**Studi Kasus**

MASALAH

Kasus :

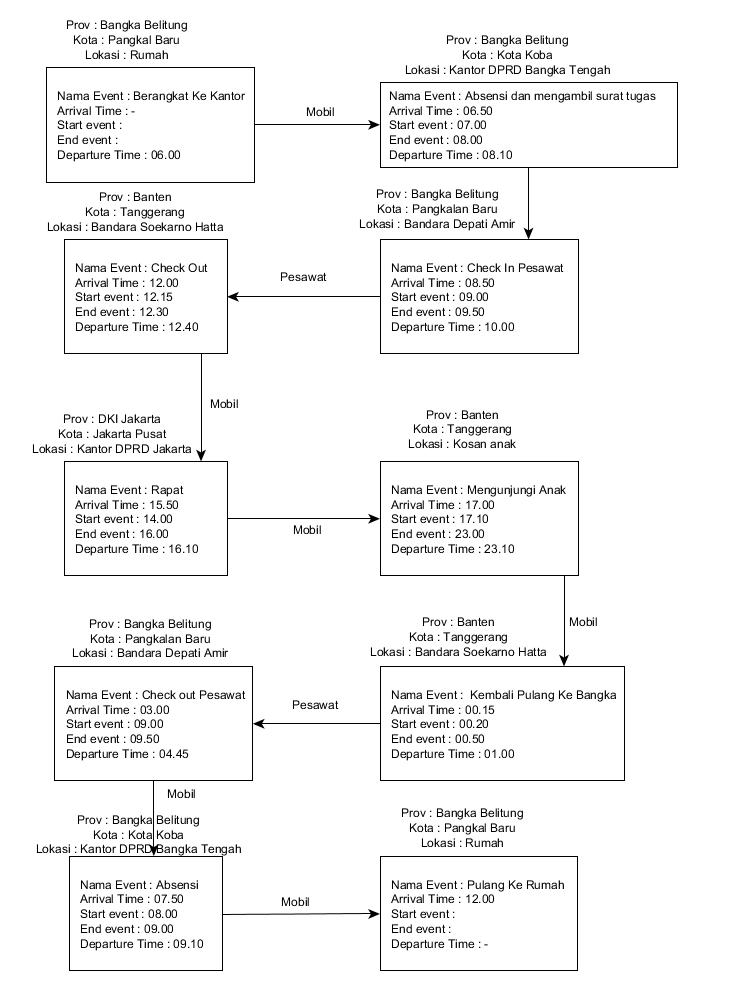
1. Bu maryam adalah DPRD kabupaten bangka tengah periode 2014-2019, hampir setiap hari beliau kerja, dan kadang hari liburnya juga di pakai untuk pergi bersama keluarga atau ada urusan kerja mendadak. Beliau bertugas di DPRD bagian pendidikan, jadi selain urusan di kantor DPRD beliau juga ke sekolah-sekolah di kampung yang masih kurang dapat perhatian pemerintah, selain itu minimal dua minggu sekali beliau harus keluar kota, misalnya ke jakarta untuk bertugas. beliau juga mempunyai anak yang sekarang sedang kuliah maka tak jarang ketika beliau sedang rapat di jakarta beliau menyempatkan diri untuk bertemu dengan anaknya di tanggerang, beliau selalu berkendara sendiri atau memakai kendaraan umum, karena beliau tidak punya supir. kadang untuk pergi rapat ke kantor yang berjarak 40km dari rumahnya bu maryam juga tidak segan memakai sepeda motor, itu tergantung bagaimana kondisi jalan dan cuaca serta keperluan lainnya, karena memakai kendaraan bermotor bisa lebih cepat jika melalui medan di jalan perkampungan atau saat berada di tengah kota koba bangka selatan.

SOLUSI PEMECAHAN MASALAH

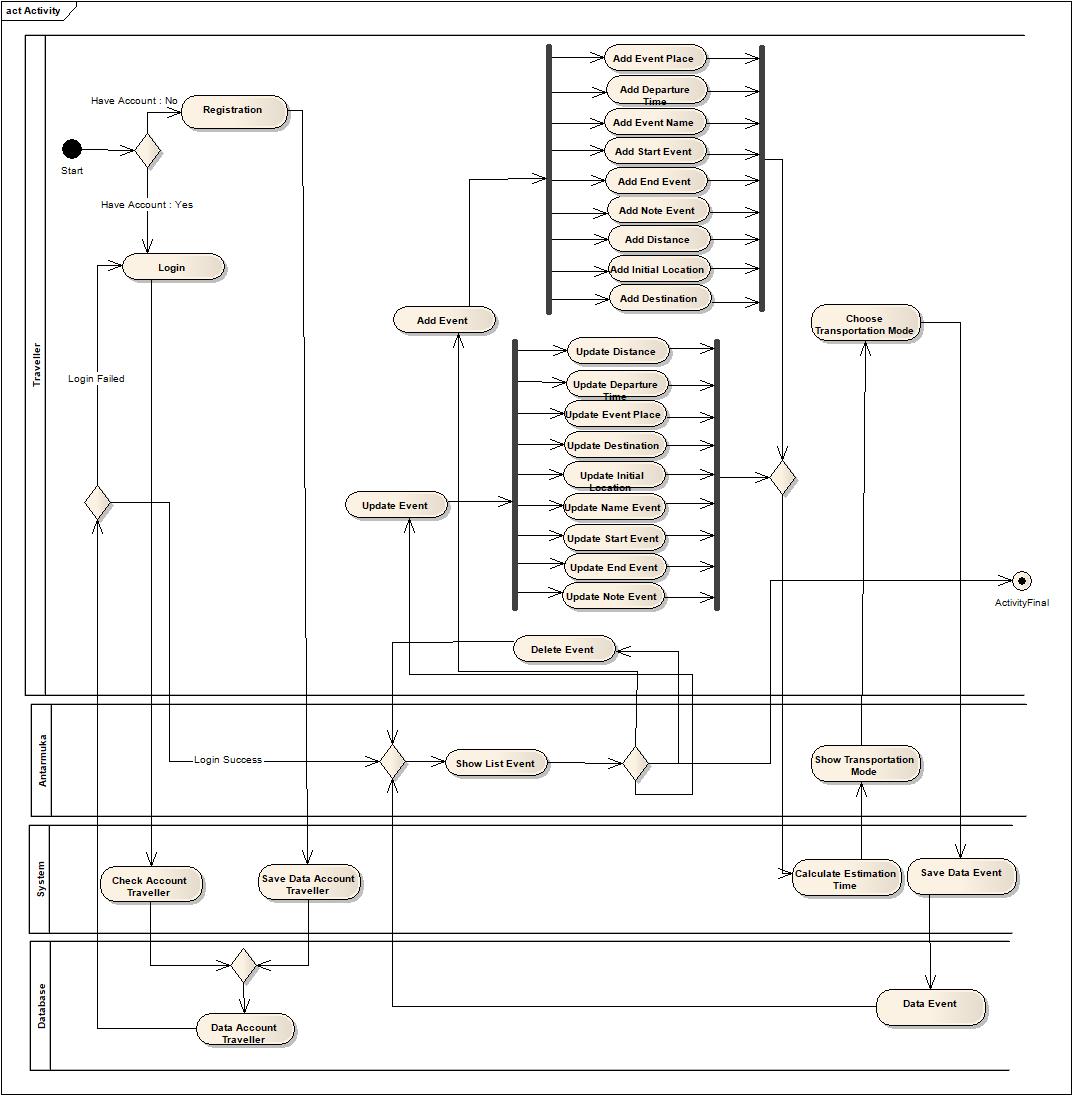
Dengan melihat kegiatan sehari hari yang dilakukan oleh bu Maryam dimana dengan kesibukannya di luar rumah dan bepergian di dalam maupun luar kota maka aplikasi travelendar cocok untuk beliau.

|  |  |
| --- | --- |
| **Masalah** | **Solusi** |
| Agenda kegiatan yang banyak | Dibuatkan fitur remainder di kalender |
| Pilihan transportasi yang banyak | Di buarkan rekomendasi kendaraan sesuai jarak, kondisi jalan, dan cuaca |
| Akan ada konsekuensi keterlambatan | Di buatkan reminder dan rekomendasi waktu keberangkatan |
| Belum hafal jalan ke tujuan | Di tambahkan fitur map atau rekomendasi kendaraan umum |
| Kesehatan yang harus tetap di jaga | Di tambahkan fitur reminder makan dan olahraga |

**Contoh Skema data kasus :**



**Use Case Scenario**

****

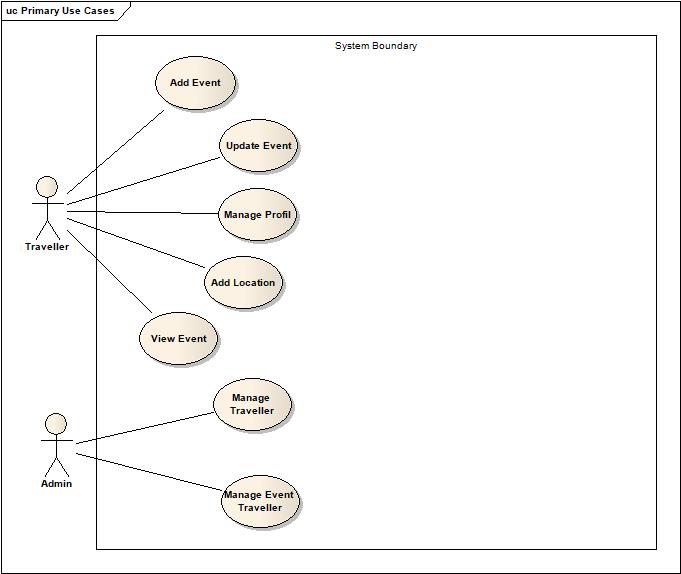
**Skenario Program**

|  |  |  |
| --- | --- | --- |
| Pengguna | UI | Sistem |
| 1. Pengguna Menambahkan sebauh Event, seperti input 2. Name Event 3. Start Date 4. End Date 5. Location 6. Arrival Time 7. Departure Time 8. Note | H utama add new event | 1. Data Event disimpan dalam database |
| 1. Pengguna bisa memilih kendaraan yang sudah di rekomendasikan oleh sistem | H Make travel plan | 1. Sistem telah menghitung jarak dan estimasi waktu dari setiap Event untuk bisa menampilkan transportasi yang direkomendasikan |
| 1. Pengguna bisa melihat List Event yang sudah ditambahkan dan bisa sewaktu waktu di rubah kembali jika mengalami kesalahan atau menghapus Event yang telah ditambahkan | H Utama | 1. Data Event sudah ditambahkan Transportasi yang digunakan dan estimasi waktu yang ditentukan oleh sistem untuk pengguna |

**Data Master Dan Data Transaksi**

|  |  |
| --- | --- |
| Data Master | Data Transaksi |
| 1. Lokasi (nama kota) | 1. Input Username |
| 1. Jarak | 2. Input Password |
| 1. Waktu Event | 3. Input Fullname |
| 1. Transportasi | 4. Input Email |
| 1. Kalender | 5. Input Event Name |

**Use Case Diagram**

****

*Use Case* yang terdapat pada rancangan system Travelendar ini memiliki 2 aktor yaitu *Traveller* dan memiliki 8 buah *use case*, berikut penjelasan dari tiap-tiap *use case*:

* *Use case*  *add* Event dilakukan oleh Traveller, memiliki fungsi melakukan penambahan Event
* *Use case add* Location dilakukan oleh Traveller, memiliki fungsi menambahkan Location
* *Use case Manage* Profil dilakukan oleh Traveller, memiliki fungsi merubah data informasi Profil Traveller
* *Use case View* Event dilakukan oleh Traveller, memiliki fungsi untuk melihat Event yang telah ditambahkan
* *Use case Update* Event dilakukan oleh Traveller, memiliki fungsi untuk merubah Event atau menghapus Event
* *Use case Manage* Event dilakukan oleh Admin, memiliki fungsi untuk mengatur daftar event yang telah ditambahkan oleh Traveller
* *Use case Manage* Traveller dilakukan oleh Admin, memiliki fungsi untuk mengatur daftar akun Traveller

**Use Case Description**

|  |  |  |
| --- | --- | --- |
| *Use case name* | Add Event | |
| *Scenario* | Traveller dapat menambahkan sebuah event | |
| *Triggering event* | Traveller bisa melihat Event yang telah ditambahkan | |
| *Brief description* | Traveller bisa melakukan penambahan event | |
| *Actors* | Traveller | |
| *Related use case* | - | |
| *Stakeholders* | Traveller menambahkan Event | |
| *Preconditions* | Traveller telah menentukan Event apa yang akan ditambahkan | |
| *Postconditions* | Event berhasil ditambahkan kedalam sistem | |
| *Flow of events* | *Actor* | *System* |
| 1. Bu Maryam menginputkan nama event 2. Bu Maryam menginputkan startevent ( tanggal dan waktu ) 3. Bu Maryam menginputkan End event (tanggal dan waktu ) 4. Bu Maryam menginputkan eventPlace 5. Bu Maryam memilih initial location ( province , city and detail ) 6. Bu Maryam menginputkan departure time 7. Bu Maryam memilih destination ( province , city and detail ) 8. Bu Maryam menginputkan Arrival time 9. Bu Maryam menginputkan Distance antara initial location dan destination 10. Bu Maryam menekan tombol next | 1.Menampilkan Screen Add event  2. Menyimpan data ke database di table event |
| *Exception conditions* |  | |

|  |  |  |
| --- | --- | --- |
| *Use case name* | Add Location | |
| *Scenario* | Traveller dapat menambahkan sebuah lokasi awal pemberangkatan dan lokasi tujuan | |
| *Triggering event* | Traveller bisa melihat Location yang telah ditambahkan | |
| *Brief description* | Traveller bisa melakukan penambahan lokasi | |
| *Actors* | Traveller | |
| *Related use case* | - | |
| *Stakeholders* | Traveller menambahkan Location | |
| *Preconditions* | Traveller telah menentukan Location mana yang akan ditambahkan | |
| *Postconditions* | Location berhasil ditambahkan kedalam sistem | |
| *Flow of events* | *Actor* | *System* |
|  |  |
| *Exception conditions* |  | |

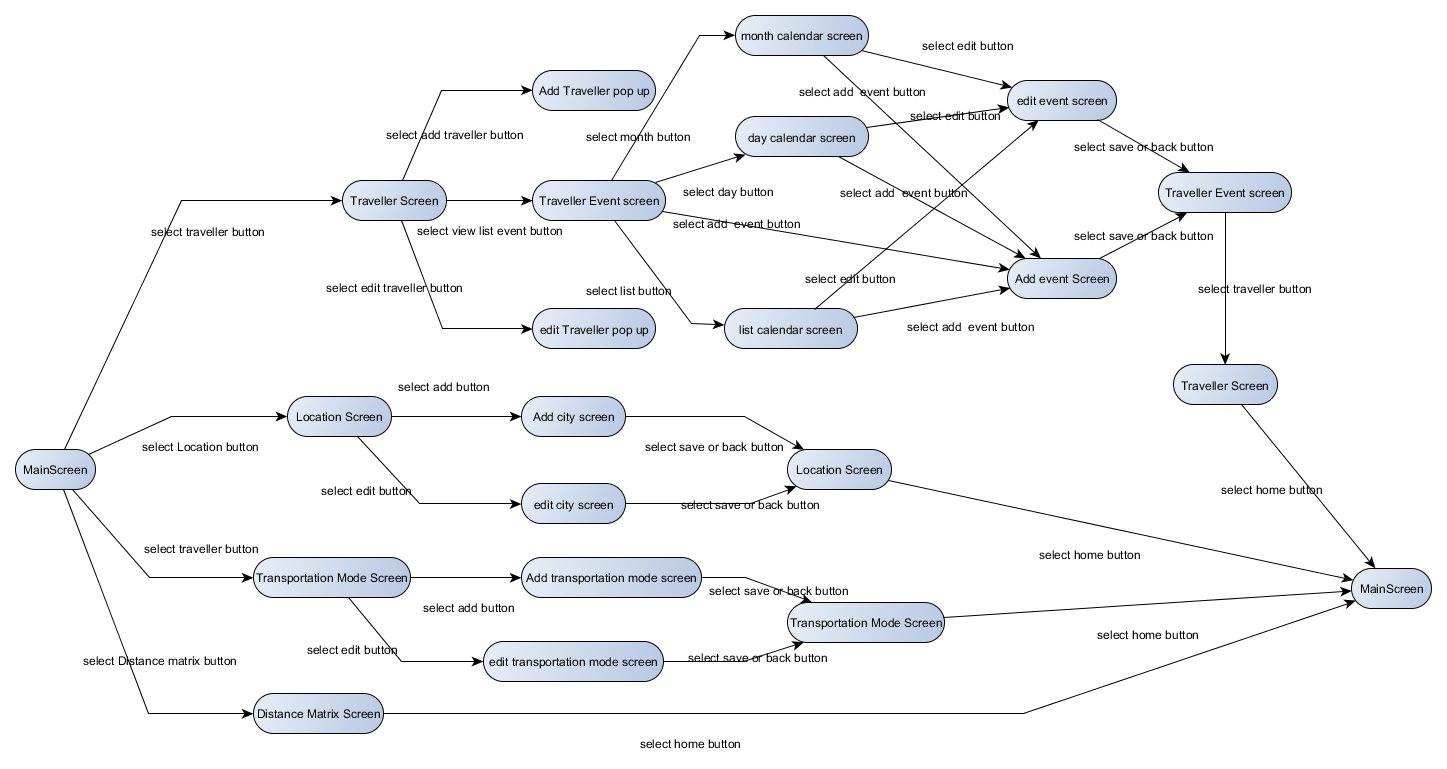
|  |  |  |
| --- | --- | --- |
| *Use case name* | Manage Profil | |
| *Scenario* | Traveller dapat merubah data informasi profil dari Traveller | |
| *Triggering event* | Traveller bisa melihat Profil yang telah dirubah | |
| *Brief description* | Traveller bisa melakukan perubahan data informasi Profil | |
| *Actors* | Traveller | |
| *Related use case* | - | |
| *Stakeholders* | Traveller merubah data informasi Profil | |
| *Preconditions* | Traveller telah merubah data informasi Profil | |
| *Postconditions* | Profil berhasil dirubah kedalam sistem | |
| *Flow of events* | *Actor* | *System* |
| 1. Kepala Sekolahmengakses menu *review* RPP 2. Kepala Sekolah memilih RPP yag akan di *review* dengan menekan tombol *download* 3. Kepala sekolah menekan tombol *approve* untuk menyetujui RPP dan menekan tombol *reject* untuk menolak RPP | * 1. Menamampilakan *screen review* RPP   2.1 Sistem melakukan *download* dokumen RPP yang dipilih  3.1 Sistem merubah status RPP yang ada pada *database* menjadi 1 jika dilakukan *approve* dan 2 jika dilakukan *reject*  3.2 Sistem menampilkan pesan sukses |
| *Exception conditions* |  | |

|  |  |  |
| --- | --- | --- |
| *Use case name* | Update Event | |
| *Scenario* | Traveller dapat merubah atau menghapus sebuah Event | |
| *Triggering event* | Traveller bisa melihat sebuah Event yang telah dirubah | |
| *Brief description* | Traveller bisa melakukan perubahan atau menghapus sebuah Event | |
| *Actors* | Traveller | |
| *Related use case* | - | |
| *Stakeholders* | Traveller merubah atau menghapus sebuah Event | |
| *Preconditions* | Traveller telah merubah atau menghapus sebuah Event | |
| *Postconditions* | Event berhasil dirubah kedalam sistem atau Event berhasil dihapus dari dalam sistem | |
| *Flow of events* | *Actor* | *System* |
|  |  |
| *Exception conditions* |  | |

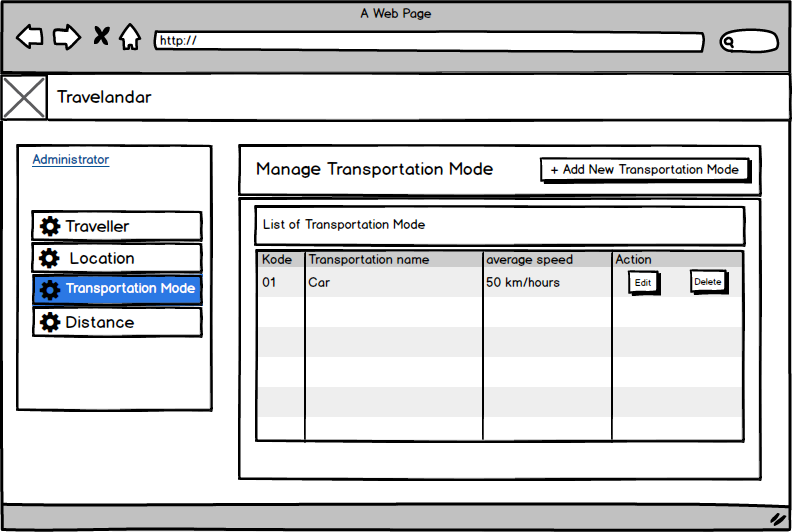
|  |  |  |
| --- | --- | --- |
| *Use case name* | View Event | |
| *Scenario* | Traveller dapat melihat sebuah Event dalam menu Calender Event | |
| *Triggering event* | - | |
| *Brief description* | Traveller bisa melihat Event yang telah ditambahkan dalam menu Calender Event | |
| *Actors* | Traveller | |
| *Related use case* | - | |
| *Stakeholders* | Traveller melihat sebuah Event | |
| *Preconditions* | - | |
| *Postconditions* | - | |
| *Flow of events* | *Actor* | *System* |
|  |  |
| *Exception conditions* |  | |

**User Interfce**

**Screenflow Admin panel**



**Mockup Aplikasi**



1. tabel TRANSPORTATION\_MODE

|  |  |  |  |
| --- | --- | --- | --- |
| NAMA FIELD | TIPE DATA | FUNGSI | HUBUNGAN DENGAN TABEL LAIN |
| TRANSPORTATION\_CODE | CHAR | YANG MENJADIKAN UNIK DAN SEBAGAI IDENTITAS DARI MASING-MAISNG KENDARAAN. | AKAN TURUN MENJADI FOREIGN KEY PADA TABLE TRAVEL |
| TRANSPORTATION\_NAME | VARCHAR | JENIS KENDARAAN | - |
| TRANSPORTATION\_SPEED | FLOAT | KECEPATAN KENDARAAN | - |

QUERY-QUERY YANG AKAN DIPERLAKUKAN PADA TABEL :  
1. CREATE TABLE

Query : CREATE table TRANSPORTATION\_MODE (‘TRANSPORTATION\_CODE CHAR() PRIMARY KEY NOT NULL’, ‘TRANSPORTATION\_NAME VARCHAR() NOT NULL’, ‘TRANSPORTATION\_SPEED FLOAT NOT NULL’);

2. INSERT

Jika ingin menambahkan transportasi baru tabel : INSERT into TRANSPORTATION\_MODE (‘TRANSPORTATION\_CODE’, ‘TRANSPORTATION\_NAME’, ‘TRANSPORTATION\_SPEED’) VALUES ()

3. SELECT

Jika ingin menampilkan seluruh isi data dari tabel TRANSPORTATION\_MODE : SELECT \* FROM TRANSPORTATION\_MODE;

Jika ingin menampilkan data kecepatan dari kendaraan mobil : SELECT TRANSPORTATION\_SPEED FROM TRANSPORTATION\_MODE WHERE TRANSPORTATION\_NAME=CAR;

Jika ingin menampilkan data kecepatan dari kendaraan motor : SELECT TRANSPORTATION\_SPEED FROM TRANSPORTATION\_MODE WHERE TRANSPORTATION\_NAME=MOTOR CYCLE;

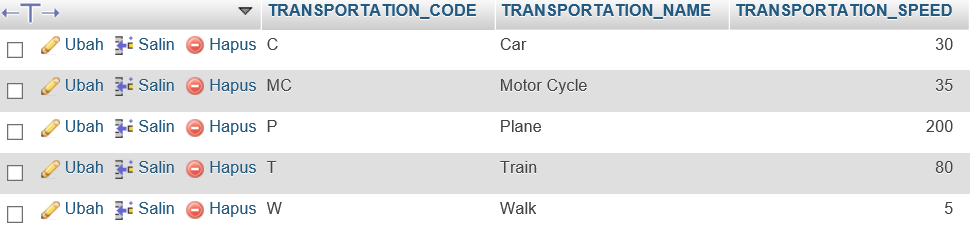
Jika ingin menampilkan data kecepatan dari kendaraan pesawat : SELECT TRANSPORTATION\_SPEED FROM TRANSPORTATION\_MODE WHERE TRANSPORTATION\_NAME=PLANE;

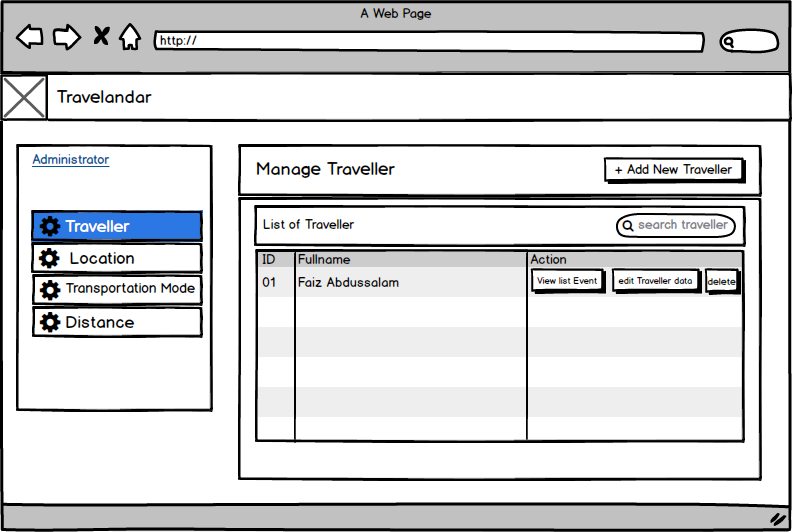
Jika ingin menampilkan data kecepatan dari kendaraan kereta : SELECT TRANSPORTATION\_SPEED FROM TRANSPORTATION\_MODE WHERE TRANSPORTATION\_NAME=TRAIN;

Jika ingin menampilkan data kecepatan jika berjalan kaki : SELECT TRANSPORTATION\_SPEED FROM TRANSPORTATION\_MODE WHERE TRANSPORTATION\_NAME=WALK;

SS :







5. tabel TRAVELLER

|  |  |  |  |
| --- | --- | --- | --- |
| NAMA FIELD | TIPE DATA | FUNGSI | HUBUNGAN DENGAN TABEL LAIN |
| TRAVELLER\_ID | LONG INTEGER |  |  |
| TRAVELLER\_NAME | VARCHAR |  |  |
| TRAVELLER\_EMAIL | VARCHAR |  |  |
| TRAVELLER\_PASSWORD | VARCHAR |  |  |
| TRAVELLER\_FULLNAME | VARCHAR |  |  |

QUERY-QUERY YANG AKAN DIPERLAKUKAN PADA TABEL :  
1.

CREATE TABLE IF NOT EXISTS `traveller` (

`TRAVELLER\_ID` bigint(20) NOT NULL AUTO\_INCREMENT,

`TRAVELLER\_NAME` varchar(20) DEFAULT NULL,

`TRAVELLER\_EMAIL` varchar(40) DEFAULT NULL,

`TRAVELLER\_PASSWORD` varchar(25) DEFAULT NULL,

`TRAVELLER\_FULLNAME` varchar(50) DEFAULT NULL,

PRIMARY KEY (`TRAVELLER\_ID`)

)

2. INSERT

Jika ingin menambahkan/mengisi data pada tabel TRAVELLER : INSERT into TRAVELLER ( `TRAVELLER\_ID`, `TRAVELLER\_ID`,`TRAVELLER\_EMAIL`, `TRAVELLER\_EMAIL`, `TRAVELLER\_PASSWORD`, `TRAVELLER\_FULLNAME`) VALUES ()

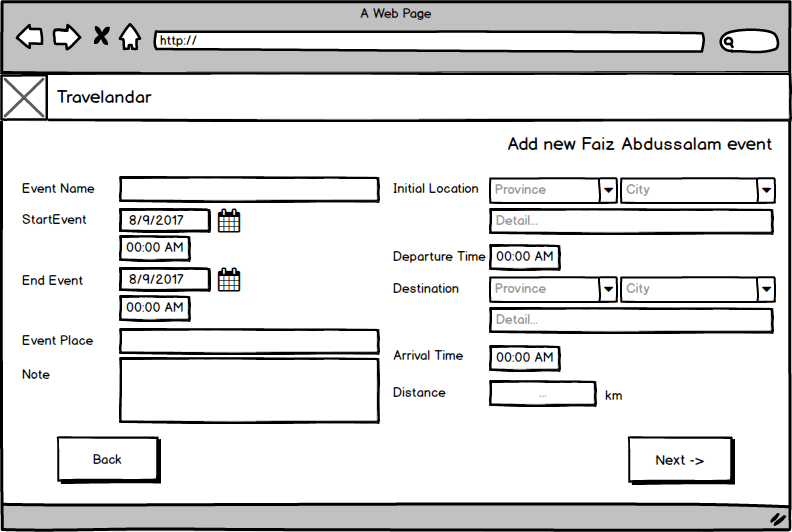
3. SELECT

Jika ingin menampilkan semua data di tabel TRAVELLER : SELECT\* FROM TRAVELLER;

SELECT TRAVELLER\_NAME FROM TRAVELLER WHERE TRAVELLER\_ID=1;

SS :





2. tabel EVENT

|  |  |  |  |
| --- | --- | --- | --- |
| NAMA FIELD | TIPE DATA | FUNGSI | HUBUNGAN DENGAN TABEL LAIN |
| EVENT\_ID | Long integer |  | AKAN TURUN MENJADI FOREIGN KEY PADA TABLE TRAVEL |
| EVENT\_NAME | Variable characters |  | - |
| START\_EVENT | DATE & TIME |  | - |
| END\_EVENT | DATE & TIME |  | - |
| NOTE | TEXT |  | - |
| PLACE | VARCHAR |  | - |

QUERY-QUERY YANG AKAN DIPERLAKUKAN PADA TABEL :  
1. CREATE TABLE

CREATE TABLE EVENT (‘EVENT\_ID Long integer() PRIMARY KEY AUTO INCREMENT NOT NULL ’,’EVENT\_NAME Variable characters() NOT NULL’,’START\_EVENT DATE & TIME() NOT NULL ’,’END\_EVENT DATE & TIME() NOT NULL’,’NOTE TEXT()’,’PLACE’)

2. INSERT

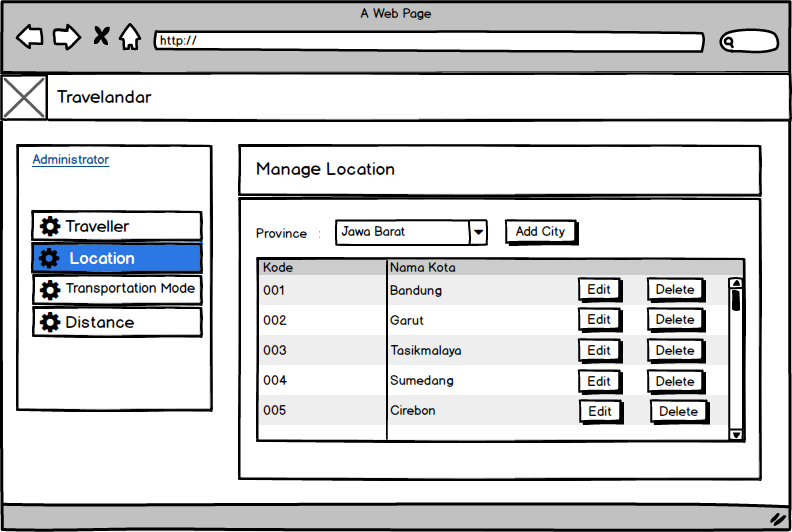
Jika ingin mengisi/menambah data pada tabel EVENT : INSERT into EVENT(‘EVENT\_ID’,’ EVENT\_NAME’,’ START\_EVENT’,’ END\_EVENT’,’ NOTE’,’ PLACE’) VALUES ()

3. SELECT

Jika ingin menampilkan seluruh isi data dari tabel EVENT : SELECT \* FROM EVENT;

Jika ingin menampilkan seluruh data tabel dimana jika yang di ketahui nya id : SELECT \* FROM EVENT WHERE EVENT\_ID=1;

SS :



6. tabel LOCATION

|  |  |  |  |
| --- | --- | --- | --- |
| NAMA FIELD | TIPE DATA | FUNGSI | HUBUNGAN DENGAN TABEL LAIN |
| **LOCATION\_IDKunci Utama** | INT |  |  |
| **CITY\_CODE** | CHAR |  |  |
| **ADDRESS\_PLACE** | VARCHAR |  |  |

QUERY-QUERY YANG AKAN DIPERLAKUKAN PADA TABEL :  
1. Create

CREATE TABLE IF NOT EXISTS `location` (

`LOCATION\_ID` int(11) NOT NULL AUTO\_INCREMENT,

`CITY\_CODE` char(2) NOT NULL,

`ADDRESS\_PLACE` varchar(55) DEFAULT NULL,

PRIMARY KEY (`LOCATION\_ID`),

KEY `FK\_HAS\_CITY\_LOC` (`CITY\_CODE`)

)

2. INSERT

Jika ingin menambahkan/mengisi data pada tabel LOCATION : INSERT into LOCATION (`ADDRESS\_PLACE`) VALUES ()

3. SELECT

SELECT \* FROM LOCATION;

SELECT ADDRES\_LOCATION FROM LOCATION WHERE LOCATION\_ID=’’;

SS :

3. tabel CITY

|  |  |  |  |
| --- | --- | --- | --- |
| NAMA FIELD | TIPE DATA | FUNGSI | HUBUNGAN DENGAN TABEL LAIN |
| CITY\_CODE | CHAR |  | AKAN TURUN SEBAGAI FOREIGN KEY PADA TABEL PROVINCE |
| CITY\_NAME | VARCHAR |  | - |
| PROVINCE\_CODE |  |  |  |

QUERY-QUERY YANG AKAN DIPERLAKUKAN PADA TABEL :  
1. CREATE TABLE

CREATE TABLE CITY (‘CITY\_CODE CHAR() PRIMARY KEY NOT NULL’,’ CITY\_NAME VARCHAR() NOT NULL’)

2. INSERT

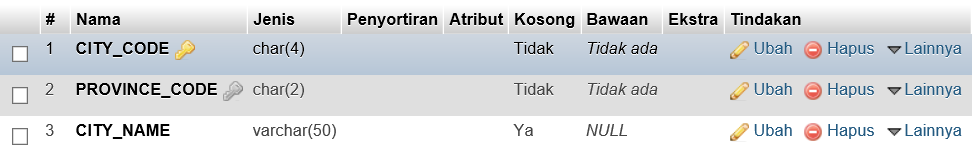
Jika ingin menambahkan data baru pada tabel CITY : INSERT into CITY (‘CITY\_CODE’,’ CITY\_NAME’) values ()

3. SELECT

Jika ingin menampilkan seluruh isi data dari tabel CITY : SELECT \* FROM CITY;

Jika ingin menampilkan nama kota saja dari tabel CITY : SELECT CITY\_NAME FROM CITY;

SS :



4. tabel PROVINCE

|  |  |  |  |
| --- | --- | --- | --- |
| NAMA FIELD | TIPE DATA | FUNGSI | HUBUNGAN DENGAN TABEL LAIN |
| PROVINCE\_CODE | CHAR |  |  |
| PROVINCE\_NAME | VARCHAR |  |  |

QUERY-QUERY YANG AKAN DIPERLAKUKAN PADA TABEL :  
1. . CREATE TABLE

CREATE TABLE PROVINCE (‘PROVINCE\_CODE

CHAR() PRIMARY KEY NOT NULL’,’ PROVINCE\_NAME VARCHAR() NOT NULL’)

2. INSERT

Jika ingin menambahkan data baru pada tabel PROVINCE : INSERT into CITY (‘PROVINCE\_CODE‘,’ PROVINCE\_NAME’) values ()

3. SELECT

Jika ingin menampilkan seluruh isi data dari tabel PROVINCE : SELECT \* FROM PROVINCE;

Jika ingin menampilkan nama provinsi dari tabel CITY : SELECT PROVINCE\_NAME FROM PROVINCE;

SS :



7. tabel distance

|  |  |  |  |
| --- | --- | --- | --- |
| NAMA FIELD | TIPE DATA | FUNGSI | HUBUNGAN DENGAN TABEL LAIN |
| distance | Float |  |  |

QUERY-QUERY YANG AKAN DIPERLAKUKAN PADA TABEL :

1. CREATE TABLE IF NOT EXISTS `distance` (

`START\_LOCATION\_ID` int(11) NOT NULL,

`END\_LOCATION\_ID` int(11) NOT NULL,

`DISTANCE` float DEFAULT NULL,

PRIMARY KEY (`START\_LOCATION\_ID`,`END\_LOCATION\_ID`),

KEY `FK\_END\_LOCATION` (`END\_LOCATION\_ID`)

)

2. INSERT

INSERT into DISTANCE (`DISTANCE`) VALUES ()

3. SELECT

SELECT \* FROM DISTANCE;

SELECT DISTANCE FROM DISTANCE;

8. tabel Travel

|  |  |  |  |
| --- | --- | --- | --- |
| NAMA FIELD | TIPE DATA | FUNGSI | HUBUNGAN DENGAN TABEL LAIN |
| LOCATION\_ID | Float |  |  |

1. CREATE TABLE IF NOT EXISTS `travel` (

`LOCATION\_ID` int(11) NOT NULL,

`EVENT\_ID` bigint(20) NOT NULL,

`TRANSPORTATION\_CODE` char(2) NOT NULL,

`DEPATURE\_TIME` datetime DEFAULT NULL,

PRIMARY KEY (`LOCATION\_ID`,`EVENT\_ID`),

KEY `FK\_NEED` (`TRANSPORTATION\_CODE`)

)

2. INSERT

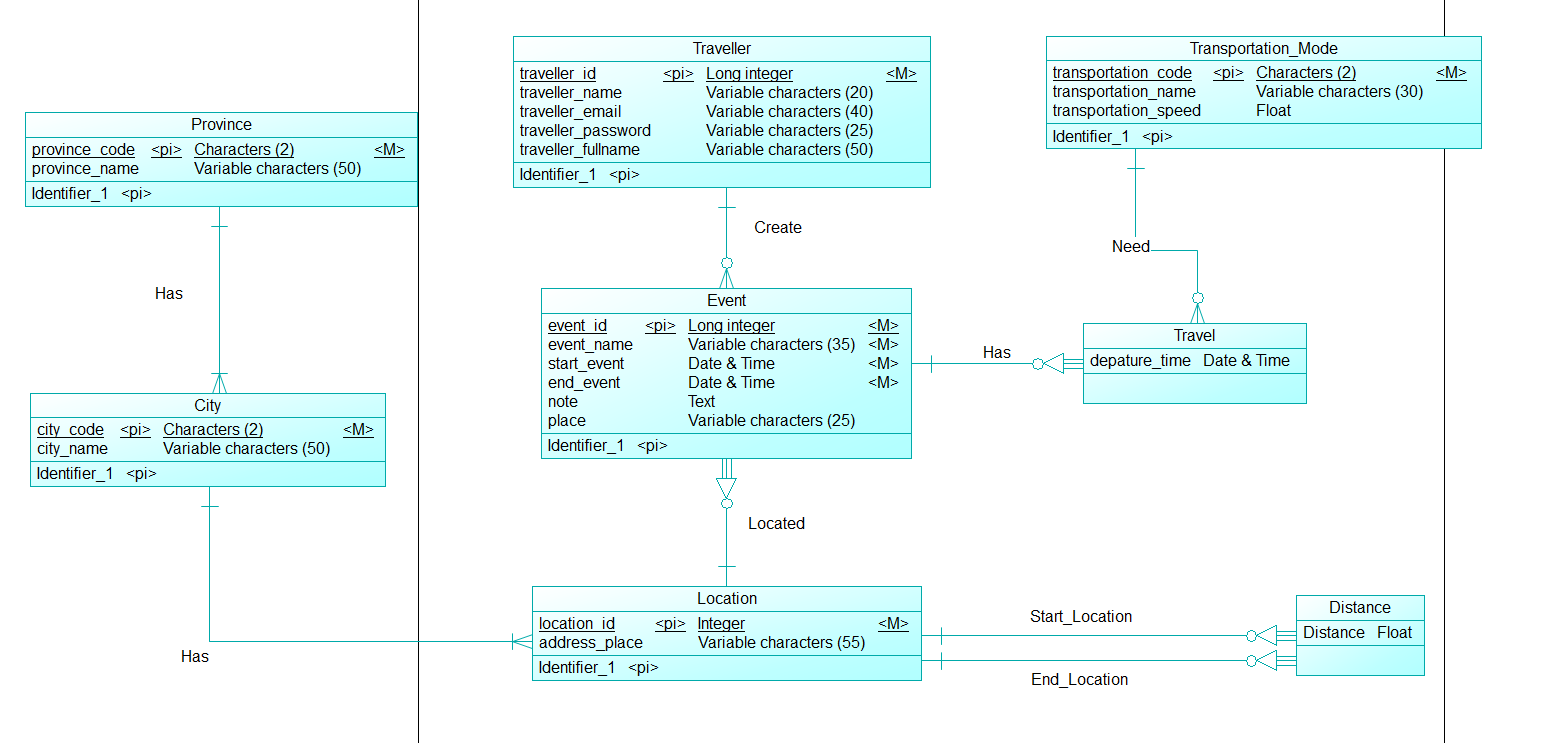
INSERT INTO TRAVEL (depature\_time) VALUES ()

3. SELECT

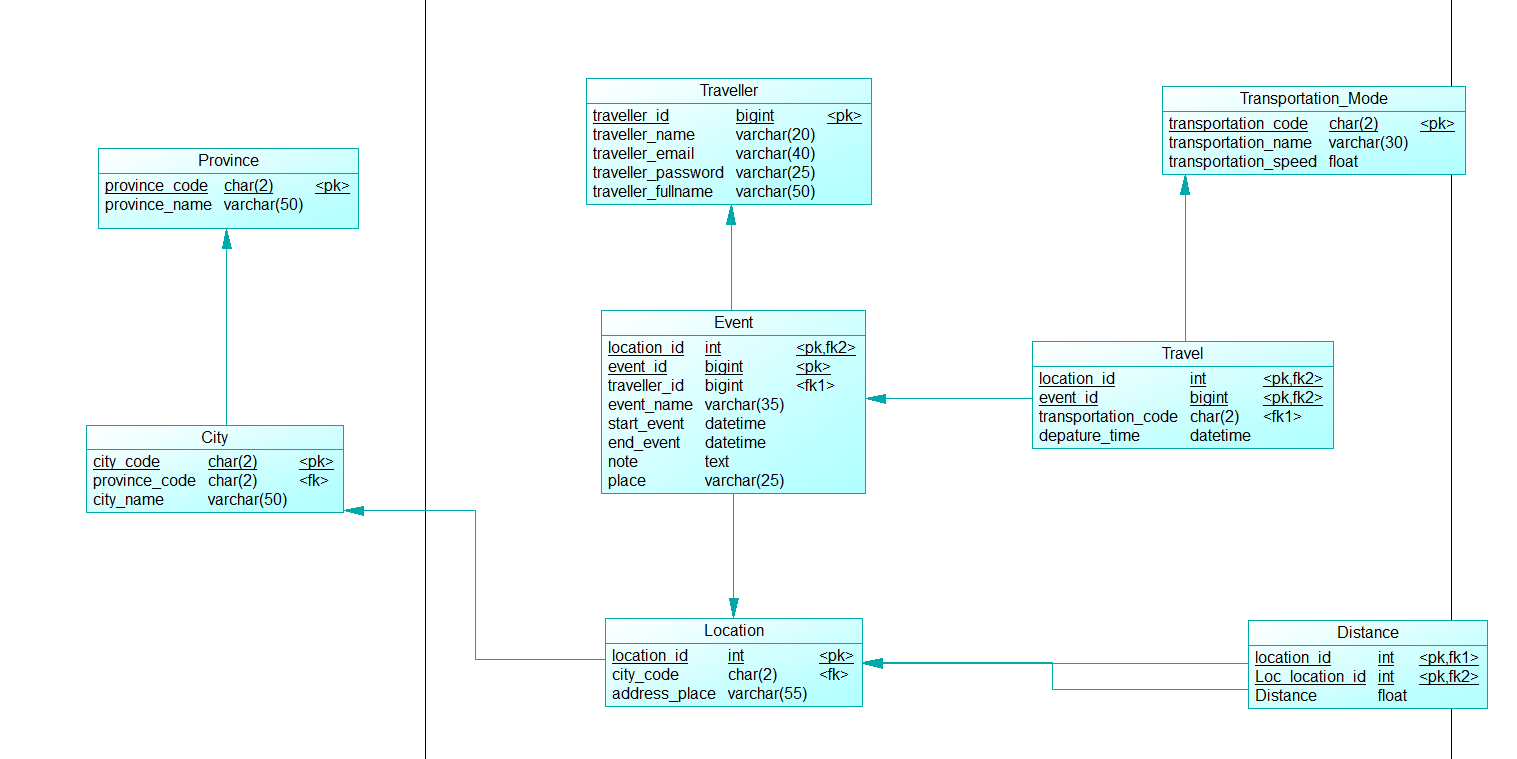
SELECT \* FROM TRAVEL;

**Model Data**

**Conspetual Database**



**Physical Database**



1. tabel TRANSPORTATION\_MODE

|  |  |  |  |
| --- | --- | --- | --- |
| NAMA FIELD | TIPE DATA | FUNGSI | HUBUNGAN DENGAN TABEL LAIN |
| TRANSPORTATION\_CODE | CHAR | YANG MENJADIKAN UNIK DAN SEBAGAI IDENTITAS DARI MASING-MAISNG KENDARAAN. | AKAN TURUN MENJADI FOREIGN KEY PADA TABLE TRAVEL |
| TRANSPORTATION\_NAME | VARCHAR | JENIS KENDARAAN | - |
| TRANSPORTATION\_SPEED | FLOAT | KECEPATAN KENDARAAN | - |

QUERY-QUERY YANG AKAN DIPERLAKUKAN PADA TABEL :  
1. CREATE TABLE

Query : CREATE table TRANSPORTATION\_MODE (‘TRANSPORTATION\_CODE CHAR() PRIMARY KEY NOT NULL’, ‘TRANSPORTATION\_NAME VARCHAR() NOT NULL’, ‘TRANSPORTATION\_SPEED FLOAT NOT NULL’);

2. INSERT

Jika ingin menambahkan transportasi baru tabel : INSERT into TRANSPORTATION\_MODE (‘TRANSPORTATION\_CODE’, ‘TRANSPORTATION\_NAME’, ‘TRANSPORTATION\_SPEED’) VALUES ()

3. SELECT

Jika ingin menampilkan seluruh isi data dari tabel TRANSPORTATION\_MODE : SELECT \* FROM TRANSPORTATION\_MODE;

Jika ingin menampilkan data kecepatan dari kendaraan mobil : SELECT TRANSPORTATION\_SPEED FROM TRANSPORTATION\_MODE WHERE TRANSPORTATION\_NAME=CAR;

Jika ingin menampilkan data kecepatan dari kendaraan motor : SELECT TRANSPORTATION\_SPEED FROM TRANSPORTATION\_MODE WHERE TRANSPORTATION\_NAME=MOTOR CYCLE;

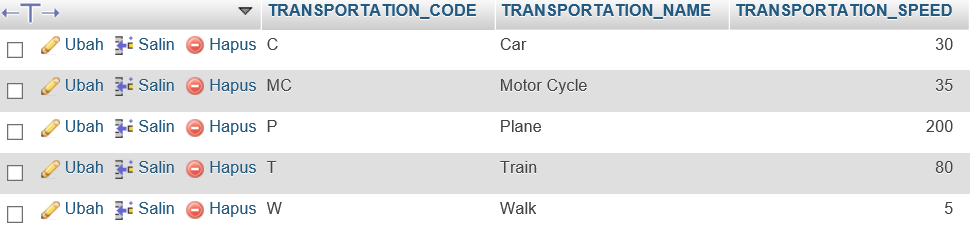
Jika ingin menampilkan data kecepatan dari kendaraan pesawat : SELECT TRANSPORTATION\_SPEED FROM TRANSPORTATION\_MODE WHERE TRANSPORTATION\_NAME=PLANE;

Jika ingin menampilkan data kecepatan dari kendaraan kereta : SELECT TRANSPORTATION\_SPEED FROM TRANSPORTATION\_MODE WHERE TRANSPORTATION\_NAME=TRAIN;

Jika ingin menampilkan data kecepatan jika berjalan kaki : SELECT TRANSPORTATION\_SPEED FROM TRANSPORTATION\_MODE WHERE TRANSPORTATION\_NAME=WALK;

SS :





2. tabel EVENT

|  |  |  |  |
| --- | --- | --- | --- |
| NAMA FIELD | TIPE DATA | FUNGSI | HUBUNGAN DENGAN TABEL LAIN |
| EVENT\_ID | Long integer |  | AKAN TURUN MENJADI FOREIGN KEY PADA TABLE TRAVEL |
| EVENT\_NAME | Variable characters |  | - |
| START\_EVENT | DATE & TIME |  | - |
| END\_EVENT | DATE & TIME |  | - |
| NOTE | TEXT |  | - |
| PLACE | VARCHAR |  | - |

QUERY-QUERY YANG AKAN DIPERLAKUKAN PADA TABEL :  
1. CREATE TABLE

CREATE TABLE EVENT (‘EVENT\_ID Long integer() PRIMARY KEY AUTO INCREMENT NOT NULL ’,’EVENT\_NAME Variable characters() NOT NULL’,’START\_EVENT DATE & TIME() NOT NULL ’,’END\_EVENT DATE & TIME() NOT NULL’,’NOTE TEXT()’,’PLACE’)

2. INSERT

Jika ingin mengisi/menambah data pada tabel EVENT : INSERT into EVENT(‘EVENT\_ID’,’ EVENT\_NAME’,’ START\_EVENT’,’ END\_EVENT’,’ NOTE’,’ PLACE’) VALUES ()

3. SELECT

Jika ingin menampilkan seluruh isi data dari tabel EVENT : SELECT \* FROM EVENT;

Jika ingin menampilkan seluruh data tabel dimana jika yang di ketahui nya id : SELECT \* FROM EVENT WHERE EVENT\_ID=1;

SS :



3. tabel CITY

|  |  |  |  |
| --- | --- | --- | --- |
| NAMA FIELD | TIPE DATA | FUNGSI | HUBUNGAN DENGAN TABEL LAIN |
| CITY\_CODE | CHAR |  | AKAN TURUN SEBAGAI FOREIGN KEY PADA TABEL PROVINCE |
| CITY\_NAME | VARCHAR |  | - |
| PROVINCE\_CODE |  |  |  |

QUERY-QUERY YANG AKAN DIPERLAKUKAN PADA TABEL :  
1. CREATE TABLE

CREATE TABLE CITY (‘CITY\_CODE CHAR() PRIMARY KEY NOT NULL’,’ CITY\_NAME VARCHAR() NOT NULL’)

2. INSERT

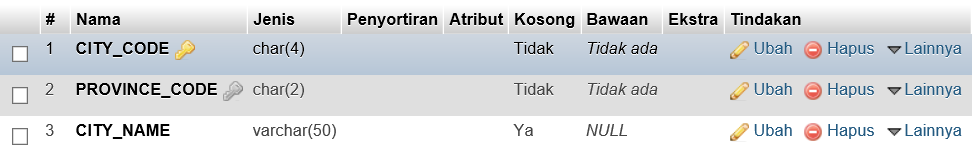
Jika ingin menambahkan data baru pada tabel CITY : INSERT into CITY (‘CITY\_CODE’,’ CITY\_NAME’) values ()

3. SELECT

Jika ingin menampilkan seluruh isi data dari tabel CITY : SELECT \* FROM CITY;

Jika ingin menampilkan nama kota saja dari tabel CITY : SELECT CITY\_NAME FROM CITY;

SS :



4. tabel PROVINCE

|  |  |  |  |
| --- | --- | --- | --- |
| NAMA FIELD | TIPE DATA | FUNGSI | HUBUNGAN DENGAN TABEL LAIN |
| PROVINCE\_CODE | CHAR |  | Akan turun menjadi foreign key pada tabel city |
| PROVINCE\_NAME | VARCHAR |  | - |

QUERY-QUERY YANG AKAN DIPERLAKUKAN PADA TABEL :  
1. . CREATE TABLE

CREATE TABLE PROVINCE (‘PROVINCE\_CODE

CHAR() PRIMARY KEY NOT NULL’,’ PROVINCE\_NAME VARCHAR() NOT NULL’)

2. INSERT

Jika ingin menambahkan data baru pada tabel PROVINCE : INSERT into CITY (‘PROVINCE\_CODE‘,’ PROVINCE\_NAME’) values ()

3. SELECT

Jika ingin menampilkan seluruh isi data dari tabel PROVINCE : SELECT \* FROM PROVINCE;

Jika ingin menampilkan nama provinsi dari tabel CITY : SELECT PROVINCE\_NAME FROM PROVINCE;

SS :



5. tabel TRAVELLER

|  |  |  |  |
| --- | --- | --- | --- |
| NAMA FIELD | TIPE DATA | FUNGSI | HUBUNGAN DENGAN TABEL LAIN |
| TRAVELLER\_ID | LONG INTEGER |  | Akan turun menjadi foreign key pada tabel travel |
| TRAVELLER\_NAME | VARCHAR |  |  |
| TRAVELLER\_EMAIL | VARCHAR |  |  |
| TRAVELLER\_PASSWORD | VARCHAR |  |  |
| TRAVELLER\_FULLNAME | VARCHAR |  |  |

QUERY-QUERY YANG AKAN DIPERLAKUKAN PADA TABEL :  
1.

CREATE TABLE IF NOT EXISTS `traveller` (

`TRAVELLER\_ID` bigint(20) NOT NULL AUTO\_INCREMENT,

`TRAVELLER\_NAME` varchar(20) DEFAULT NULL,

`TRAVELLER\_EMAIL` varchar(40) DEFAULT NULL,

`TRAVELLER\_PASSWORD` varchar(25) DEFAULT NULL,

`TRAVELLER\_FULLNAME` varchar(50) DEFAULT NULL,

PRIMARY KEY (`TRAVELLER\_ID`)

)

2. INSERT

Jika ingin menambahkan/mengisi data pada tabel TRAVELLER : INSERT into TRAVELLER ( `TRAVELLER\_ID`, `TRAVELLER\_ID`,`TRAVELLER\_EMAIL`, `TRAVELLER\_EMAIL`, `TRAVELLER\_PASSWORD`, `TRAVELLER\_FULLNAME`) VALUES ()

3. SELECT

Jika ingin menampilkan semua data di tabel TRAVELLER : SELECT\* FROM TRAVELLER;

SELECT TRAVELLER\_NAME FROM TRAVELLER WHERE TRAVELLER\_ID=1;

SS :



6. tabel LOCATION

|  |  |  |  |
| --- | --- | --- | --- |
| NAMA FIELD | TIPE DATA | FUNGSI | HUBUNGAN DENGAN TABEL LAIN |
| **LOCATION\_IDKunci Utama** | INT |  | Akan turun menjadi foreign key pada tabel event |
| **CITY\_CODE** | CHAR |  |  |
| **ADDRESS\_PLACE** | VARCHAR |  |  |

QUERY-QUERY YANG AKAN DIPERLAKUKAN PADA TABEL :  
1. Create

CREATE TABLE IF NOT EXISTS `location` (

`LOCATION\_ID` int(11) NOT NULL AUTO\_INCREMENT,

`CITY\_CODE` char(2) NOT NULL,

`ADDRESS\_PLACE` varchar(55) DEFAULT NULL,

PRIMARY KEY (`LOCATION\_ID`),

KEY `FK\_HAS\_CITY\_LOC` (`CITY\_CODE`)

)

2. INSERT

Jika ingin menambahkan/mengisi data pada tabel LOCATION : INSERT into LOCATION (`ADDRESS\_PLACE`) VALUES ()

3. SELECT

SELECT \* FROM LOCATION;

SELECT ADDRES\_LOCATION FROM LOCATION WHERE LOCATION\_ID=’’;

SS :

7. tabel distance

|  |  |  |  |
| --- | --- | --- | --- |
| NAMA FIELD | TIPE DATA | FUNGSI | HUBUNGAN DENGAN TABEL LAIN |
| distance | Float |  | Hasil data penghitungan jarak antara 2 lokasi di tabel location akan disimpan disini |

QUERY-QUERY YANG AKAN DIPERLAKUKAN PADA TABEL :

1. CREATE TABLE IF NOT EXISTS `distance` (

`START\_LOCATION\_ID` int(11) NOT NULL,

`END\_LOCATION\_ID` int(11) NOT NULL,

`DISTANCE` float DEFAULT NULL,

PRIMARY KEY (`START\_LOCATION\_ID`,`END\_LOCATION\_ID`),

KEY `FK\_END\_LOCATION` (`END\_LOCATION\_ID`)

)

2. INSERT

INSERT into DISTANCE (`DISTANCE`) VALUES ()

3. SELECT

SELECT \* FROM DISTANCE;

SELECT DISTANCE FROM DISTANCE;

8. tabel Travel

|  |  |  |  |
| --- | --- | --- | --- |
| NAMA FIELD | TIPE DATA | FUNGSI | HUBUNGAN DENGAN TABEL LAIN |
| LOCATION\_ID | Float |  |  |
| EVENT\_ID |  |  |  |

1. CREATE TABLE IF NOT EXISTS `travel` (

`LOCATION\_ID` int(11) NOT NULL,

`EVENT\_ID` bigint(20) NOT NULL,

`TRANSPORTATION\_CODE` char(2) NOT NULL,

`DEPATURE\_TIME` datetime DEFAULT NULL,

PRIMARY KEY (`LOCATION\_ID`,`EVENT\_ID`),

KEY `FK\_NEED` (`TRANSPORTATION\_CODE`)

)

2. INSERT

INSERT INTO TRAVEL (depature\_time) VALUES ()

3. SELECT

SELECT \* FROM TRAVEL;