan *Entity* lain, karena pada perkembangan bayi hanya akan menjelaskan atau menampilkan informasi tentang perkembangan bayi seperti motorik kasar, motorik halus, bahasa, dan sosial berdasarkan usia bayi yaitu pada usia 1 sampai dengan 12 bulan.

6. *Entity* Info Jadwal Imunisasi

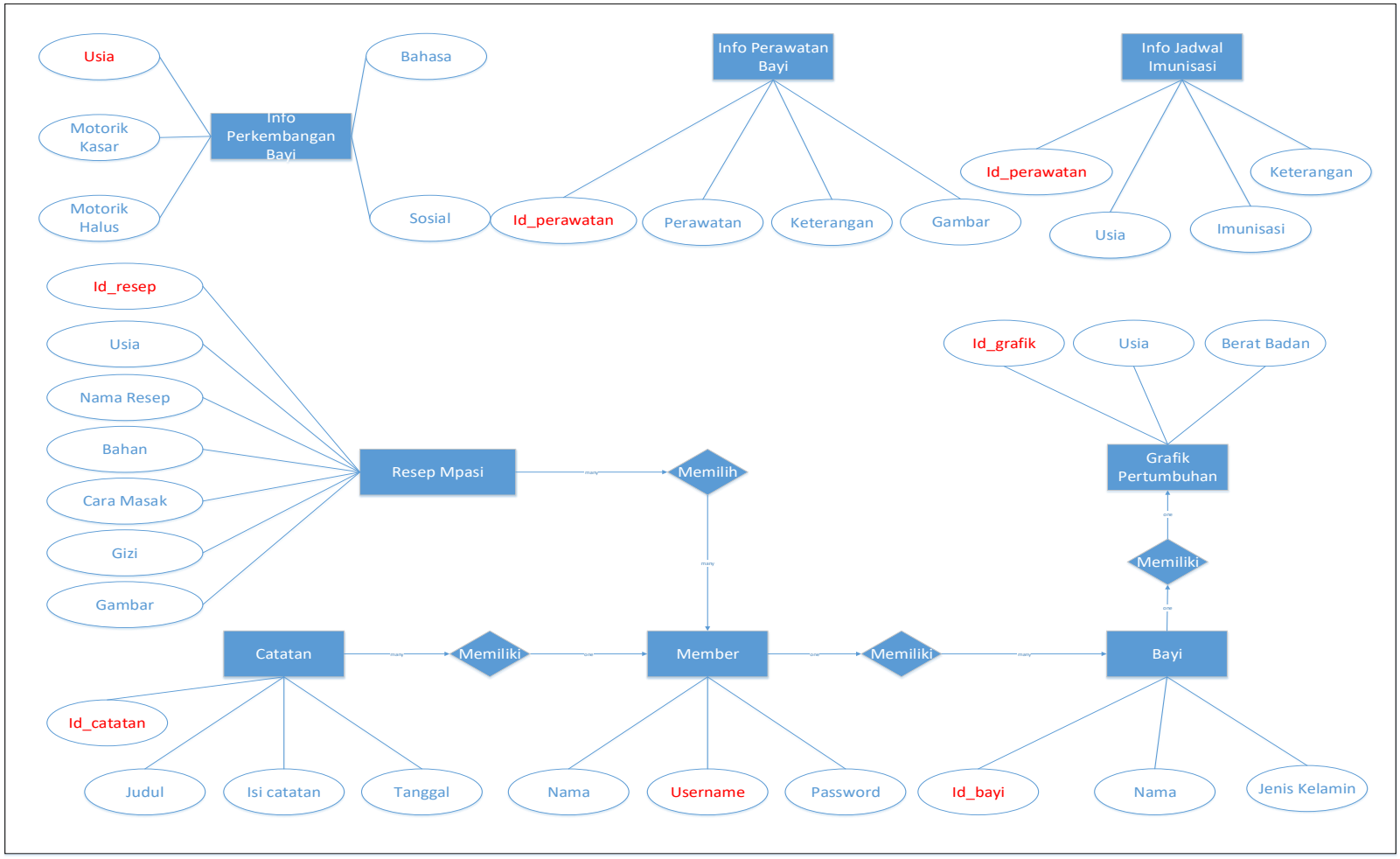
Entity info jadwal imunisasi tidak memiliki relasi dengan *Entity* lain, karena pada jadwal imunisasi hanya akan menjelaskan atau menampilkan informasi jenis imunisasi apa yang harus diberikan kepada bayi yang dilakukan pada setiap bulan sesuai dengan usia bayi.

7. *Entity* Info Perawatan Bayi

Entity info perawatan bayi tidak memiliki relasi dengan *Entity* lain, karena pada perawatan bayi hanya akan menampilkan atau menjelaskan tentang bagaiman melakukan perawatan bayi. Pengguna dapat melihat perawatan bayi dengan memilih jenis perawatan bayi.

3.3.4 *ERD (Entity Relationship Diagram)*

Dalam perancangan *database* pada aplikasi android *Monitoring* perkembangan dan kecerdasan bayi yang dibuat terdapat *Entity* grafik pertumbuhan, *Entity* resep mpasi, *Entity* info perkembangan bayi, *Entity* info jadwal imunisasi, *Entity* info perawatan bayi, *Entity* catatan, *Entity member*, dan *Entity* bayi. Dari semua *Entity t*ersebut membentuk sebuah ERD (*Entity Relationship Diagram).* ERD tersebut dapat dilihat pada gambar 3.14:



Gambar 3.14 *Entity Relationship Diagram*

Penjelasan ERD pada gambar 3.15 adalah sebagai berikut:

a. *Entity* bayi mempunyai relasi *one to one* dengan *Entity* grafik pertumbuhan, artinya satu bayi hanya bisa memiliki satu grafik pertumbuhan. Begitupun sebaliknya satu grafik pertumbuhan hanya bisa dimiliki oleh satu bayi
b. *Entity member* mempunyai relasi *one to many* dengan *Entity* bayi, artinya satu *member* bisa memiliki satu atau lebih dari satu bayi
c. *Entity member* mempunyai relasi *one to many* dengan *Entity* catatan, artinya satu *member* bisa memiliki satu atau lebih dari satu catatan.
d. *Entity member* mempunyai relasi *one to many* dengan *Entity* resep, artinya *member* bisa memilih satu atau lebih dari satu resep menjadi resep *favorite*
e. *Entity* info jadwal imunisasi tidak mempunyai relasi dengan *Entity* manapun
f. *Entity* info perawatan tidak mempunyai relasi dengan *Entity* manapun
g. *Entity* info perkembangan bayi tidak mempunyai relasi dengan *Entity* manapun

3.4 Perancangan *Database Level* Logikal

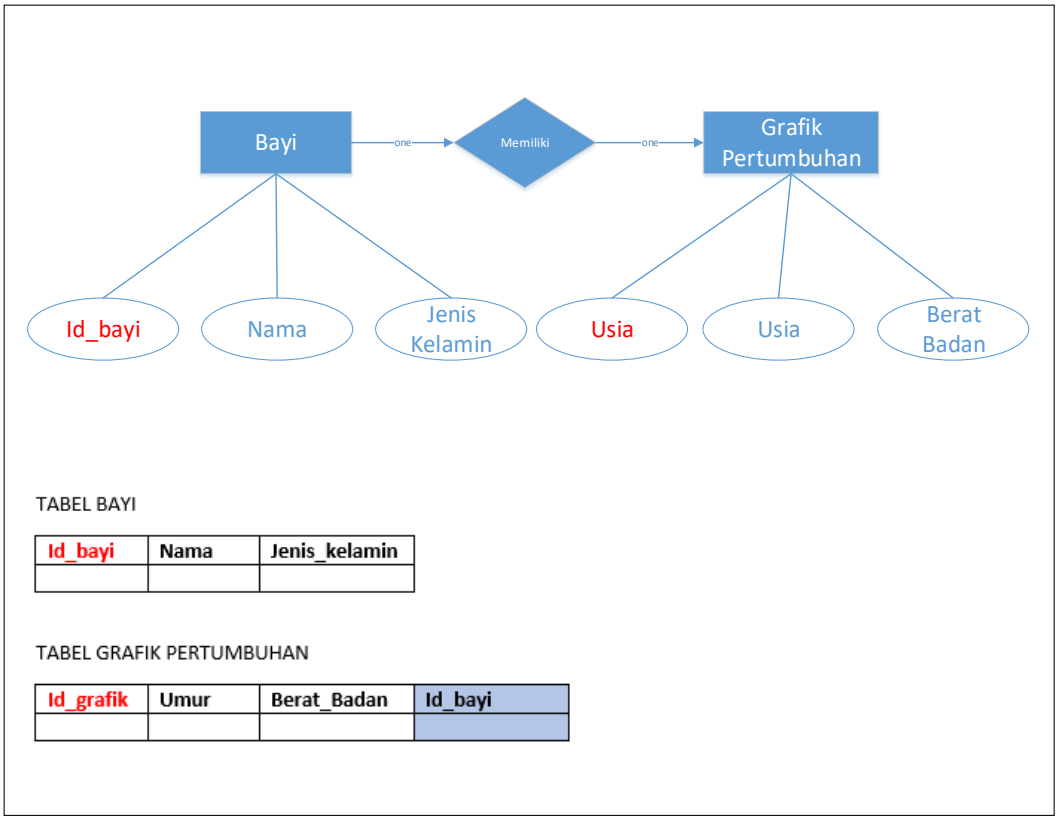
Perancangan *database* level logikal merupakan tahapan untuk memetakan proses perancangan konseptual kedalam model *database* yang akan digunakan, model yang digunakan dalam perancangan *database* ini adalah model relasi.

3.4.1 Konversi ERD ke dalam Tabel

Konversi ERD ke dalam bentuk tabel adalah proses memasukan *Entity* menjadi nama tabel dan *attribute-attribute* menjadi nama kolom beserta dengan relasi nya.

1. Konversi *Entity* bayi dengan *Entity* grafik pertumbuhan

Entity bayi memiliki relasi *one to one* dengan *Entity* grafik pertumbuhan. Berikut adalah konversi dari relasi *Entity* bayi dengan *Entity* grafik pertumbuhan. Dapat dilihat pada gambar 3.15.



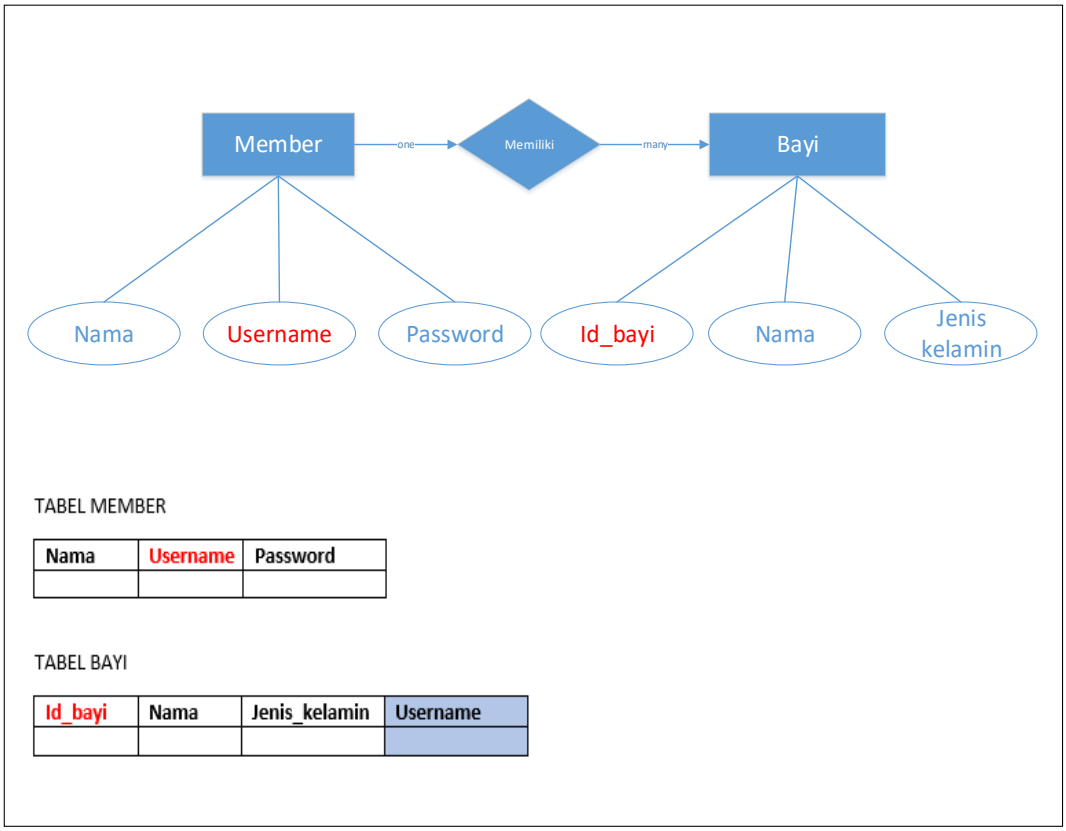
Gambar 3.15 Konversi relasi *enity* bayi dan *Entity* grafik pertumbuhan

Penjelasan gambar 3.15 adalah:

Setiap baris data pada tabel bayi dihubungkan hanya ke satu baris data pada tabel grafik pertumbuhan begitu juga sebaliknya, maka pada tabel grafik pertumbuhan perlu ditambahkan kolom *id_bayi* yang berfungsi sebagai *foreign key* dari tabel bayi.

2. Konversi *Entity member* dengan *Entity* bayi

Entity member memiliki relasi *one to many* dengan *Entity* bayi. Berikut adalah konversi dari relasi *Entity member* dengan *Entity* bayi. Dapat dilihat pada gambar 3.16.



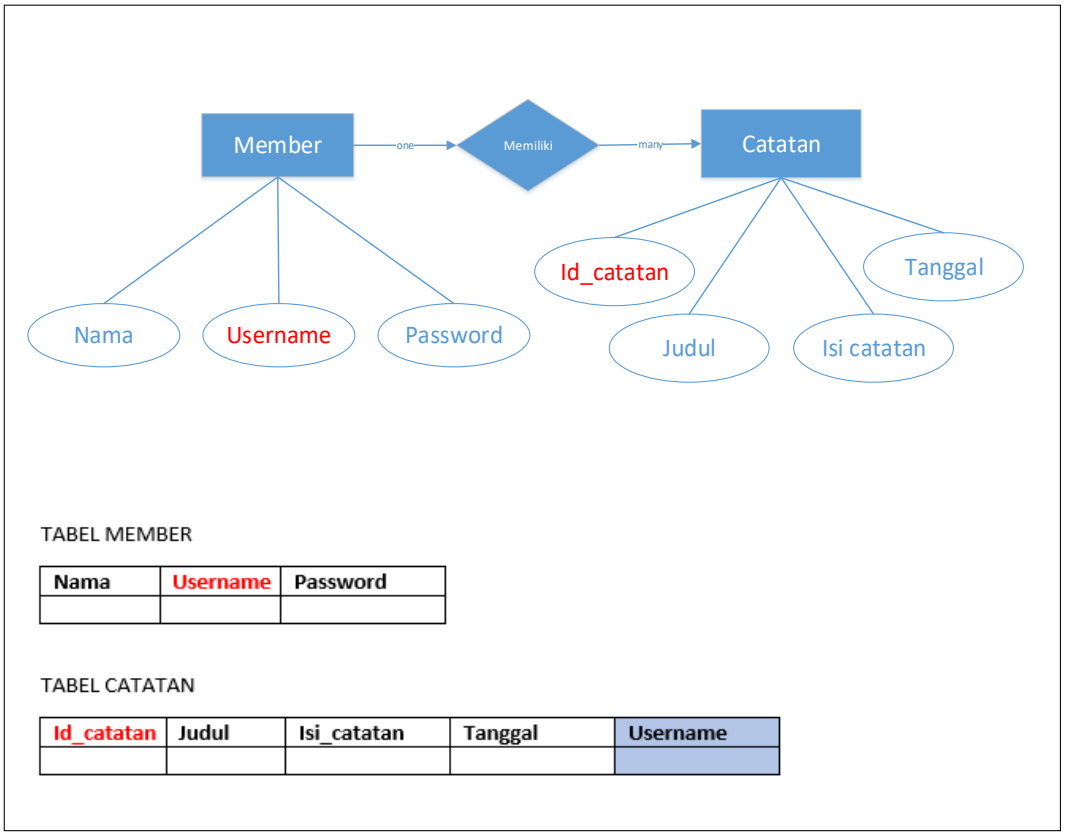
Gambar 3.16 Konversi relasi *Entity member* dengan *Entity* bayi

Penjelasan pada gambar 3.16 adalah:

Satu baris data pada tabel *member* dapat dihubungkan ke satu baris atau lebih dari satu baris data pada tabel bayi. maka pada tabel bayi perlu ditambahkan kolom *username* yaitu sebagai *foreign key* dari tabel *member*.

3. Konversi *Entity member* dengan *Entity* catatan

Entity member memiliki relasi *one to many* dengan *Entity* catatan. Berikut adalah konversi dari relasi *Entity member* dengan *Entity* catatan. Dapat dilihat pada gambar 3.17.



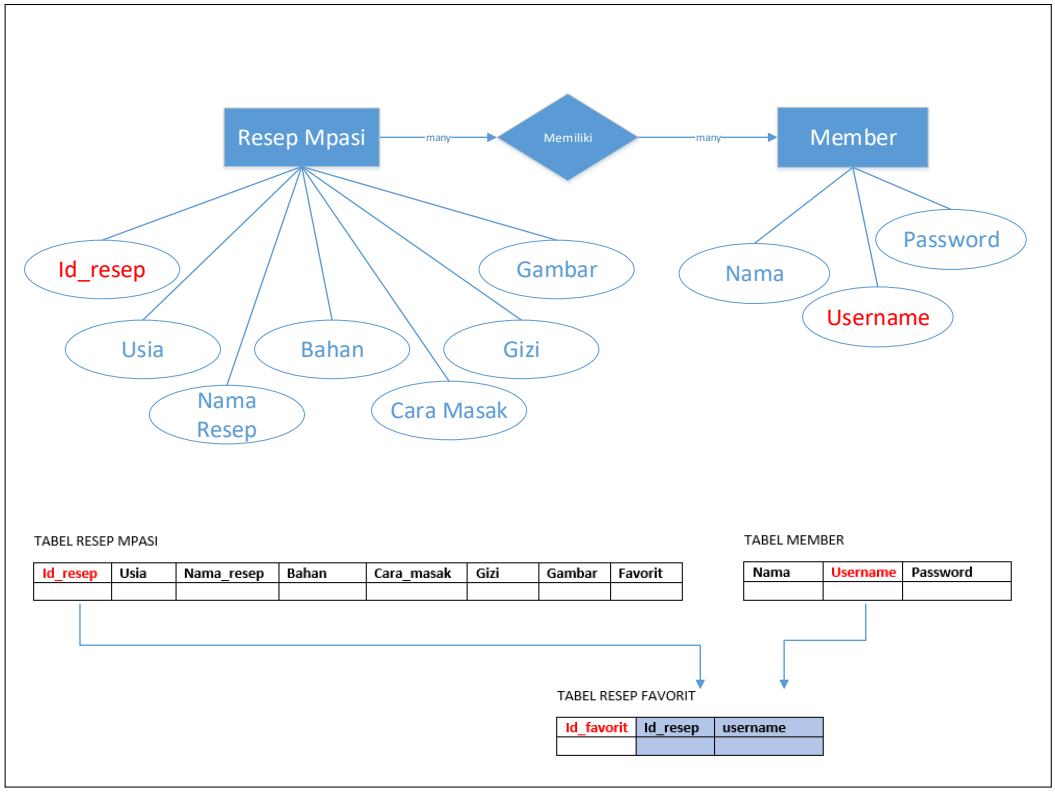
Gambar 3.17 Konversi relasi *Entity member* dengan *Entity* catatan

Penjelasan pada gambar 3.17 adalah:

Satu baris data pada tabel *member* dapat dihubungkan ke satu baris atau lebih dari satu baris data pada tabel catatan. maka pada tabel catatan perlu ditambahkan kolom *username* yaitu sebagai *foreign key* dari tabel *member* yang berfungsi sebagai penghubung antara tabel *member* dengan tabel catatan.

4. Konversi *Entity* Resep Mpasi dengan *Entity Member*

Entity resep mpasi memiliki relasi *many to many* dengan *Entity member*. Berikut adalah konversi dari relasi *Entity* resep mpasi dengan *Entity member*. Dapat dilihat pada gambar 3.18.



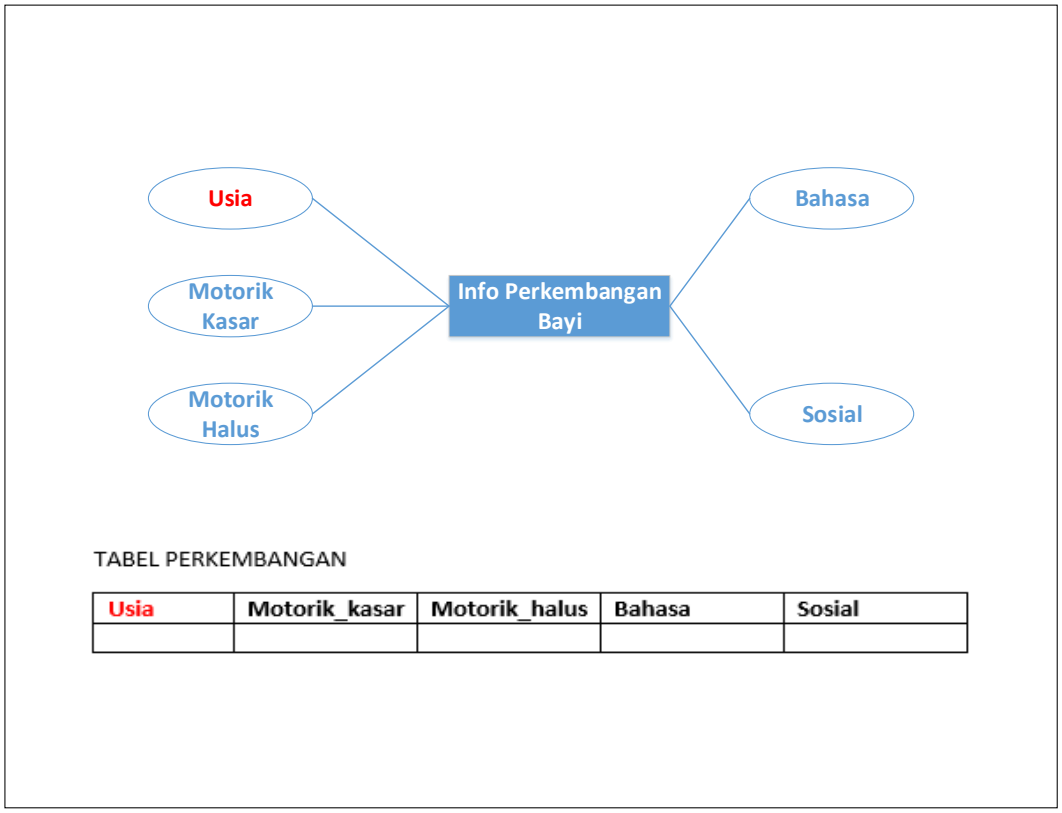
Gambar 3.18 konversi relasi *Entity* resep mpasi dan *Entity member*

Penjelasan pada gambar 3.18 adalah:

Satu baris data atau lebih pada tabel resep mpasi dapat dihubungkan ke satu baris atau lebih dari satu baris data pada tabel *member*, maka perlu ditambahkan tabel baru sebagai penghubung antara tabel resep mpasi dan tabel *member* yaitu tabel favorit, tabel favorit ini berfungsi untuk menyimpan data resep yang dipilih member sebagai favorit. Pada tabel favorit perlu ditambahkan kolom id_resep dan *username* sebagai penghubung antara tabel favorit dengan tabel resep mpasi dan tabel favorit dengan tabel *member*. Pada tabel resep mpasi ditambahkan kolom favorit yang berfungsi sebagai tanda bahwa resep tersebut sudah dipilih sebagai favorit atau belum.

5. Konversi *Entity* Info perkembangan bayi

Entity perkembangan bayi merupakan *Entity* yang berdiri sendiri (tidak memiliki relasi dengan *Entity* lain). Berikut adalah konversi *Entity* perkembangan bayi. Dapat dilihat pada gambar 3.19.



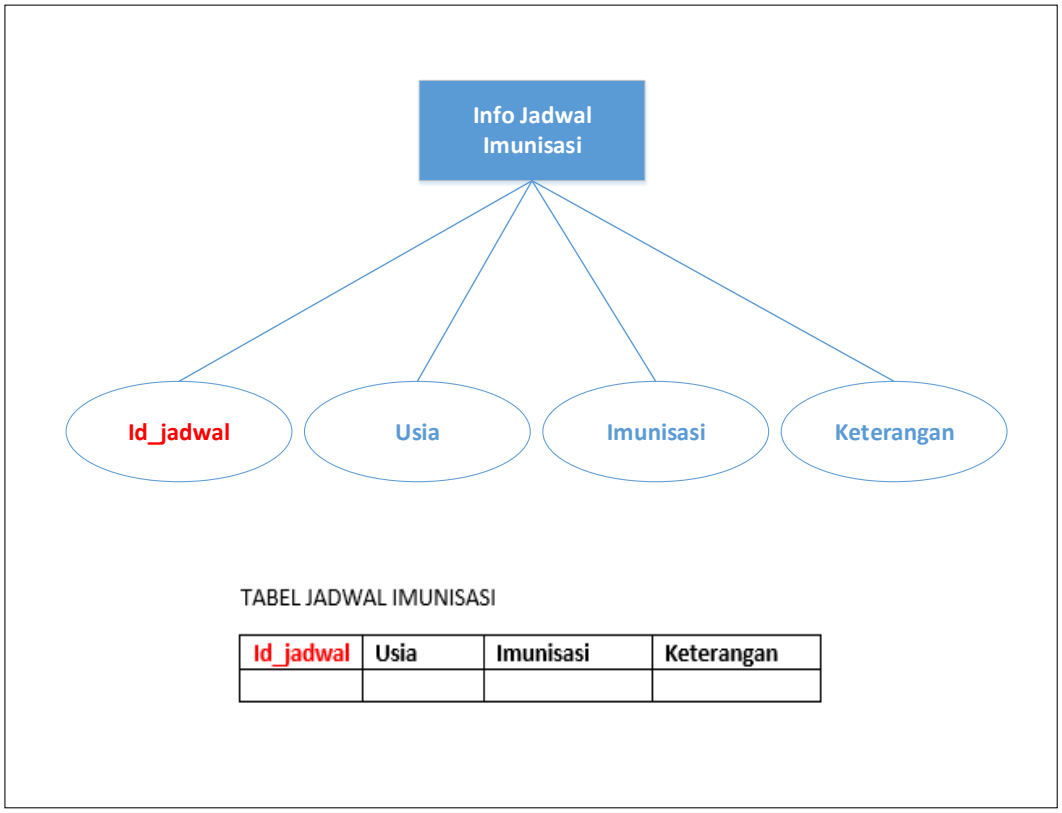
Gambar 3.19 konversi *Entity* perkembangan bayi

Penjelasan pada gambar 3.19 adalah:

Pada proses konversi *Entity* perkembangan bayi ke dalam bentuk tabel pada gambar 3.19 dijelaskan bahwa *Entity* dikonversi menjadi nama tabel dan *attribute* menjadi nama kolom.

6. Konversi *Entity* Info Jadwal Imunisasi

Entity jadwal imunisasi merupakan *Entity* yang berdiri sendiri (tidak memiliki relasi dengan *Entity* lain). Berikut adalah konversi *Entity* jadwal imunisasi. Dapat dilihat pada gambar 3.20.



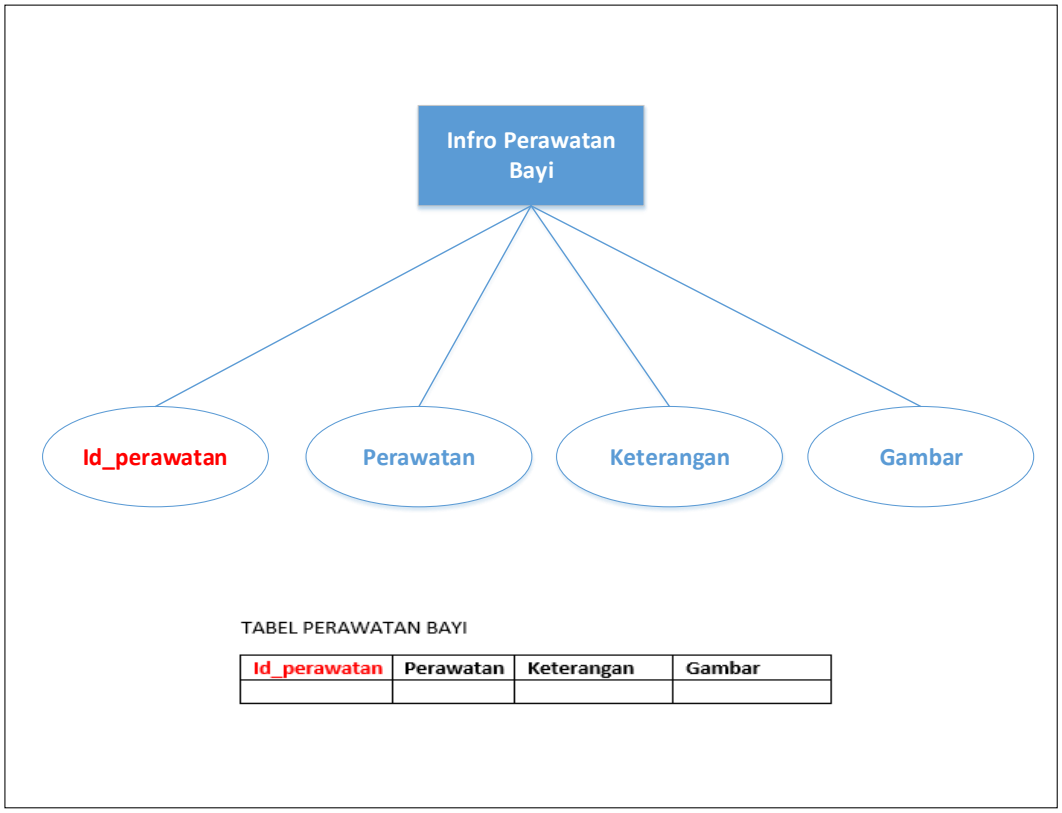
Gambar 3.20 konversi *Entity* jadwal imunisasi

Penjelasan pada gambar 3.20 adalah:

Pada proses konversi *Entity* jadwal imunisasi ke dalam bentuk tabel pada gambar 3.20 dijelaskan bahwa *Entity* dikonversi menjadi nama tabel dan *attribute* menjadi nama kolom.

7. Konversi *Entity* Info Perawatan Bayi

Entity perawatan bayi merupakan *Entity* yang berdiri sendiri (tidak memiliki relasi dengan *Entity* lain). Berikut adalah konversi *Entity* perawatan bayi. Dapat dilihat pada gambar 3.21.



Gambar 3.21 konversi *Entity* perawatan bayi

Penjelasan pada gambar 3.21 adalah:

Pada proses konversi *Entity* perawatan bayi ke dalam bentuk tabel pada gambar 3.21 dijelaskan bahwa *Entity* dikonversi menjadi nama tabel dan *attribute* menjadi nama kolom.

3.4.2 Kamus Data *Database*

Kamus data (*data dictionary*), merupakan salah satu komponen kunci dari DBMS (*Database Management System*) yang mencakup informasi mengenai struktur *database*. Setiap elemen data yang disimpan dalam *database* memiliki catatan kamus data yang mendeskripsikan elemen-elemen tersebut.

1. Tabel Grafik Pertumbuhan

Berikut ini merupakan kamus data dari tabel Guru:

Tabel 3.1 kamus data tabel grafik pertumbuhan

No	Nama Field	Tipe Data	Length field	Keterangan
1	Id_grafik	Int	-	Berfungsi sebagai primary key. Contoh data (1, 2, 3 , . . .)
2	Usia	Int	2	Berfungsi untuk menyimpan usia bayi dalam bulan, contoh data (1,2,3 s/d 12) bulan.
3	Berat Badan	Float	3	Berfungsi untuk menyimpan data berat badan bayi dalam kg. contoh (2,5 , 3,5 .) kg.
4	Id_bayi	Int	-	Berfungsi sebagai *foregin key* dari tabel bayi.

2. Tabel Bayi

Berikut ini merupakan kamus data dari tabel Bayi:

Tabel 3.2 kamus data tabel bayi

No	Nama Field	Tipe Data	Length field	Keterangan
1	Id_bayi	Int	-	Berfungsi sebagai primary key. Contoh data (1, 2, 3 , . . .)
2	Nama	Varchar	30	Berfungsi untuk menyimpan nama bayi. Dengan panjang maksimal 30 karakter
3	Jenis Kelamin	Varchar	10	Berfungis untuk menyimpan informasi jenis kelamin bayi, contoh data (laki-laki, perempuan)
4	*username*	varchar	10	Berfungsi sebagai *foregin key* dari tabel *member*.

3. Tabel *Member*

Berikut ini merupakan kamus data dari tabel *Member*:

Tabel 3.3 kamus data tabel member

No	Nama Field	Tipe Data	Length field	Keterangan
1	Nama	Varchar	30	Berfungsi untuk menyimpan nama *user*. Dengan panjang maksimal 30 karakter
2	*username*	Varchar	10	Berfungsi sebagai primary key. Dengan panjang maksimal 10 karakter. Digunaka member ketika login.
3	*password*	Varchar	8	Berfungis untuk menyimpan *password* yang akan digunakan *user* ketika *login*. Dengan panjang maksimal 8 karakter

8. Tabel Catatan

Berikut ini merupakan kamus data dari tabel Catatan:

Tabel 3.4 kamus data tabel catatan

No	Nama Field	Tipe Data	Length field	Keterangan
1	Id_catatan	Int	-	Berfungsi sebagai primary key. Contoh data (1, 2, 3 , . . .)
2	Judul	Varchar	20	Berfungsi untuk menyimpan judul dari catatan dengan panjang maksimal 20 karakter
3	Isi_catatatn	Text	-	Berfungsi untuk menyimpan isi catatan.
4	Tanggal	Date	-	Berfungis untuk menyimpan data tanggal, contoh data (01-01-2010)
6	*username*	varchar	10	Berfungsi sebagai *foreign key* dari tabel *member. Dengan panjang maksimal 20 karakter*

9. Tabel Resep MPASI

Berikut ini merupakan kamus data dari tabel Resep MPASI:

Tabel 3.5 kamus data tabel resep MPASI

No	Nama Field	Tipe Data	Length field	Keterangan
1	Id resep	Int	-	Berfungsi sebagai primary key. Contoh data (1, 2, 3 , . . .)
2	Usia	Int	2	Berfungsi untuk menyimpan data usia dalam bentuk bulan contoh data (1-12) bulan
3	Nama resep	Varchar	50	Berfungsi untuk menyimpan isi nama resep, contoh data (Sauteed Kale Promina Ayam Kampung Kacang Polong)
4	Bahan	Text	-	Berfungis untuk menyimpan data bahan-bahan dari resep.
5	Cara masak	Text	-	Berfungsi untuk menyimpan data cara masak dari resep.
6	gizi	Text	-	Berfungsi untuk menyimpan data angka kandungan gizi pada resep,
7	Gambar	Text	-	Berfungsi untuk menyimpan gambar dari resep
8	Favorit	enum	-	Berfungsi untuk mengetahui apakah resep sudah dipilih menjadi favorit atau belum. Contoh data (Yes dan No) .

10. Tabel Perkembangan Bayi

Berikut ini merupakan kamus data dari tabel Perkembangan Bayi:

Tabel 3.6 kamus data tabel perkembangan bayi

No	Nama Field	Tipe Data	Length field	Keterangan
1	Umur	Int	2	Berfungsi sebagai primary key. Contoh data (1,2,3 s/d 12) bulan
2	Motorik kasar	Text	-	Berfungsi untuk menyimpan informasi gerakan kasar bayi,
3	Motorik halus	Text	-	Berfungsi untuk menyimpan informasi gerakan halus bayi,
4	Bahasa	Text	-	Berfungsi untuk menyimpan informasi komunikasi bayi,
5	Sosial	Text	-	Berfungsi untuk menyimpan informasi terkait kemandirian bayi

11. Tabel Jadwal Imunisasi

Berikut ini merupakan kamus data dari tabel Jadwal Imunisasi:

Tabel 3.7 kamus data tabel jadwal imunisasi

No	Nama Field	Tipe Data	Length field	Keterangan
1	Id_jadwal	Int	-	Berfungsi sebagai primary key. Contoh data (1,2,3, . . .)
2	Usia	Int	2	Berfungsi untuk menyimpan usia bayi dalam bentuk bulan dengan panjang 2 karakter, contoh data (1,2 s/d 12) bulan
3	Imunisasi	varchar	20	Berfungsi untuk menyimpan nama jenis imunisasi, dengan panjang kurang dari 20 karakter, contoh data (polio, hepatitis)

No	Nama Field	Tipe Data	Length field	Keterangan
4	Keterangan	Text	-	Berfungsi untuk menyimpan informasi keterangan dari imunisasi.

12. Tabel Perawatan Bayi

Berikut ini merupakan kamus data dari tabel Perawatan Bayi:

Tabel 3.8 kamus data tabel perawatan

No	Nama Field	Tipe Data	Length field	Keterangan
1	Id_perawatan	Int	-	Berfungsi sebagai primary key. Contoh data (1,2,3, . . .)
2	Perawatan	Varchar	50	Berfungsi untuk menyimpan nama perawatan bayi dengan jumlah karakter kurang dari 50, contoh data (perawatan hidung, perawatan mulut)
4	Keterangan	Text	-	Berfungsi untuk menyimpan informasi keterangan dari perawatan bayi
5	Gambar	Text	-	Berfungsi unyuk menyimpan data gambar dari perawatan bayi

13. Tabel Resep *Favorite*

Berikut ini merupakan kamus data dari tabel Resep *Favorite*:

Tabel 3.9 kamus data tabel resep favorite

No	Nama Field	Tipe Data	Length field	Keterangan
1	Id_favorite	Int	-	Berfungsi sebagai primary key. Contoh data (1,2,3, . . .)
2	Id_resep	Int	-	Berfungsi sebagai *foreign key* dari tabel resep, untuk menampilkan data resep.
4	*username*	Varchar	10	Berfungsi sebagai *foreign key* dari tabel *member*, dengan panjang maksimal 10 karakter

3.5 Kebijakan Akses

3.5.1 Kebijakan Akses Admin

1. *Admin* dapat Menambah jadwal imunisasi, jenis imunisasi, perkembangan bayi, perawatan bayi, dan resep mpasi
2. *Admin* dapat Mengubah jadwal imunisasi, jenis imunisasi, perkembangan bayi, perawatan bayi, dan resep mpasi
3. *Admin* dapat Menghapus jadwal imunisasi, jenis imunisasi, perkembangan bayi, perawatan bayi, dan resep mpasi
4. *Admin* hanya dapat melihat data nama *member*, data bayi, dan data grafik pertumbuhan bayi.

3.5.2 Kebijakan Akses Master

Master mempunyai hak penuh atas *database*, Master dapat melihat, menambah, mengubah, dan menghapus data pada *database.*