**ApiBarang**

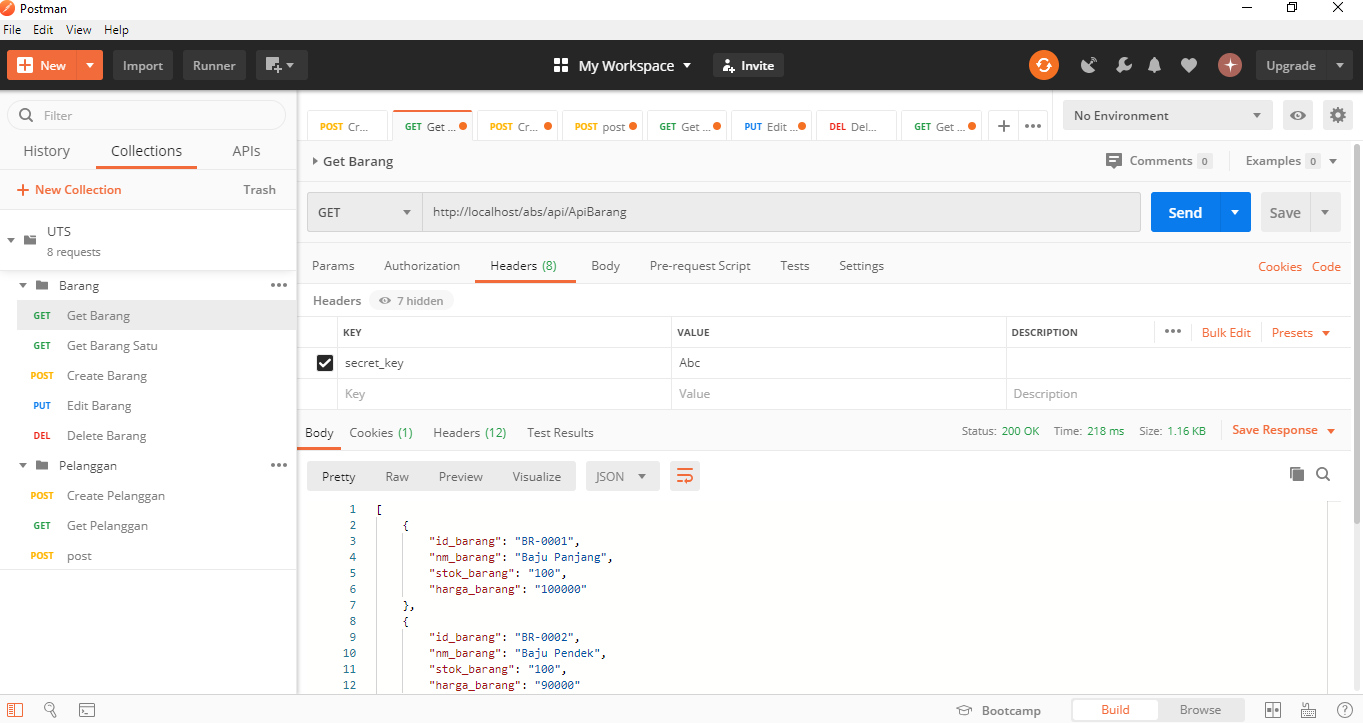
**Function Index**

Function index. Dengan memanfaatkan switch case Maka kita dapat mengalihkan ke Req Method sudah dibuat. $req sebagai variable. Yang mana req dapat menjadi GET , POST, PUT atau DELETE.

|  |
| --- |
| public function index()  {  $req = $\_SERVER['REQUEST\_METHOD'];    switch ($req) {  case 'GET':  ….  case 'POST':  ….  case 'PUT':  ….  case 'DELETE':  …. |

///////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

**Get semua barang**

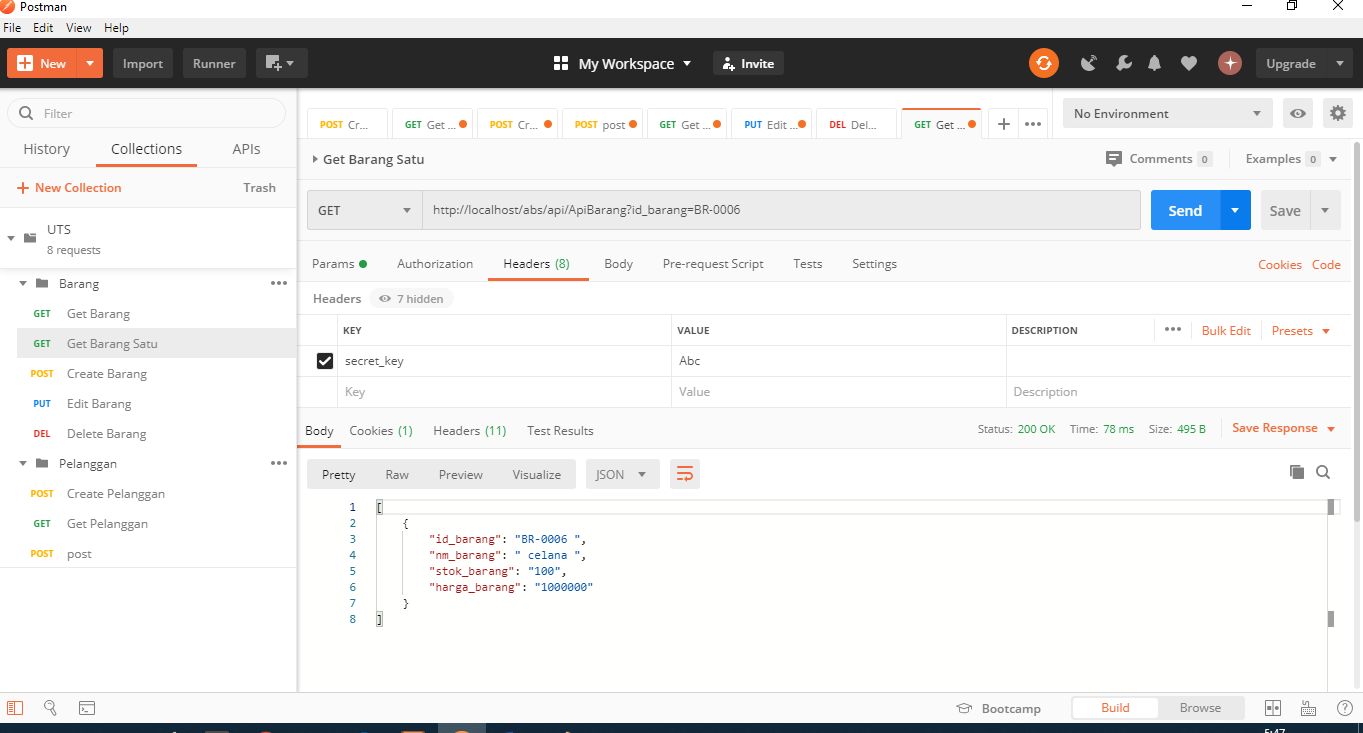
­

Digunakan untuk membaca resource dari REST server. Pada gambar, jika id\_barang != maka akan memunculkan semua data pada table dengan memanggil id pada table.

Parameter : http://localhost/abs/api/ApiBarang

|  |
| --- |
| case 'GET':  $data = $this->Model\_barang->show\_barang()->result();  if ($this->input->get('id\_barang') != '') {  $id\_barang = $this->input->get('id\_barang');  $data = $this->Model\_barang->show\_one\_barang($id\_barang)->result();  }  echo json\_encode($data);  break; |

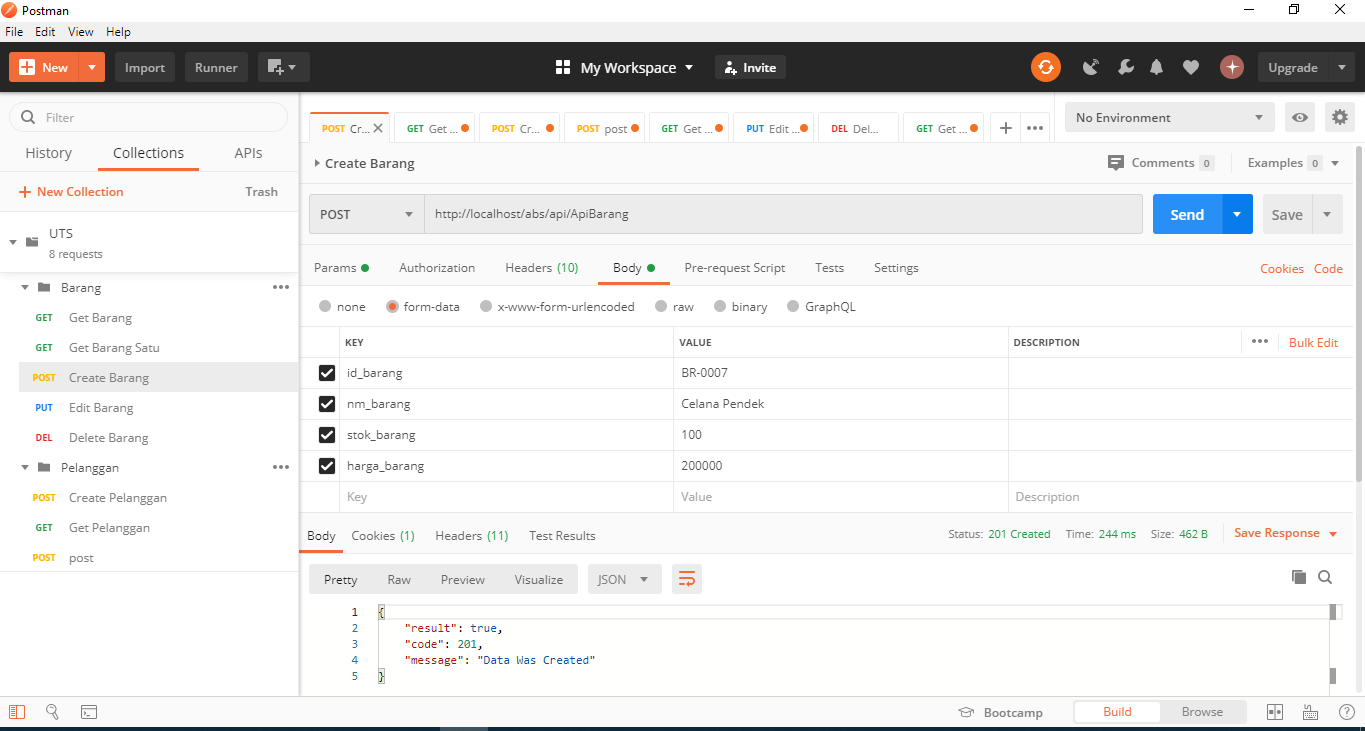
**Get satu barang**



Digunakan untuk membaca resource dari REST server. Pada gambar, jika id\_barang = x maka akan memunculkan data x pada table.

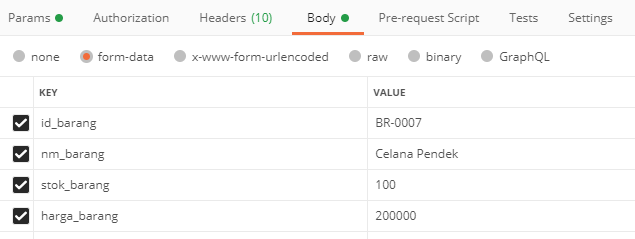
Parameter : <http://localhost/abs/api/ApiBarang?id_barang=BR-0006>

**Post barang**



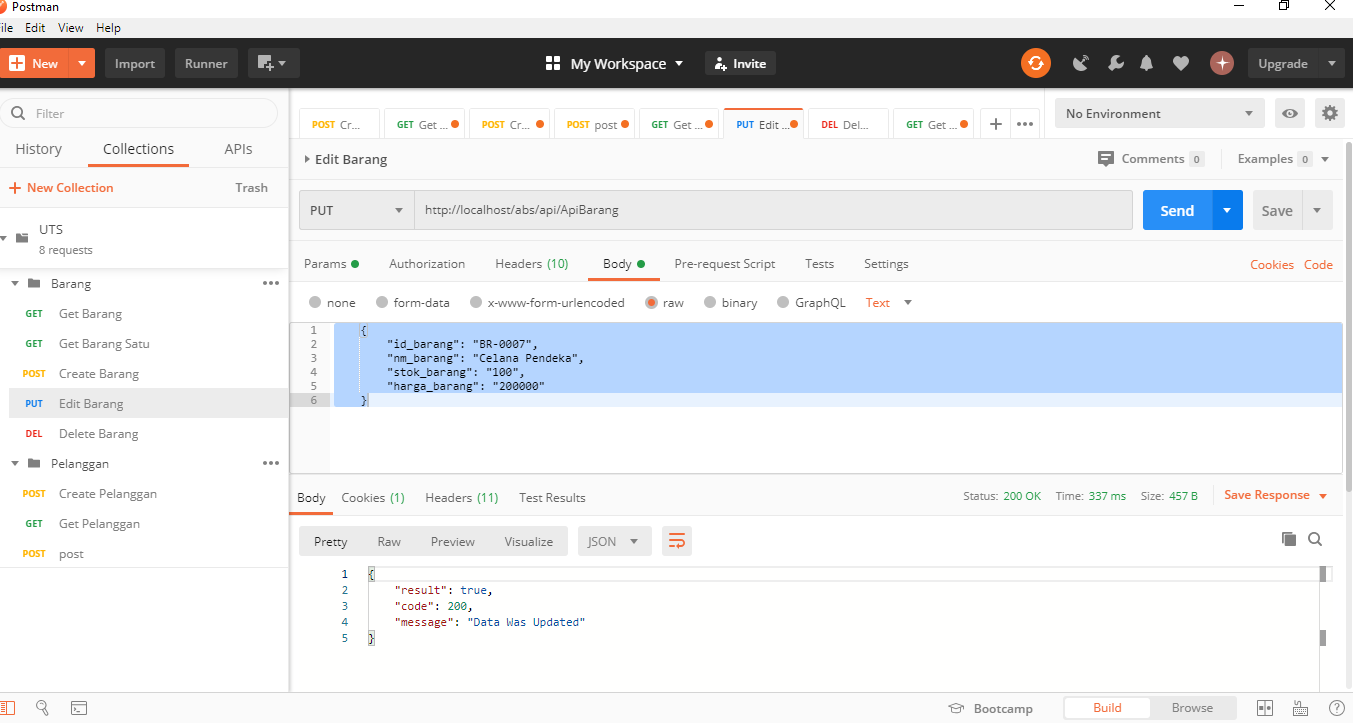
Digunakan untuk membuat resource baru di REST server. Menggunakan Body, cara kerja mengisi key dengan sesuai nama yang ada di field database dan mengisi value.

Parameter : http://localhost/abs/api/ApiBarang



|  |
| --- |
| case 'POST':  $data = $this->Model\_barang->add();  http\_response\_code(500);  $arrResult = array(  'result' => false,  'code' => 500,  'message' => 'Internal Server Error'  );  if ($data == 1) {  http\_response\_code(201);  $arrResult = array(  'result' => true,  'code' => 201,  'message' => 'Data Was Created'  );    }  echo json\_encode($arrResult);  break; |

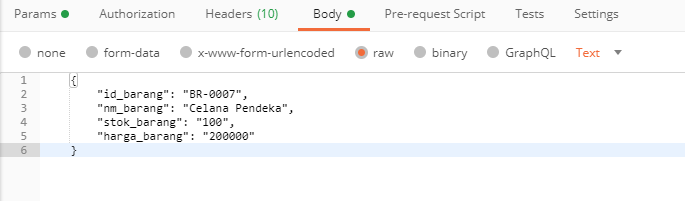
**PUT BARANG**



Digunakan untuk memperbaharui resource di REST server. Menggunakan Body, cara kerja mengganti value dengan sesuai field yang ada.

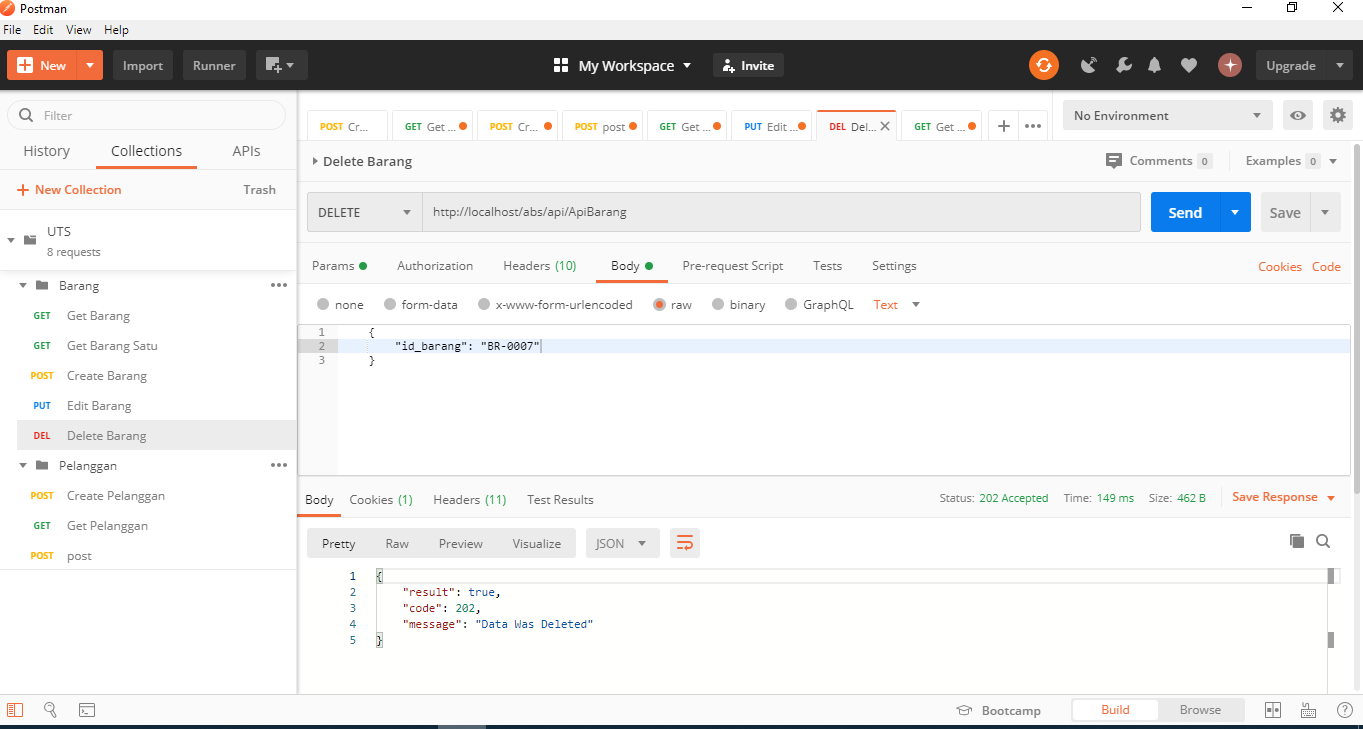
Muncul pesan 200 itu artinya bahwa request yang dilakukan berhasil.

Parameter : http://localhost/abs/api/ApiBarang



|  |
| --- |
| case 'PUT':  $putfp = fopen('php://input', 'r');  $putdata = '';  while($data = fread($putfp, 1024))  $putdata .= $data;  fclose($putfp);  $var = json\_decode($putdata);  foreach ($var as $key => $value) {  $$key = $value;  }  http\_response\_code(404);  $arrResult = array(  'result' => false,  'code' => 404,  'message' => 'Data Not Found'  );  $barang = $this->Model\_barang->show\_one\_barang($id\_barang)->result();  $data = $this->Model\_barang->update($id\_barang, $nm\_barang, $stok\_barang, $harga\_barang);  if (count($barang) == 1) {  http\_response\_code(200);  $arrResult = array(  'result' => true,  'code' => 200,  'message' => 'Data Was Updated'  );  }  echo json\_encode($arrResult);  break; |

**DELETE BARANG**



Digunakan untuk menghapus resource dari REST server. Dengan mengisi id beserta value field id yang mau dihapus.

Pesan code 202 artinya req diterima



|  |
| --- |
| case 'DELETE':  $putfp = fopen('php://input', 'r');  $putdata = '';  while($data = fread($putfp, 1024))  $putdata .= $data;  fclose($putfp);  $var = json\_decode($putdata);  foreach ($var as $key => $value) {  $$key = $value;  }  http\_response\_code(404);  $arrResult = array(  'result' => false,  'code' => 404,  'message' => 'Data Not Found'  );  $barang = $this->Model\_barang->show\_one\_barang($id\_barang)->result();  if (count($barang) == 1) {  $data = $this->Model\_barang->delete($id\_barang);  http\_response\_code(202);  $arrResult = array(  'result' => true,  'code' => 202,  'message' => 'Data Was Deleted'  );  }  echo json\_encode($arrResult);  break; |

Full Script

|  |
| --- |
| public function index()  {  $req = $\_SERVER['REQUEST\_METHOD'];    switch ($req) {  case 'GET':  $data = $this->Model\_barang->show\_barang()->result();  if ($this->input->get('id\_barang') != '') {  $id\_barang = $this->input->get('id\_barang');  $data = $this->Model\_barang->show\_one\_barang($id\_barang)->result();  }  echo json\_encode($data);  break;  case 'POST':  $data = $this->Model\_barang->add();  http\_response\_code(500);  $arrResult = array(  'result' => false,  'code' => 500,  'message' => 'Internal Server Error'  );  if ($data == 1) {  http\_response\_code(201);  $arrResult = array(  'result' => true,  'code' => 201,  'message' => 'Data Was Created'  );    }  echo json\_encode($arrResult);  break;  case 'PUT':  $putfp = fopen('php://input', 'r');  $putdata = '';  while($data = fread($putfp, 1024))  $putdata .= $data;  fclose($putfp);  $var = json\_decode($putdata);  foreach ($var as $key => $value) {  $$key = $value;  }  http\_response\_code(404);  $arrResult = array(  'result' => false,  'code' => 404,  'message' => 'Data Not Found'  );  $barang = $this->Model\_barang->show\_one\_barang($id\_barang)->result();  $data = $this->Model\_barang->update($id\_barang, $nm\_barang, $stok\_barang, $harga\_barang);  if (count($barang) == 1) {  http\_response\_code(200);  $arrResult = array(  'result' => true,  'code' => 200,  'message' => 'Data Was Updated'  );  }  echo json\_encode($arrResult);  break;  case 'DELETE':  $putfp = fopen('php://input', 'r');  $putdata = '';  while($data = fread($putfp, 1024))  $putdata .= $data;  fclose($putfp);  $var = json\_decode($putdata);  foreach ($var as $key => $value) {  $$key = $value;  }  http\_response\_code(404);  $arrResult = array(  'result' => false,  'code' => 404,  'message' => 'Data Not Found'  );  $barang = $this->Model\_barang->show\_one\_barang($id\_barang)->result();  if (count($barang) == 1) {  $data = $this->Model\_barang->delete($id\_barang);  http\_response\_code(202);  $arrResult = array(  'result' => true,  'code' => 202,  'message' => 'Data Was Deleted'  );  }  echo json\_encode($arrResult);  break;  }  } |

BEBERAPA KODE STATUS HTTP

200 OK

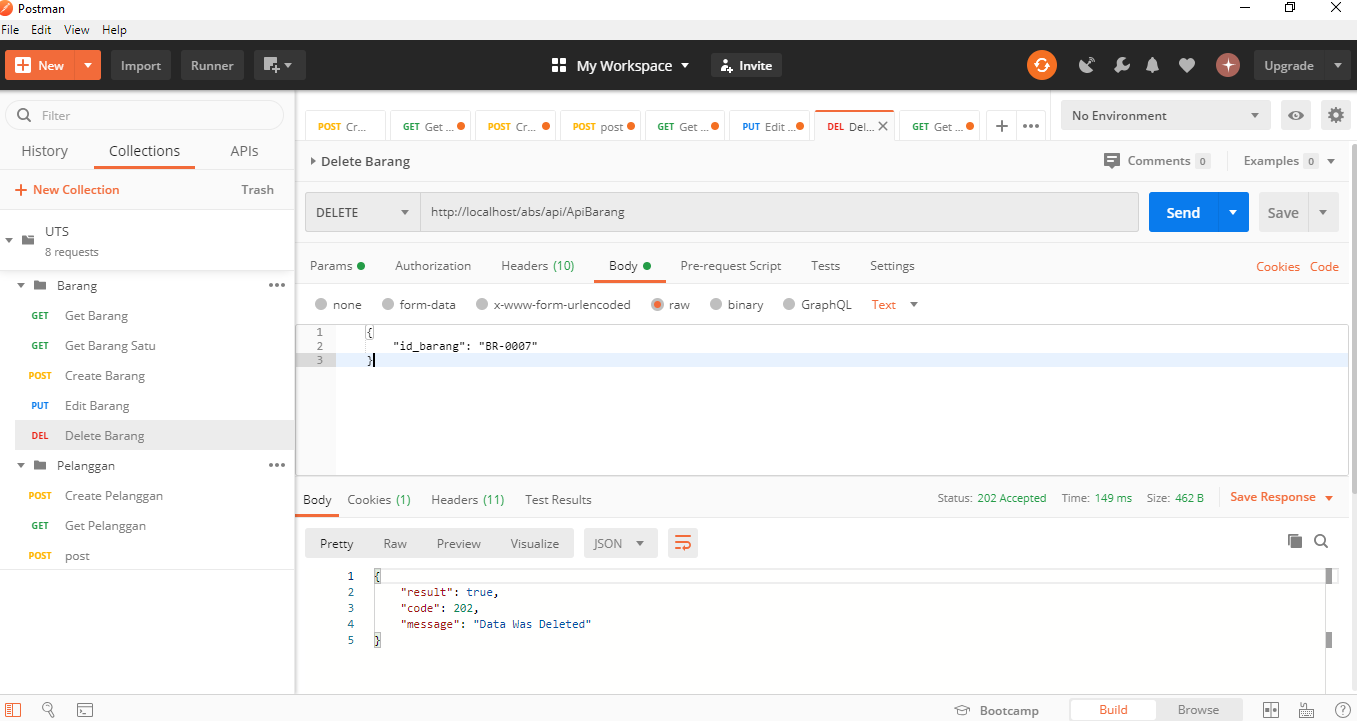
Response code ini menandakan bahwa request yang dilakukan berhasil.

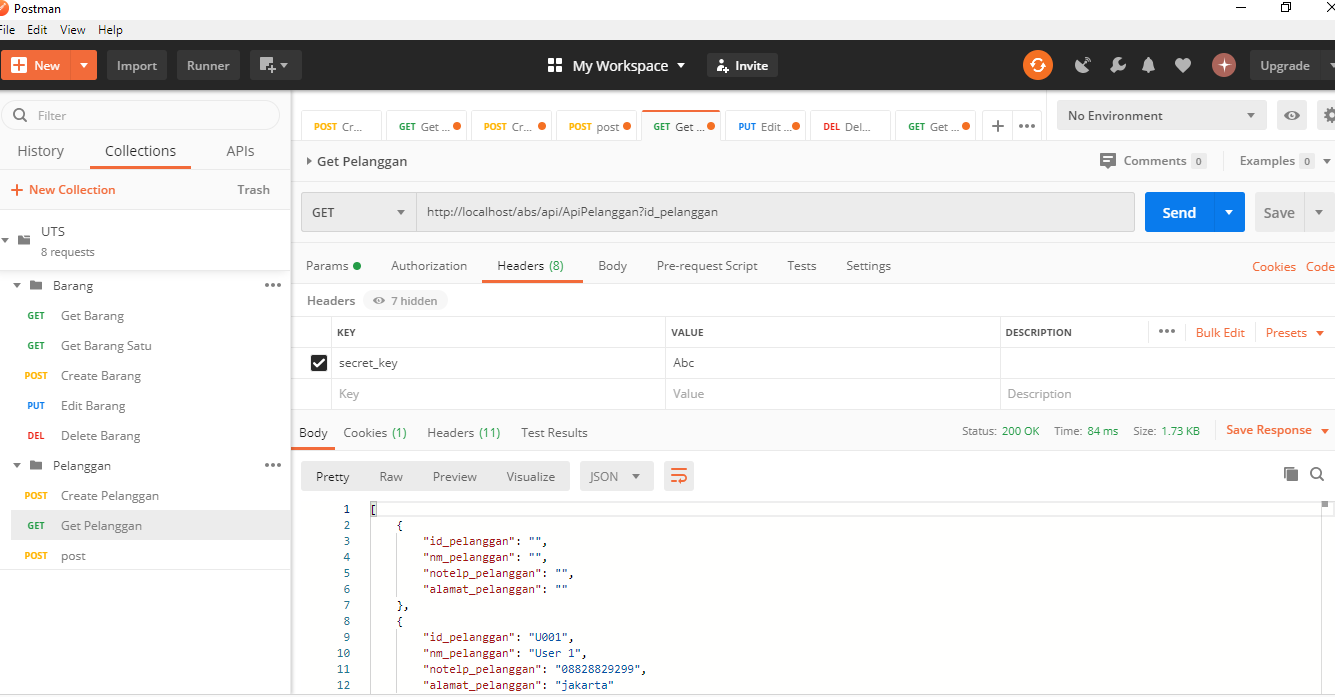
201 Created

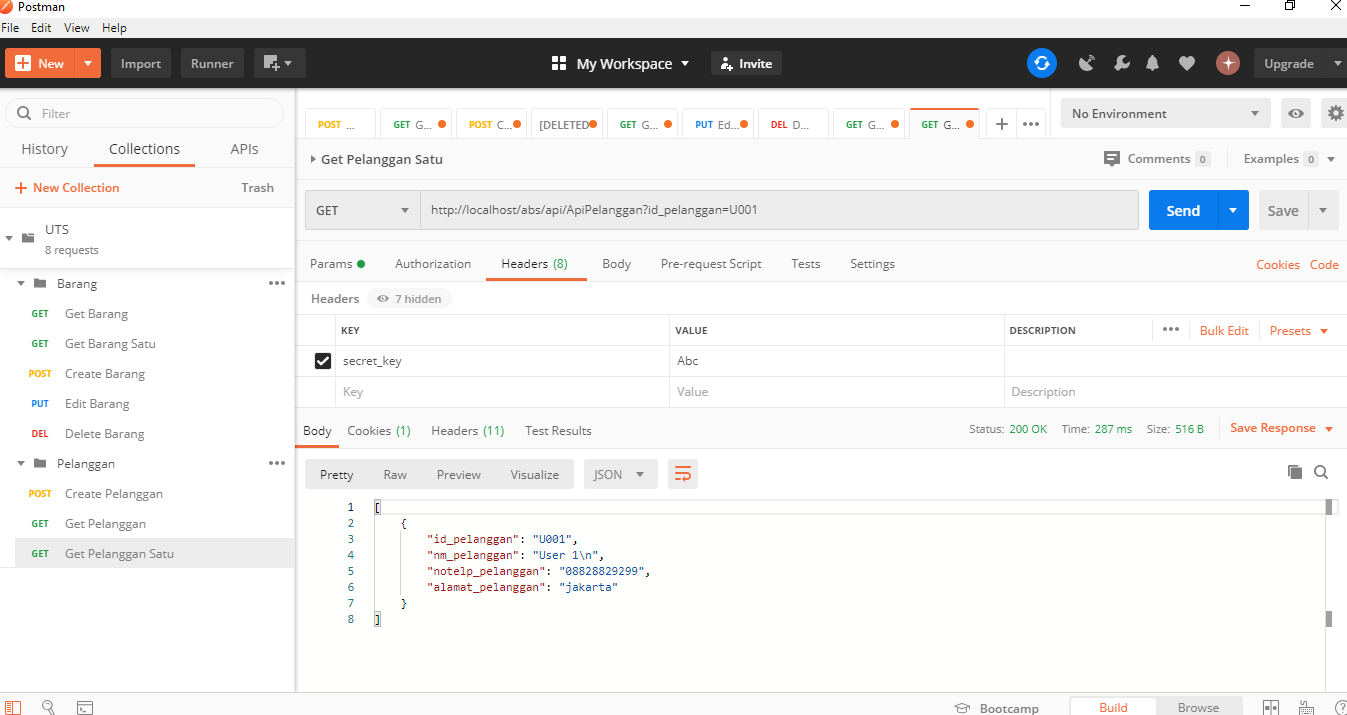
Response code ini menandakan bahwa request yang dilakukan berhasil dan data telah dibuat. Kode ini digunakan untuk mengkonfirmasi berhasilnya request PUT atau POST.

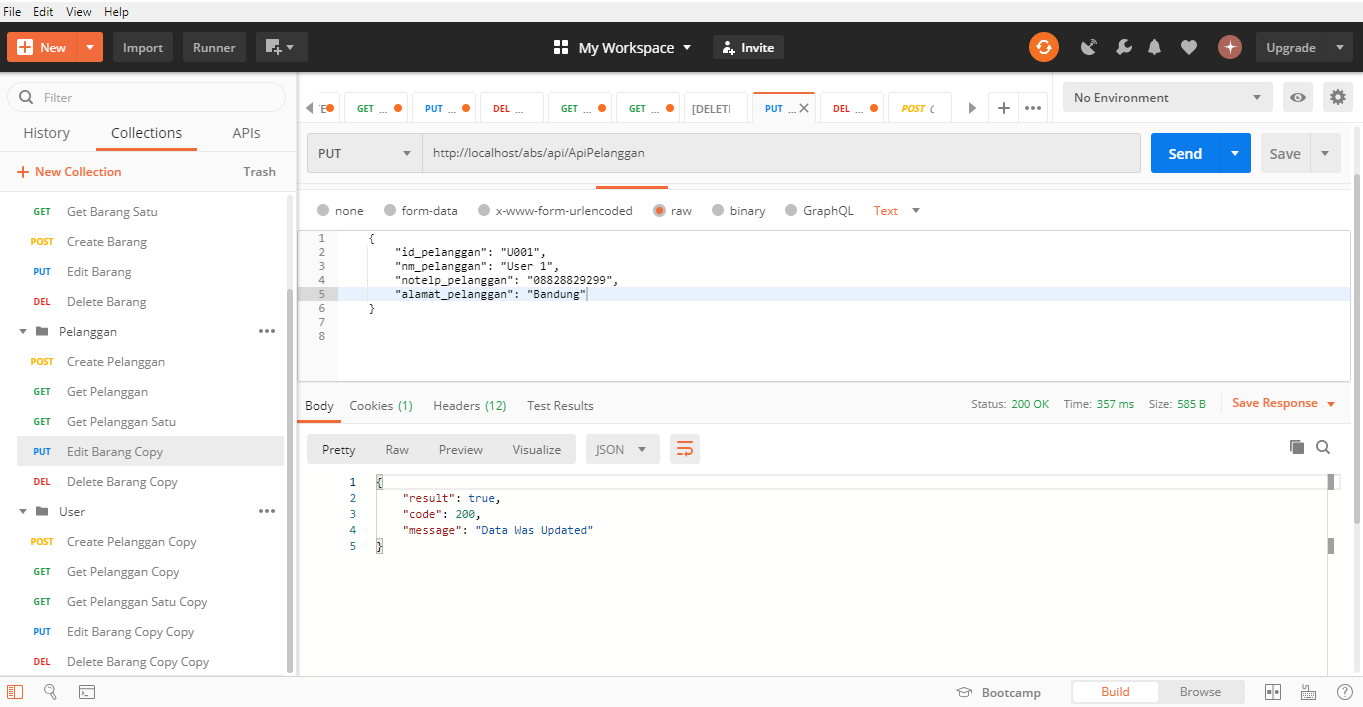
400 Bad Request

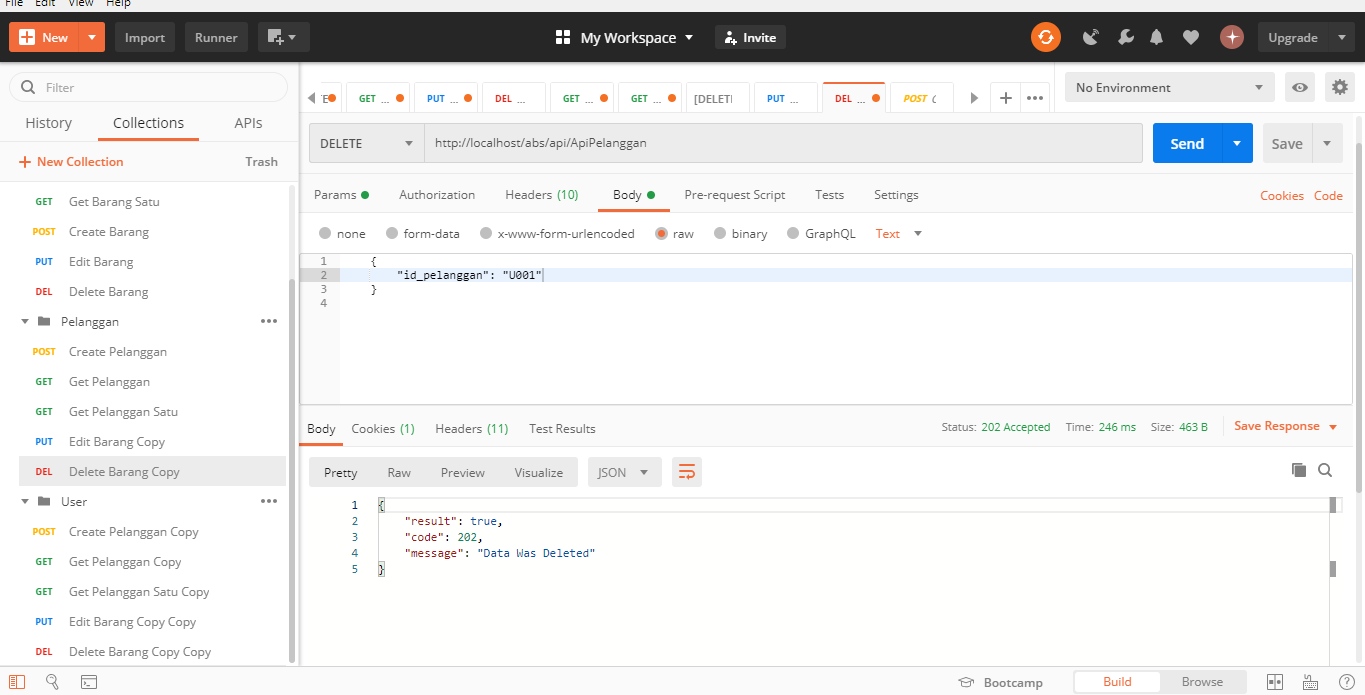
Response code ini menandakan bahwa request yang dibuat salah atau data yang dikirim tidak ada.







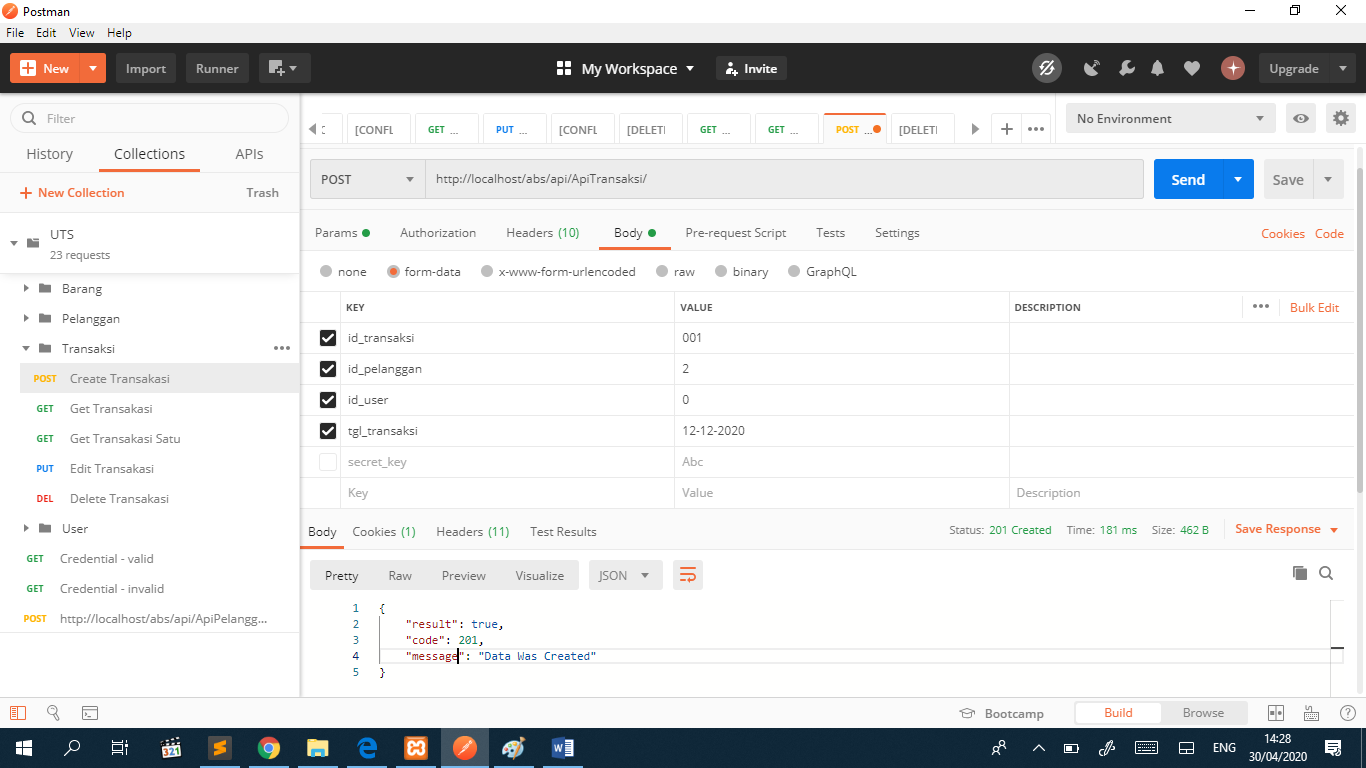




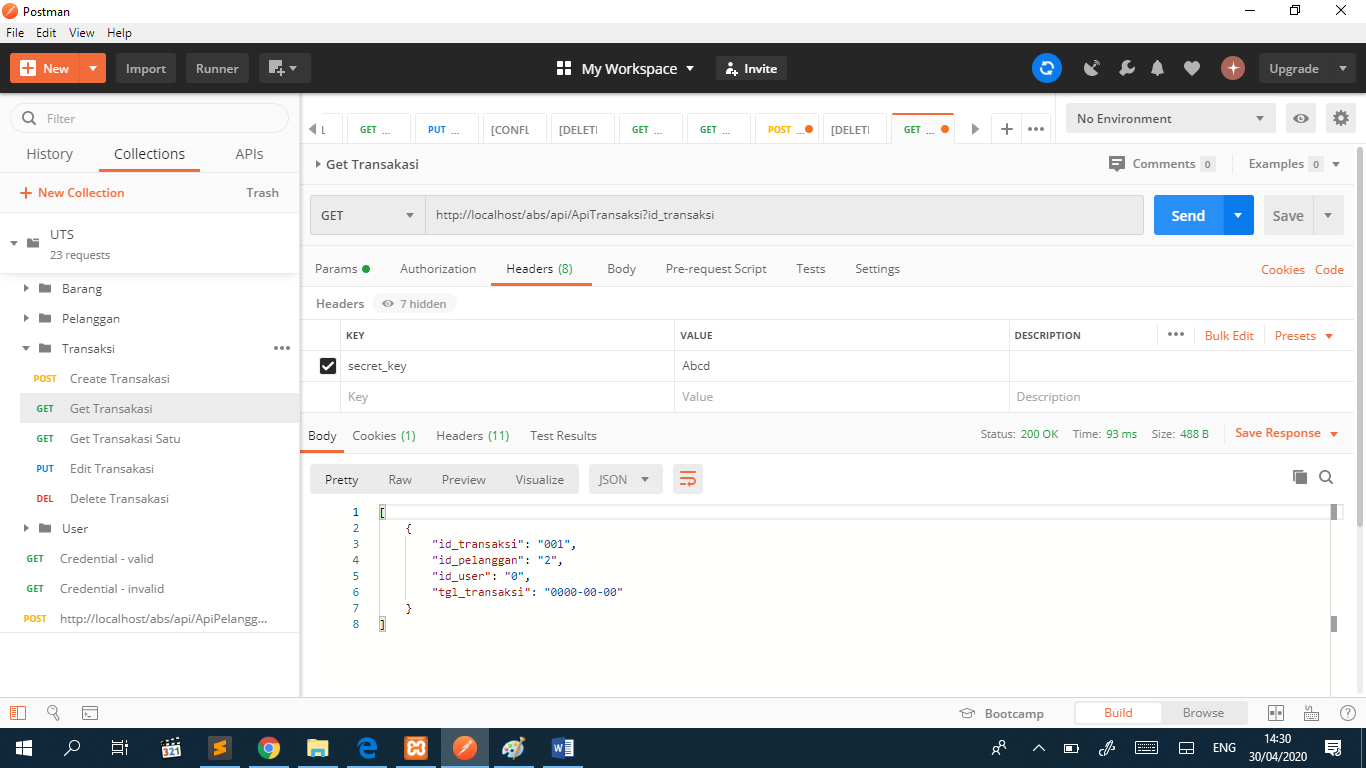
API USER

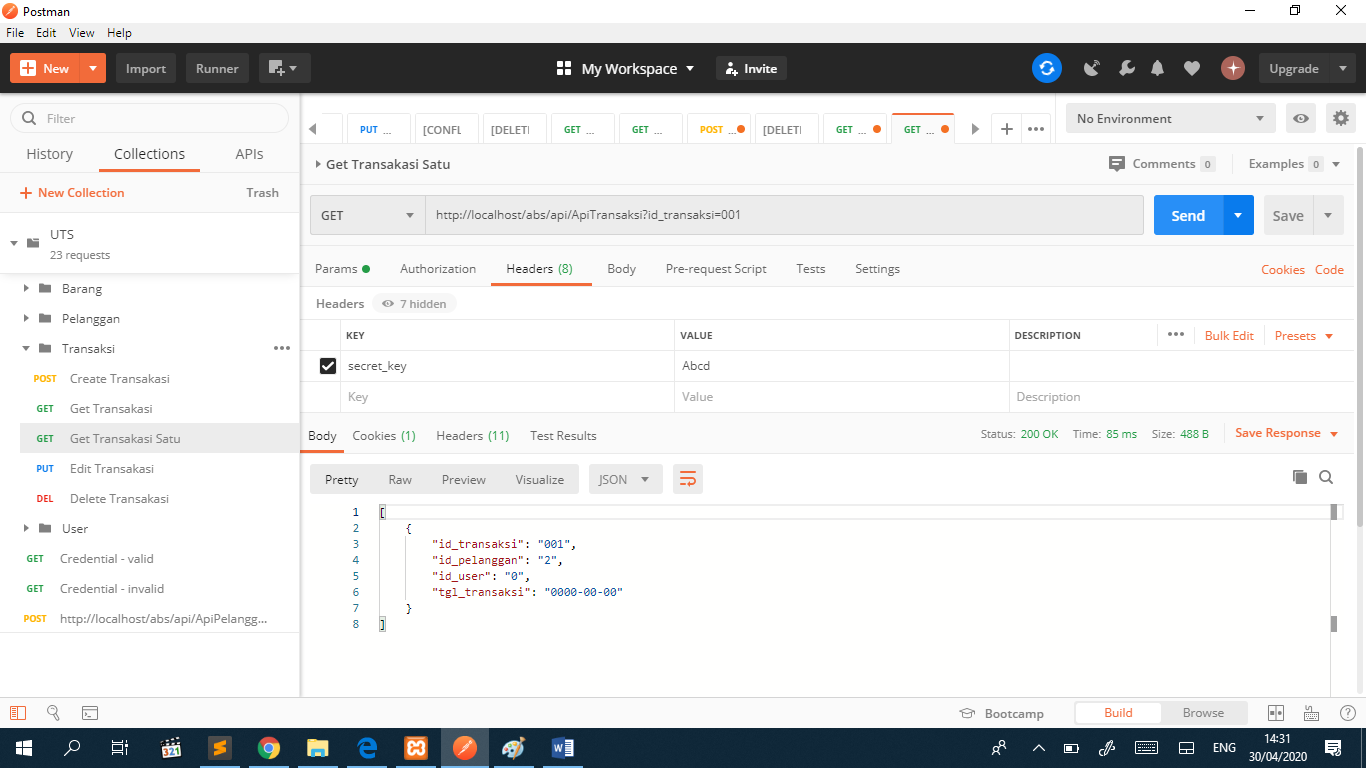
|  |
| --- |
| case 'POST':  $data = $this->Model\_user->add();  http\_response\_code(500);  $arrResult = array(  'result' => false,  'code' => 500,  'message' => 'Internal Server Error'  );  if ($data == 1) {  http\_response\_code(201);  $arrResult = array(  'result' => true,  'code' => 201,  'message' => 'Data Was Created'  );    }  echo json\_encode($arrResult);  break;  case 'PUT':  $putfp = fopen('php://input', 'r');  $putdata = '';  while($data = fread($putfp, 1024))  $putdata .= $data;  fclose($putfp);  $var = json\_decode($putdata);  foreach ($var as $key => $value) {  $$key = $value;  }  http\_response\_code(404);  $arrResult = array(  'result' => false,  'code' => 404,  'message' => 'Data Not Found'  );  $barang = $this->Model\_user->show\_one\_user($id\_user)->result();  $data = $this->Model\_user->update($id\_user, $nm\_user, $stok\_user, $harga\_user, $email\_user, $level\_user, $sk\_user);  if (count($barang) == 1) {  http\_response\_code(200);  $arrResult = array(  'result' => true,  'code' => 200,  'message' => 'Data Was Updated'  );  }  echo json\_encode($arrResult);  break;  case 'DELETE':  $putfp = fopen('php://input', 'r');  $putdata = '';  while($data = fread($putfp, 1024))  $putdata .= $data;  fclose($putfp);  $var = json\_decode($putdata);  foreach ($var as $key => $value) {  $$key = $value;  }  http\_response\_code(404);  $arrResult = array(  'result' => false,  'code' => 404,  'message' => 'Data Not Found'  );  $barang = $this->Model\_user->show\_one\_user($id\_user)->result();  if (count($barang) == 1) {  $data = $this->Model\_user->delete($id\_user);  http\_response\_code(202);  $arrResult = array(  'result' => true,  'code' => 202,  'message' => 'Data Was Deleted'  );  }  echo json\_encode($arrResult);  break;  }  }  }  ?> |

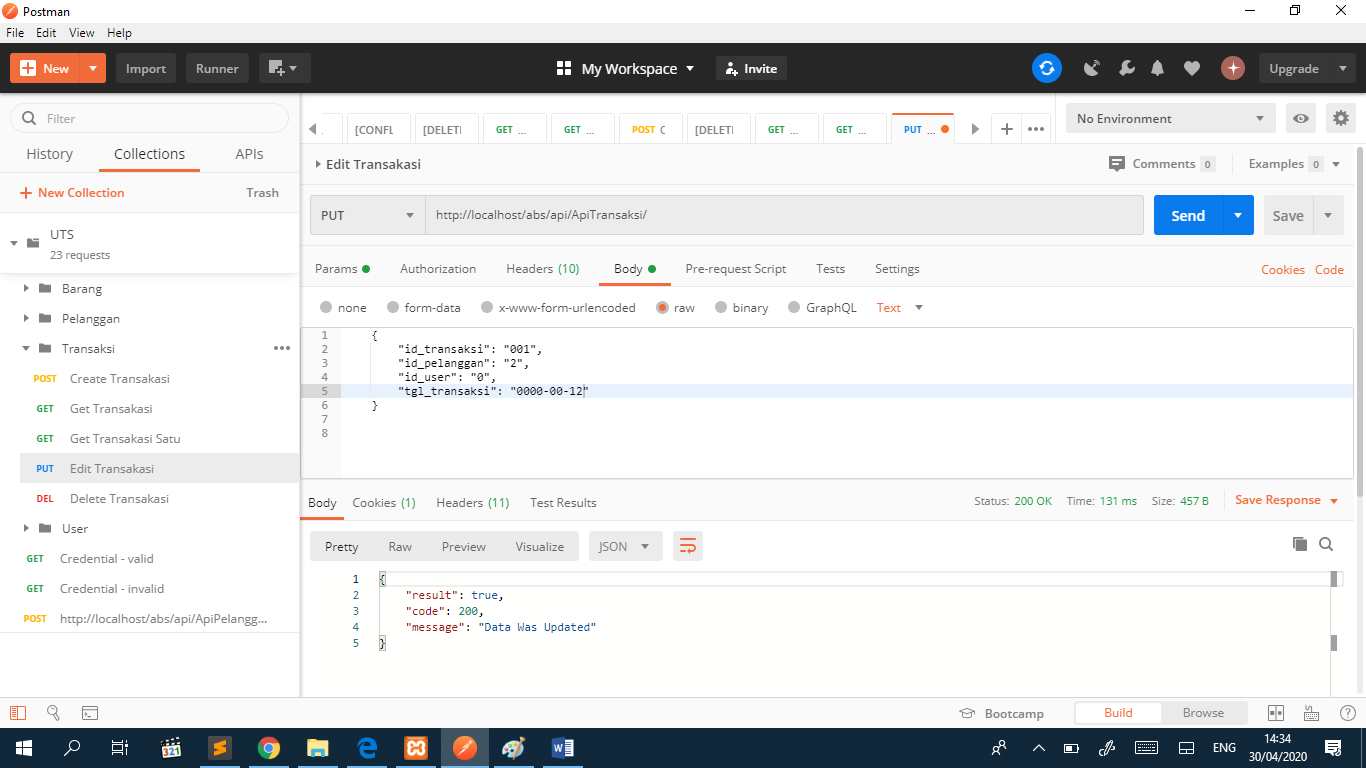
Transaksi

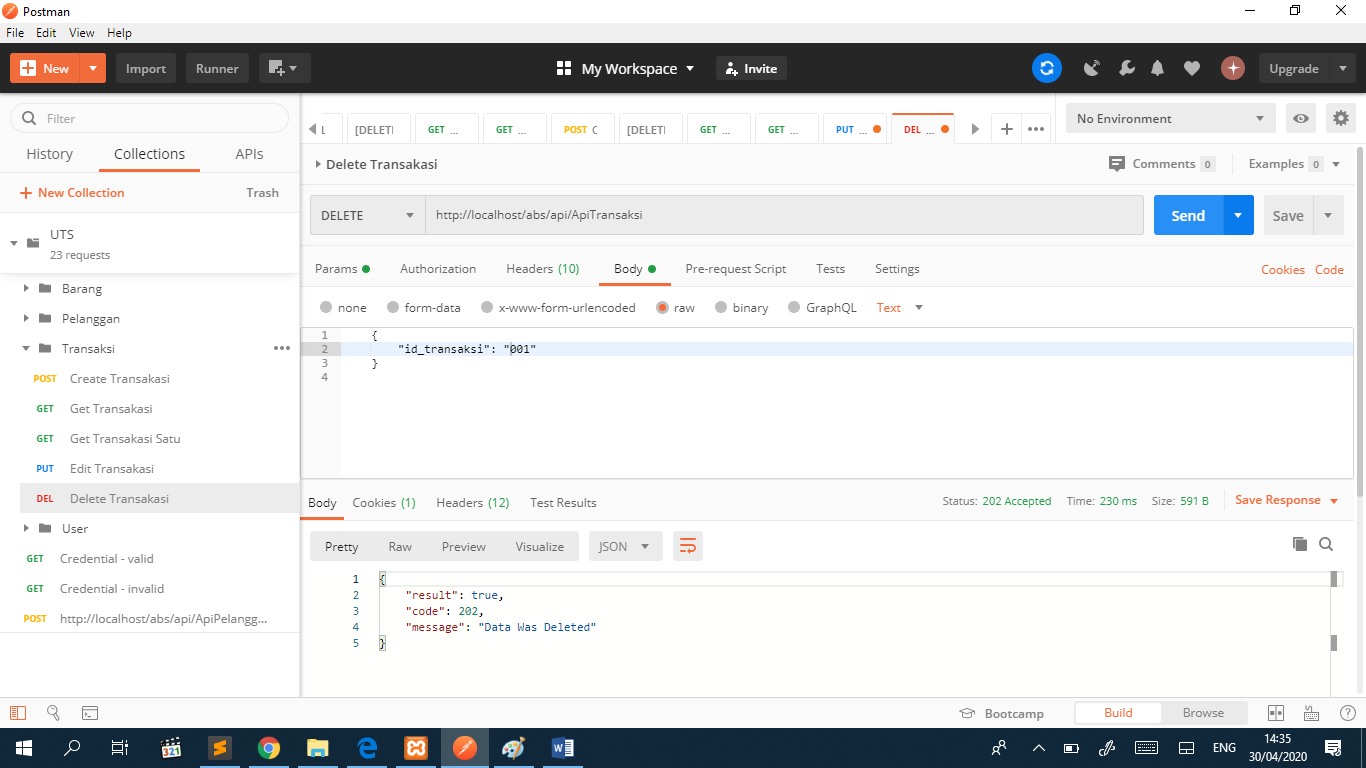


Id foreign key harus sama dengan yang di db jika tidak maka muncul pesan error.

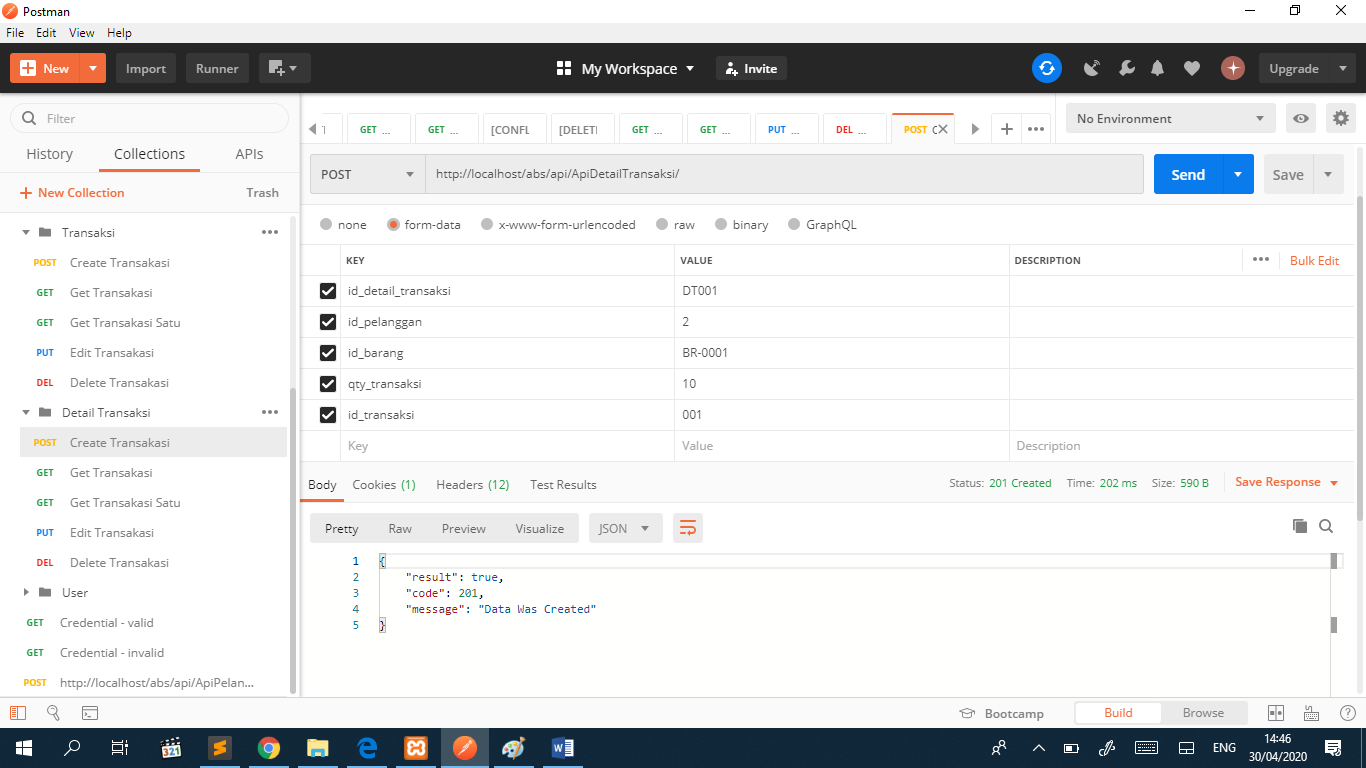


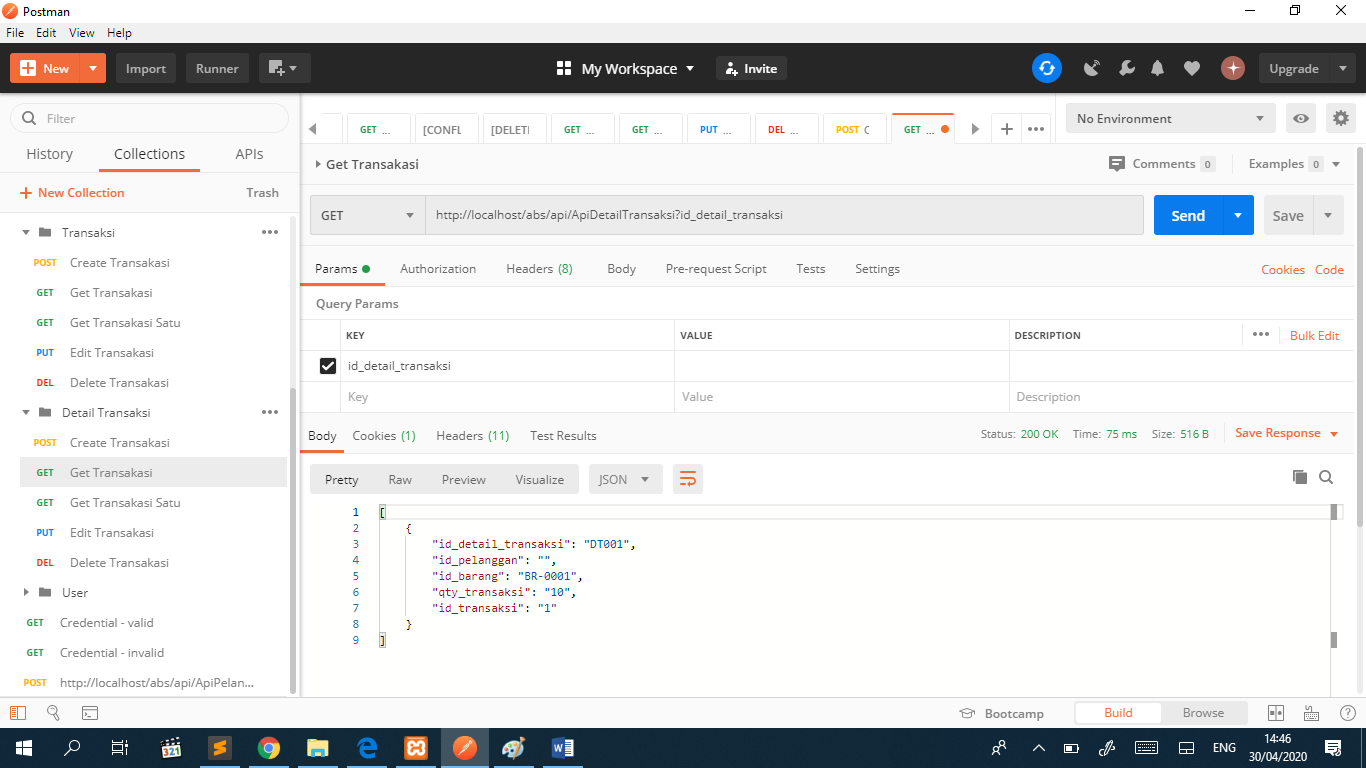


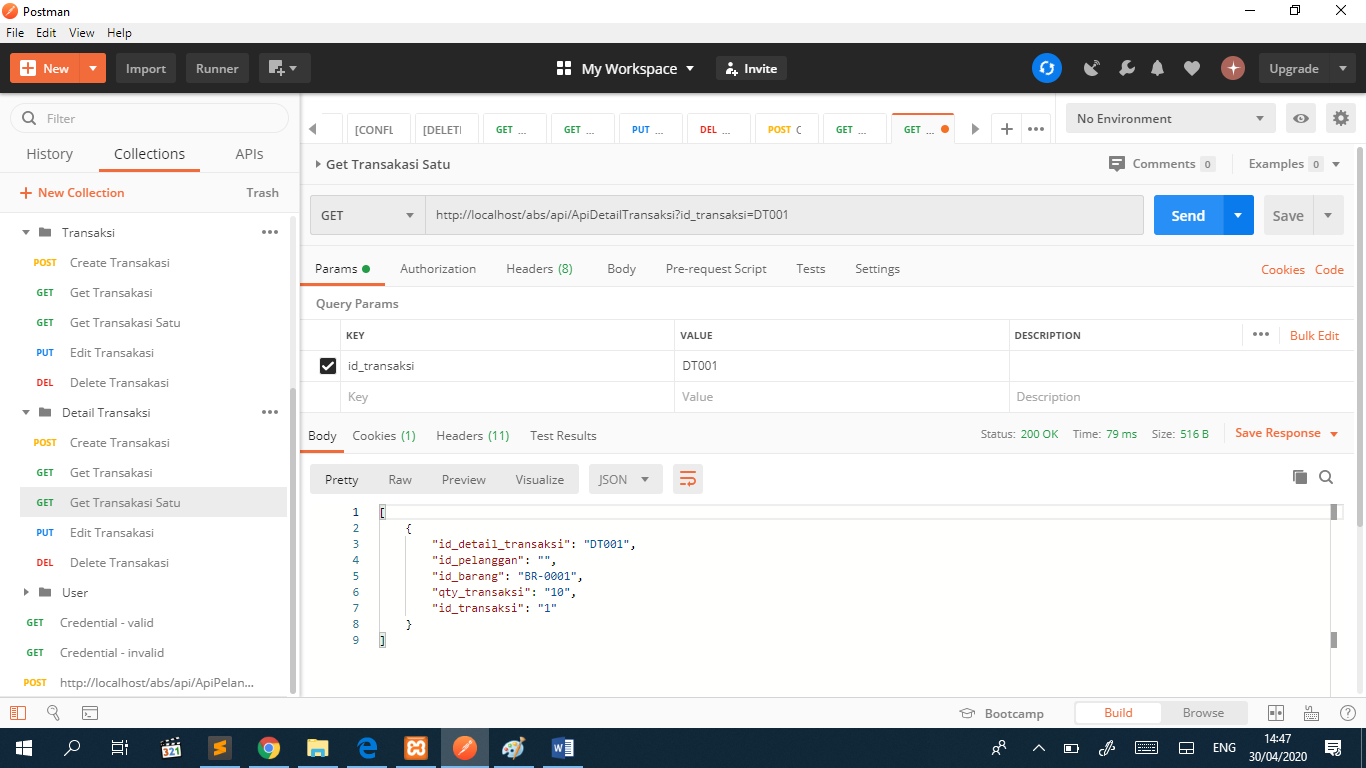




Detail Transaksi

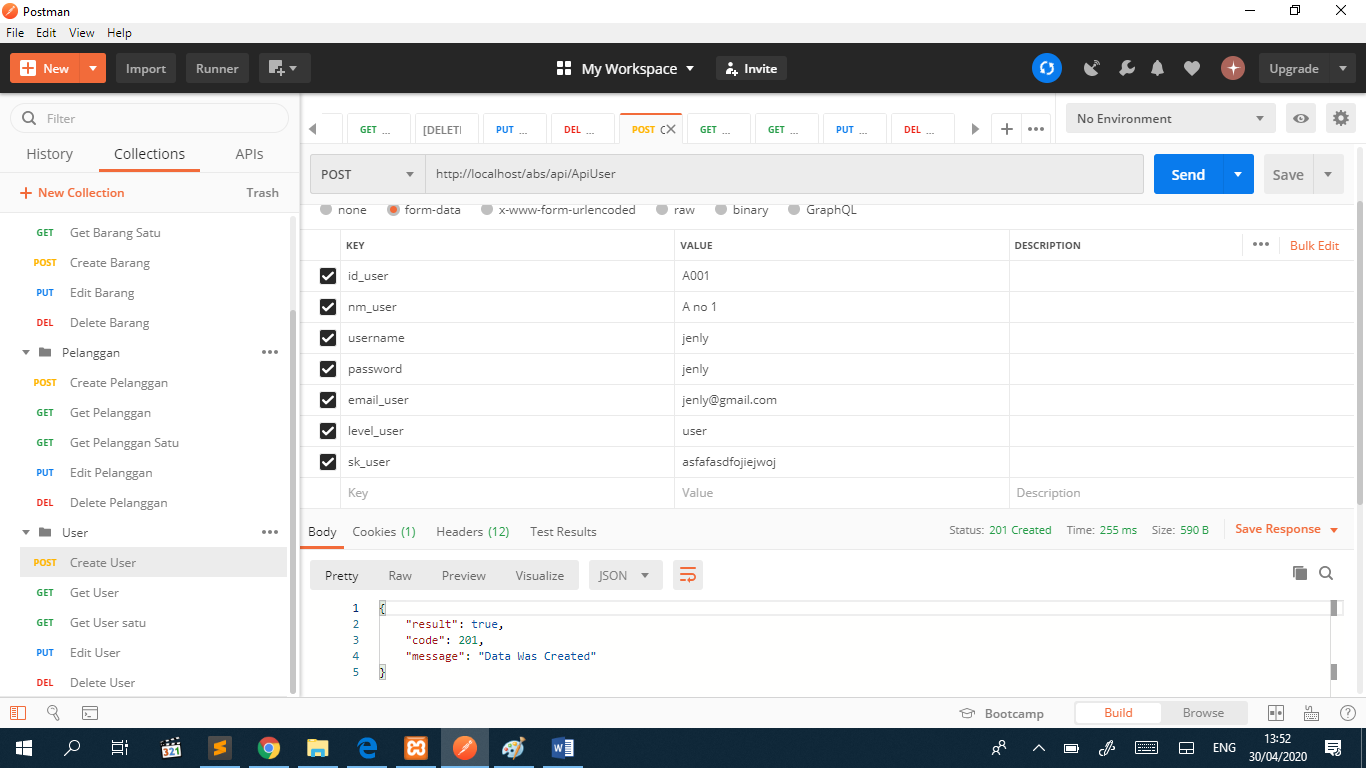


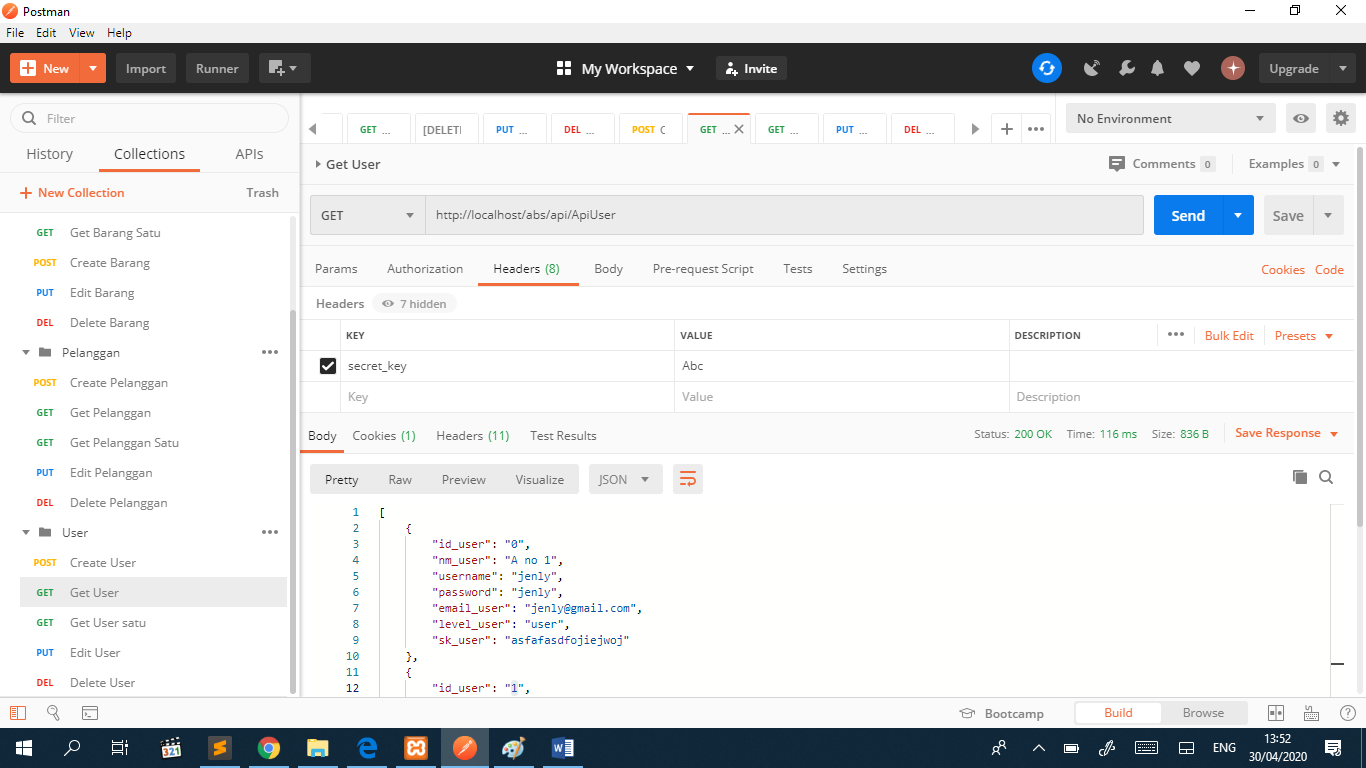


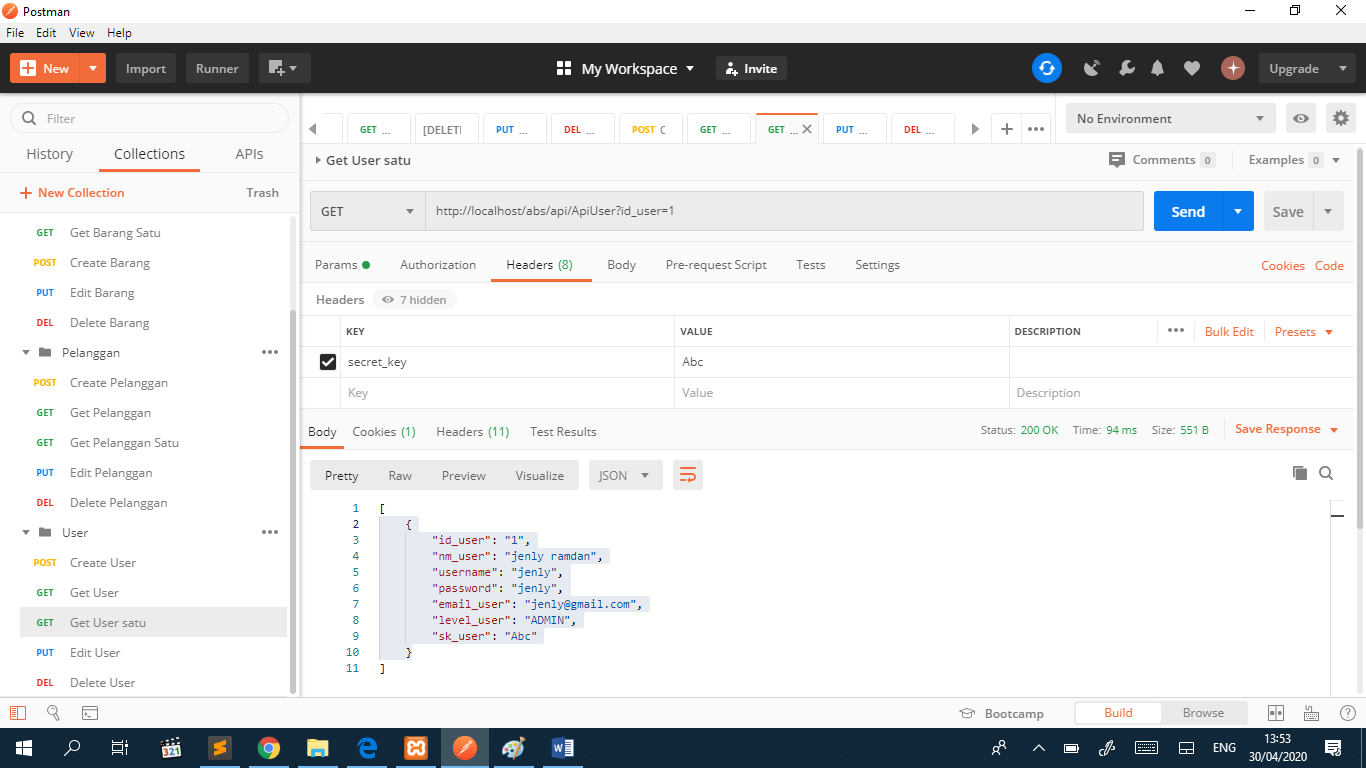


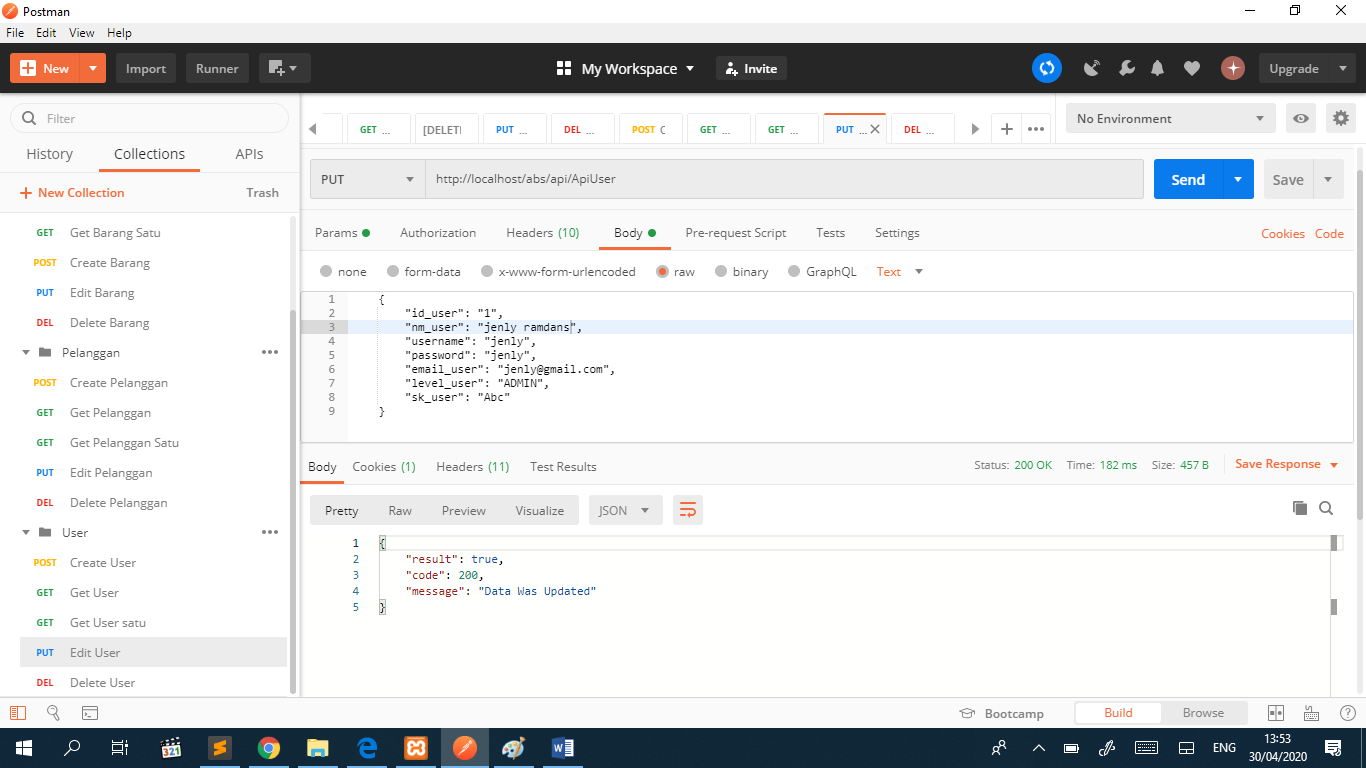
Class Model\_transaksi

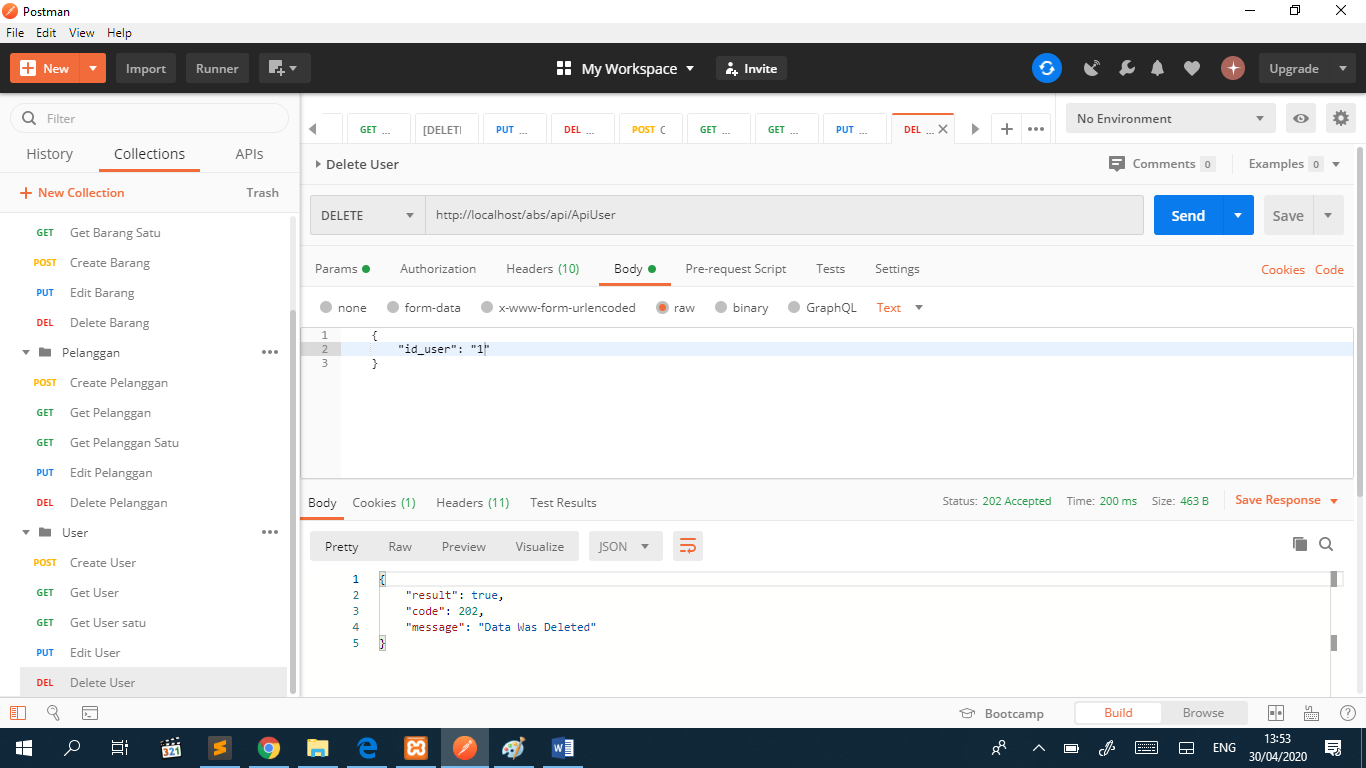
|  |  |
| --- | --- |
| public function show\_transaksi()  {  $data=$this->db->get('tbl\_transaksi');  return $data;  }  public function show\_one\_transaksi($id)  {  $data=$this->db->get\_where('tbl\_transaksi',array('id\_transaksi'=>$id));  return $data;  }  public function add()  {  $data=[  'id\_transaksi' => $this->input->post('id\_transaksi'),  'id\_pelanggan' => $this->input->post('id\_pelanggan'),  'id\_user' => $this->input->post('id\_user'),  'tgl\_transaksi' => $this->input->post('tgl\_transaksi'),  ];  return $this->db->insert('tbl\_transaksi',$data);  }  public function edit($id)  {  $data=$this->db->get\_where('tbl\_transaksi',array('id\_transaksi'=>$id));  return $data;  }  public function update($id\_transaksi, $id\_pelanggan, $id\_user, $tgl\_transaksi)  {  $data=[  'id\_pelanggan' => $id\_pelanggan,  'id\_user' => $id\_user,  'tgl\_transaksi' => $tgl\_transaksi,  ];  $this->db->where('id\_transaksi',$id\_transaksi);  return $this->db->update('tbl\_transaksi',$data);  }  public function delete($id\_transaksi)  {  $id\_transaksi = $id\_transaksi;  $this->db->where('id\_transaksi',$id\_transaksi);  return $this->db->delete('tbl\_transaksi');  } | Digunanakan untuk memunculkan data yang ada di table transaksi.  Parameter : \* (semua)  http://localhost/abs/api/ApiBarang  Digunakan untuk memunculkan salah satu data pada table transaksi dengan memilih value dari id\_transaksi.  Parameter :  Digunakan untuk menambahkan data (insert) ke dalam table.  Parameter : |





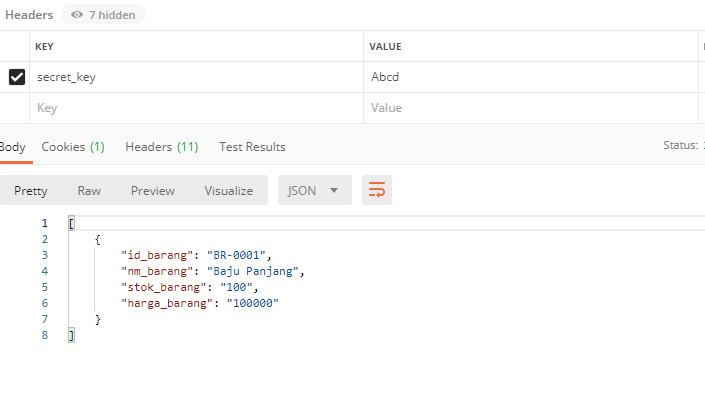


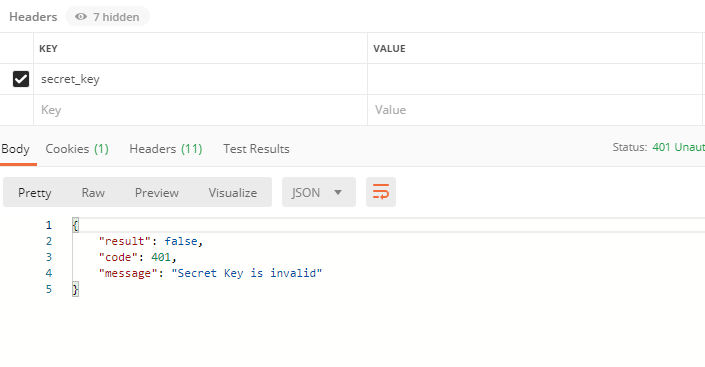




Credential

Secret\_key didapat ketika post ke user. Oleh karena itu jika secret key salah maka akan muncul pesan 401 yang berarti secret key nya salah atau tidak valid





|  |
| --- |
| <?php  $headers = apache\_request\_headers();  $secret\_key = '';  $authentication = false;  foreach ($headers as $header => $value) {  if ($header == 'secret\_key') {  $secret\_key = $value;  }  }  if ($secret\_key != '') {  $CI = get\_instance();  $CI->load->model('Model\_User');  $data = $CI->Model\_User->check\_secret\_key($secret\_key)->result();  (count($data) != 0) ? $authentication = true : $authentication = false;  }  if (!$authentication) {  http\_response\_code(401);  $arrResult = array(  'result' => false,  'code' => 401,  'message' => 'Secret Key is invalid'  );  echo json\_encode($arrResult);  exit();  } |

Jika auth salah maka akan code 401 yang berarti false atau secret key is invalid

Mendefinisikan bahwa var class berasal dari class Ci

Secret key tidak boleh kosong