

BAB III

PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan koordinasi

Dalam melakukan kerja praktek magang pada PT KPC, mahasiswa ditempatkan untuk melakukan tugas pekerjaan pada posisi bagian *IT Development*. Mahasiswa dibimbing langsung oleh ketua pembimbing pada bagian kerja praktek magang yaitu Bapak Henry Pratama serta dibantu oleh Bapak Widya Saputra selaku bagian staf IT. Selain itu, mahasiswa juga melakukan koordinasi dengan bagian HRD yaitu Bapak Muhammad Arif Rivai yang juga memiliki peran sebagai PIC *Internship* PT KPC. Seluruh proses kerja praktek magang dilakukan sesuai dengan instruksi dan arahan yang telah diberikan oleh pembimbing dan pelaksana kegiatan magang.

3.2 Tugas yang Dilakukan & Tools yang digunakan

Dalam pelaksanaan kegiatan kerja praktek magang, mahasiswa memiliki peran sebagai *Engineer* IT pada divisi *IT Development* yang bertujuan untuk membantu dalam merancang suatu aplikasi berbasis *website* dalam pembuatan *e-Form Statement of Equipment Defact* yang akan membantu *Admin* IT dalam proses penerbitan SoD (*Statement of Defact*) secara *online*. *Gadget* / Peralatan IT yang telah rusak akan dibuatkan SoD-nya dan SoD tersebut akan terbit setelah mendapatkan *approval* dari *Supt/Supv Computer Operations* yang selanjutnya terkumpul ke *Admin* IT.

Perancangan dan pembuatan aplikasi ini dilakukan dengan menggunakan beberapa *tools* yang dapat menjadi sarana pendukung dalam melakukan proses pengerjaan. *Tools* yang digunakan adalah sebagai berikut :

- SQL System Management Server (*Database*)
- Visual Studio 2019
- Google Chrome
- GitHub
- Trello (*Scheduling process System*)

3.3 Uraian Pelaksanaan Kerja Magang

Berikut merupakan *timeline* selama melakukan kegiatan kerja magang yang berisi keterangan terkait kegiatan kerja praktek magang mahasiswa.

Tabel 3.1 *Timeline* Kerja Praktek Magang

No.	Minggu ke-	Waktu Pelaksanaan	<i>Job description</i>
1.	1	21/06/2021 – 25/06/2021	<i>Interview</i> dan pengenalan kepada mahasiswa.
2.	2	28/06/2021 - 02/07/2021	<i>Instalasi software</i> dan menyesuaikan spesifikasi pada

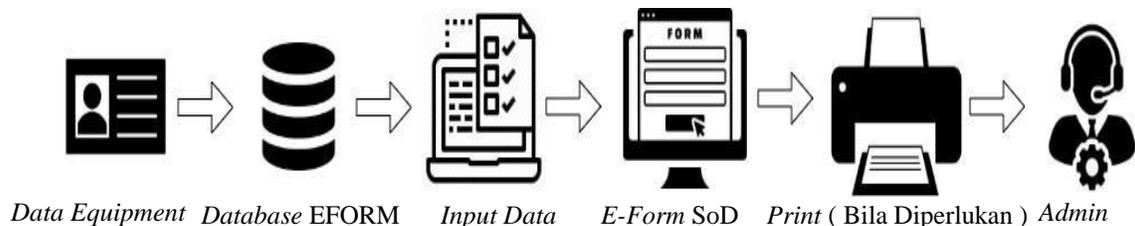
			ketentuan yang diberikan oleh pembimbing.
3.	3	05/07/2021 – 09/07/2021	Melakukan pembuatan <i>database</i> dan <i>Penginput data</i> .
4.	4	12/07/2021 – 16/07/2021	Pengembangan pada rancangan <i>database</i> .
5.	5	19/07/2021 – 23/07/2021 (20/07/2021 Libur)	Uji <i>database</i> dan pengabungan <i>table</i> .
6.	6	26/07/2021 – 30/07/2021	Persiapan konfigurasi dan <i>instalasi tools</i> Visual Studio.
7.	7	02/08/2021 – 13/08/2021 (10/08/2021 – 11/08/2021 Libur)	Membuat <i>form</i> dan mengunduh <i>file script code</i> .
8.	8	16/08/2021 – 31/08/2021 (17/08/2021 Libur)	Pembuatan sistem.

9.	9	01/09/2021 – 30/09/2021	Pengujian Website.
----	---	-------------------------	--------------------

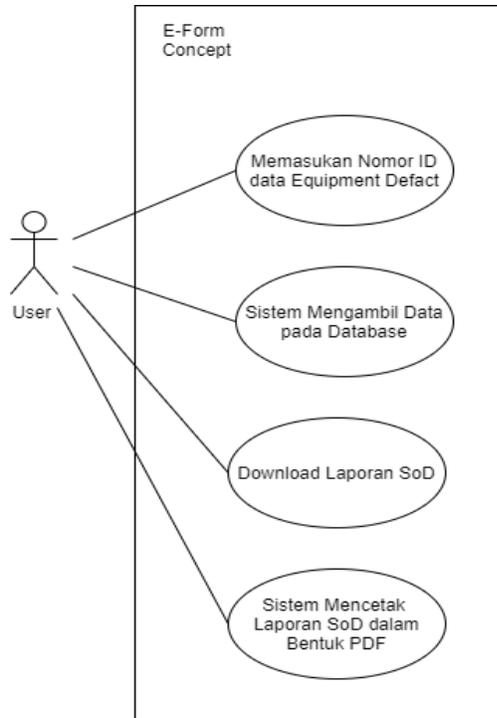
3.3.1 *Interview* dan Pengenalan Kepada Mahasiswa (Minggu 1)

Pada minggu pertama, kegiatan yang dilakukan oleh mahasiswa adalah melakukan *interview* pada pihak penyelenggara kegiatan praktek kerja magang dan melakukan interaksi dengan pembimbing dalam bentuk aktifitas tanya-jawab mengenai proses kerja apa seperti apa yang akan dihadapi oleh mahasiswa, posisi dan divisi apa yang akan menjadi tempat mahasiswa untuk melakukan kerja magang, aplikasi apa yang akan digunakan dan saran-saran pembimbing serta arahan sebelum memulai kegiatan kerja magang.

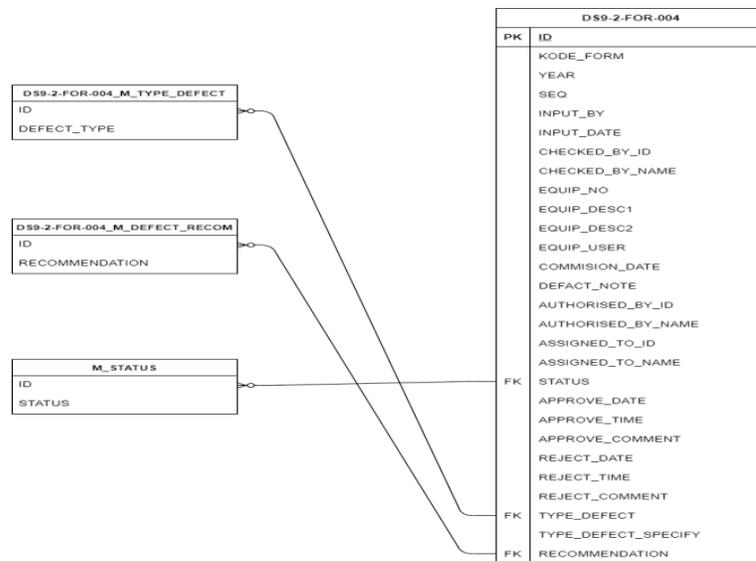
Adapun proses dari *interview* dan pengenalan ini dilakukan secara virtual atau daring (*online*). Aplikasi ataupun *software* yang digunakan dalam melakukan proses *interview* dan pengenalan adalah Zoom. Berikut merupakan alur dari proses *project* yang akan dikerjakan oleh mahasiswa selama dalam kegiatan kerja magang.



Gambar 3.1. Alur Proses *Project*



Gambar 3.2 Class Diagram E-Form Concept



Gambar 3.3 Struktur Database EFORM

Pada Gambar 3.1 merupakan alur proses dari rencana *project* yang akan dikerjakan oleh mahasiswa dalam kegiatan praktek kerja magang. Tahap pertama yang dilakukan adalah data kerusakan peralatan IT yang telah terkumpul akan dimasukkan ke dalam *database* yang akan dibuat untuk menampung setiap *informasi* data yang telah diberikan oleh petugas IT yang mencatat setiap kerusakan. Kemudian pada *database* disesuaikan kembali setiap data yang ada dengan menggabungkan setiap klasifikasi pada data dengan metode *join table* dan menentukan parameter pada data.

Pada tahap selanjutnya data kemudian diolah pada sistem dengan memasukan nomor ID dari data yang menjadi acuan dari parameter data. Lalu data akan secara otomatis tertulis pada *form* yang terdapat pada aplikasi dan kemudian akan secara otomatis *form* tersebut akan menjadi *format* pdf. Setelah data tersebut telah *diinput* pada *e-form* dan menjadi surat elektronik, selanjutnya *file form* tersebut dapat dicetak atau dapat langsung diberikan kepada *Admin* sebagai bentuk *informasi* dalam mengevaluasi setiap *equipment* yang digunakan oleh *staff-staff* IT yang bertugas.

Pada Gambar 3.2 merupakan gambaran konsep dari *e-form* yang dimana tahapan-tahapan konsep tersebut adalah *user* pertama kali memasukan nomor ID dari data yang terdapat pada *database* (1,2,3, dst). Kemudian sistem akan secara otomatis mengambil data yang sesuai dengan yang diinginkan oleh *user*. Kemudian data akan *terdownload* dan masuk ke dalam laporan sesuai dengan *statement* yang terdapat pada laporan. Kemudian setelah selesai *download* maka laporan tersebut akan secara otomatis menjadi bentuk *pdf* dan laporan SoD telah siap dikirimkan kepada *admin* baik berupa *softcopy* laporan maupun di print.

Gambar 3.3 merupakan struktur *table* data yang terdapat pada *database*. *Table* DS9-2-FOR-004 adalah *table* utama dari setiap kumpulan *table* yang akan dimasukan datanya. Pada *table* ini ID merupakan *primary key* yang berfungsi

sebagai kunci untuk membangun relasi pada setiap *table*. Di dalam *table* lain terdapat ID yang akan menjadi bagian untuk menyatukan relasi ID pada *table* DS9-2-FOR-004 dengan *table* DS9-2-FOR-004_M_TYPE_DEFECT, DS9-2-FOR-004_M_DEFECT_RECOM, dan M_STATUS.

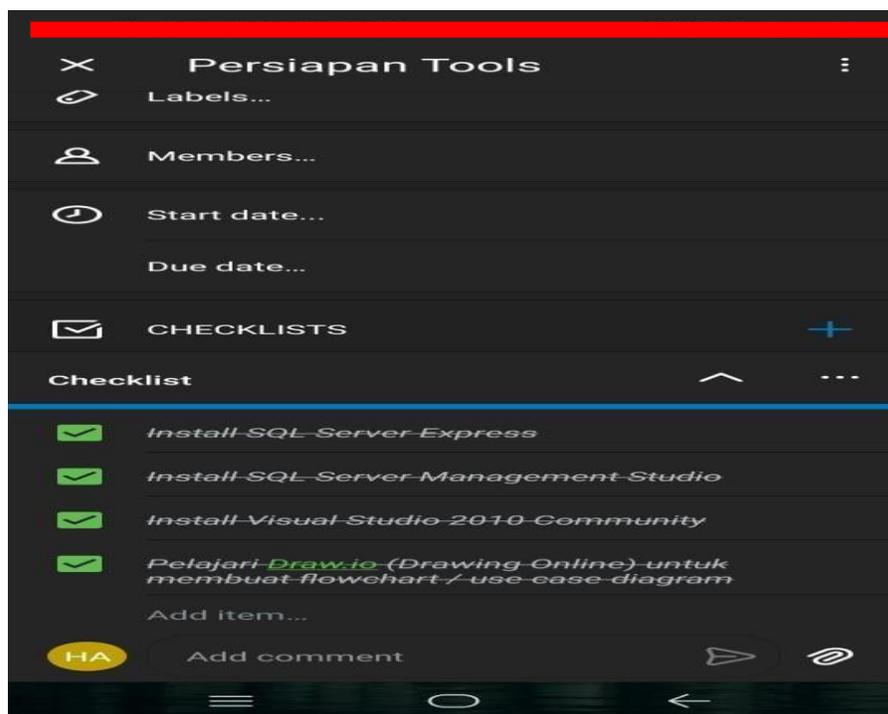
3.3.2 Instalasi Software dan Menyesuaikan Spesifikasi pada Ketentuan Yang Diberikan Oleh Pembimbing (Minggu 2)

Sebelum melakukan proses kerja magang, mahasiswa terlebih dahulu diberikan instruksi untuk melakukan *instalasi software* yang akan digunakan selama proses pengerjaan *project* magang. Arahan pertama yang diberikan oleh pembimbing magang adalah *menginstal* aplikasi Trello. Pada aplikasi Trello seluruh penjadwalan dari kerja magang dan *progress* tugas akan diberikan serta sebagai wadah untuk melakukan *sharing folder file* apabila ada yang harus diperbaiki kembali dan perlu dievaluasi kembali oleh pembimbing magang sebagai bentuk pengamatan untuk *progress project*. Pada aplikasi ini dapat juga sebagai catatan untuk kontribusi yang dilakukan oleh mahasiswa.



Gambar 3.4 Aplikasi Trello

Adapun kegiatan pertama yang dilakukan mahasiswa dalam mengerjakan kegiatan kerja magang adalah melakukan *penginstalan software* yang akan digunakan. Ketentuan tersebut terdapat di dalam aplikasi Trello. Berikut merupakan gambar mengenai daftar *software* yang harus dipersiapkan.



Gambar 3.5 Daftar *Software* yang Harus Dipersiapkan

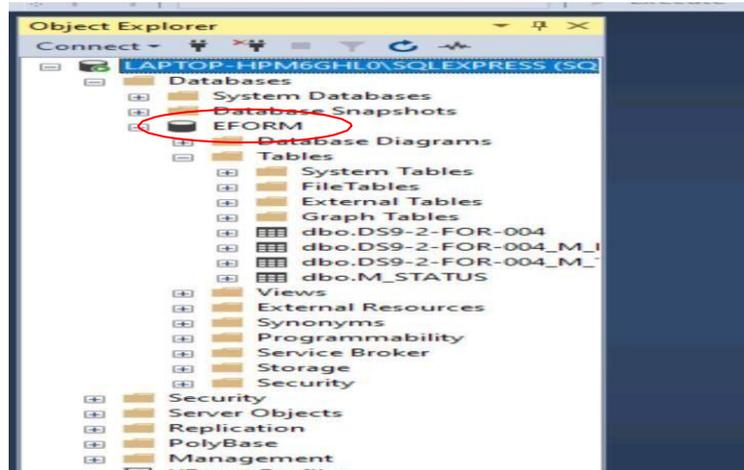
Pada Gambar 3.5 merupakan *list* dari *software* yang akan digunakan dalam melakukan proses pengerjaan *project* yang akan diberikan oleh pembimbing. *Software* yang akan digunakan adalah SQL Server Express sebagai tempat untuk mengolah data-data lengkap mengenai kerusakan *equipment* pada

kantor divisi IT dari perusahaan PT.KPC. kemudian melakukan *penginstalan* SSMS (SQL Server Management Studio) sebagai tempat untuk melakukan pembuatan *database* yang menjadi penyimpanan data-data yang akan *diinput*. Selanjutnya *menginstal* Visual Studio 2010 Community namun *software* tersebut *tidak* bisa menyesuaikan dengan sistem dan spesifikasi pada *device* sehingga menggunakan alternatif versi lain dari *software* tersebut yaitu Visual Studio 2019. Pada *software* ini akan dilakukan proses pembuatan *form* dan *script code* untuk membangun *website e-form*. Setelah semua *software* yang dibutuhkan telah siap, kemudian mahasiswa membuat *flowchart/use case* diagram sebagai panduan dalam mengetahui alur dari fungsi *project*. Selama proses penginstalan *software* mahasiswa mengalami beberapa kendala sehingga dalam seminggu mahasiswa dipandu agar *tools* yang dibutuhkan telah dinyatakan sesuai dan siap untuk digunakan pada proses pengerjaan *project*.

3.3.3 Melakukan Pembuatan *Database* dan *Penginputan* Data

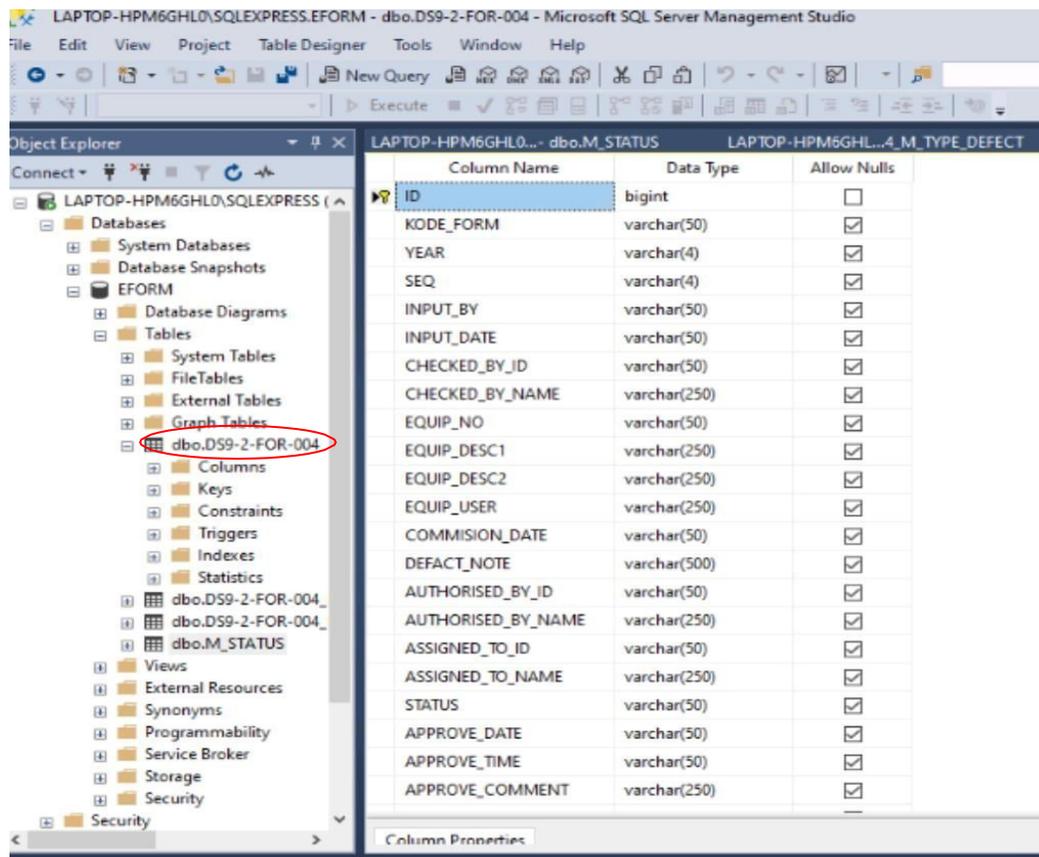
(Minggu 3)

Pada minggu ini mahasiswa diberikan tugas untuk melakukan pembuatan *database* pada SSMS (*SQL Server Management Studio*) dan *menginput* data yang telah diberikan pada aplikasi Trello.



Gambar 3.6 *Create Database EFORM*

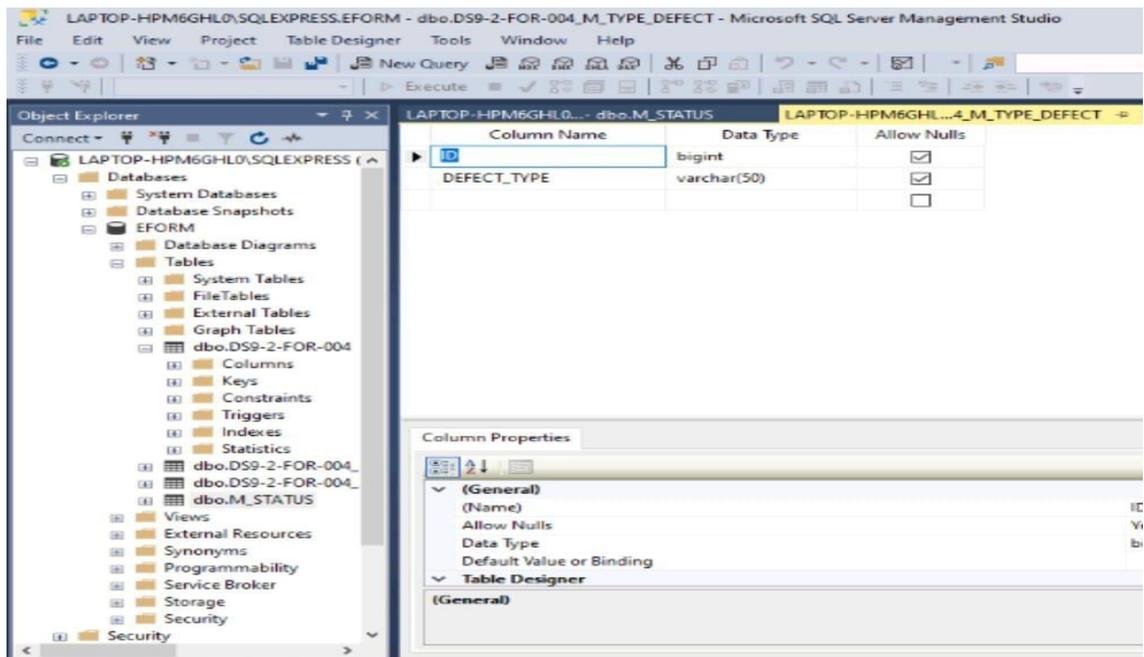
Tahap pertama dalam pembuatan *database* adalah dengan membuka *software* terlebih dahulu. Kemudian akan muncul tampilan SQL Server untuk mengkoneksikan *configstring* SQL Server dengan *device* penyimpanan data. Pada tampilan tersebut kemudian tekan tombol *connect* untuk terhubung ke dalam SSMS. Kemudian pada tampilan dalam SSMS terdapat *folder* yaitu *database* yang selanjutnya diklik kanan dan kemudian memilih opsi *New Database* untuk membuat *database* baru yang akan menjadi *database* EFORM. Selanjutnya pada *database* EFORM yang telah tersedia selanjutnya dibuat *table* yang sesuai dengan panduan yang telah diberikan oleh pembimbing.



Gambar 3.7 Table DS9-2-FOR-004

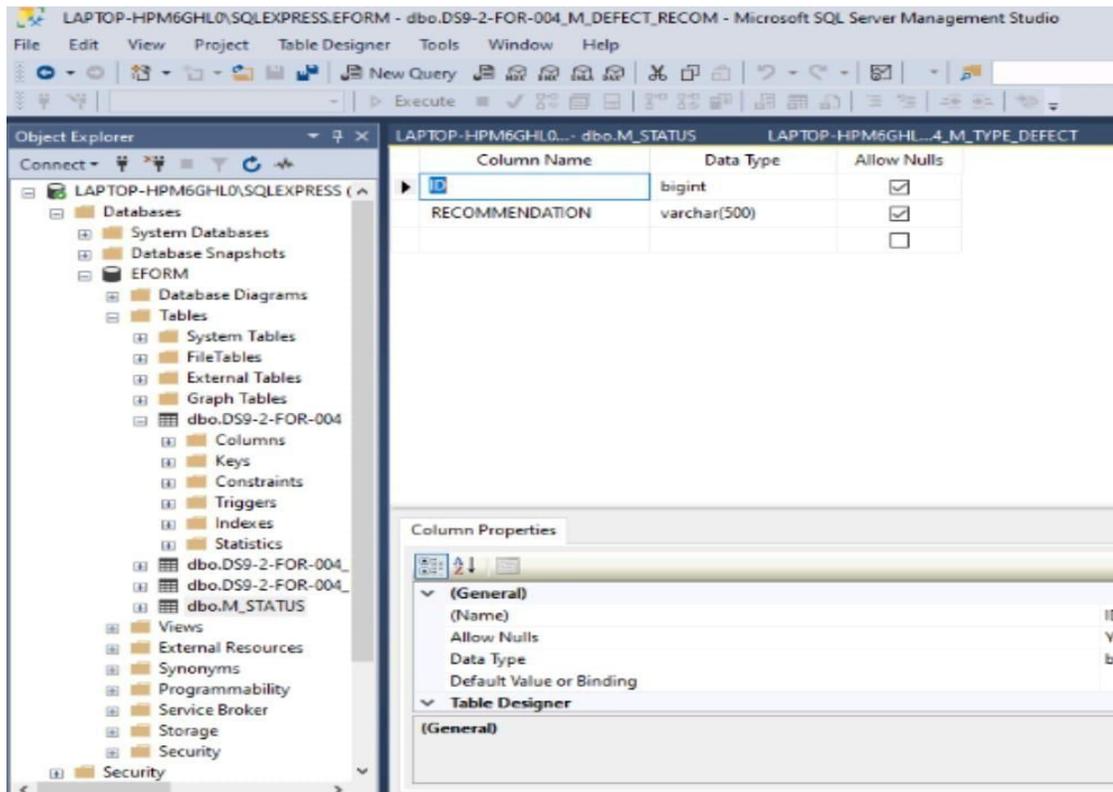
Pada Gambar 3.7 merupakan bentuk dari *table* DS9-2-FOR-004. *Table* tersebut berisi data-data yang terkait dengan kerusakan peralatan pada kantor IT PT.KPC. Didalam data terdapat ID sebagai *primary key* data, kemudian ada KODE_FORM sebagai inisiasi dari nomor kode *form* yang tercatat, YEAR untuk tahun pencatatan, SEQ untuk *sequences* atau urutan data, INPUT_BY dan INPUT_DATE merupakan nomor *input* data dan waktu penginputan data, CHECKED_BY_ID dan CHECKED_BY_NAME merupakan nomor *id* dan nama dari IT yang melakukan pencatatan kerusakan, kemudian EQUIP_NO, EQUIP_DESC1, EQUIP_DESC2, dan EQUIP_USER merupakan data dari nomor

equipment, deskripsi kerusakan *equipment*, dan pengguna dari *equipment* tersebut, lalu *COMMISSION_DATE* merupakan tanggal komisi, *DEFACT_NOTE* sebagai catatan dari kerusakan, *AUTHORISED_BY_ID* dan *AUTHORISED_BY_NAME* merupakan *id* dan nama pihak yang menyetujui dari kesesuaian laporan kerusakan, *ASSIGNED_TO_ID* dan *ASSIGNED_TO_NAME* merupakan *id* dan nama yang ditugaskan untuk memperbaiki maupun mencari solusi perbaikan, *STATUS* merupakan status dan bentuk dari kerusakan peralatan, dan *APPROVE_DATE*, *APPROVE_TIME*, *APPROVE_COMMENT* merupakan tanggal persetujuan laporan, waktu dan *comment* dari pihak *admin* IT mengenai laporan tersebut.



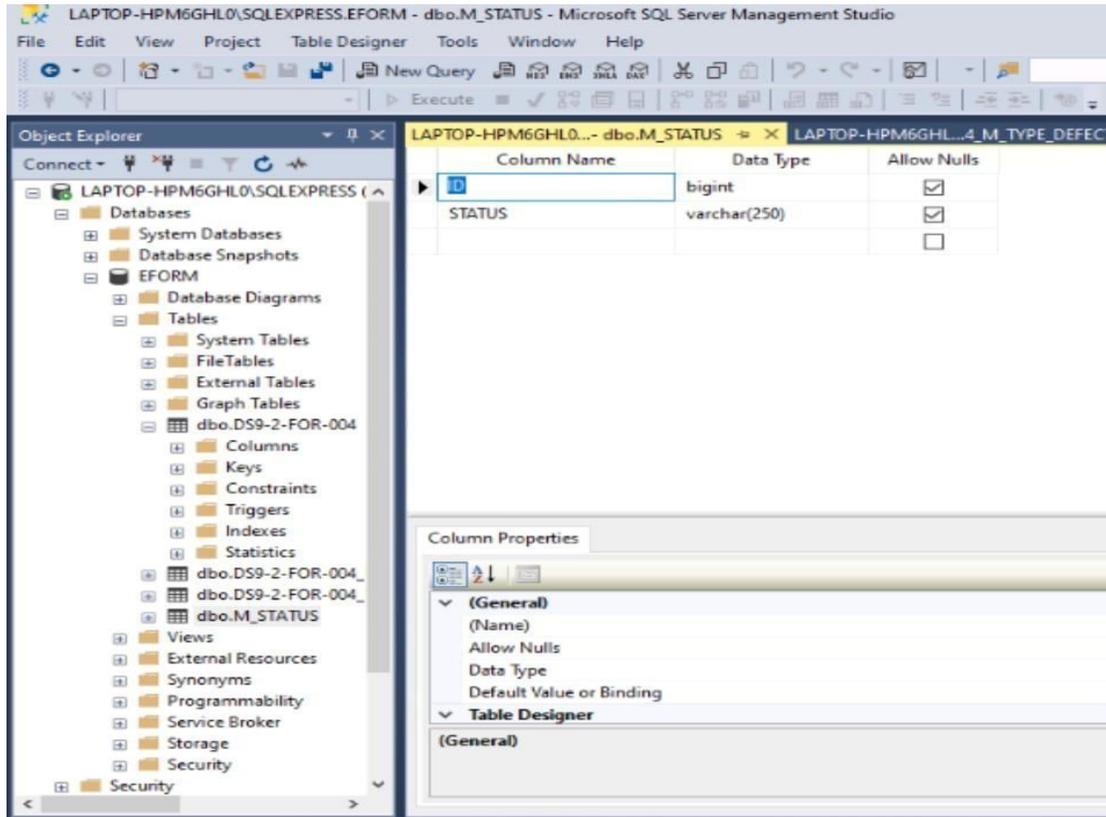
Gambar 3.8 Table DS9-2-FOR-004-M-TYPE-DEFECT

Pada gambar 3.8 merupakan *table* untuk mengetahui bentuk dari jenis kerusakan yang terjadi. Pada *table* terdapat 2 tipe kerusakan yaitu *normal use* dan *other than normal use*. Komponen yang terdapat di dalam *table* adalah ID dan DEFECT_TYPE.



Gambar 3.8 Table DS9-2-FOR-004_M_DEFECT_RECOMMENDATION

Pada gambar 3.9 merupakan *table* untuk rekomendasi dari cara mengatasi kerusakan pada peralatan. Terdapat 4 rekomendasi pada perbaikan yang terdapat dalam RECOMMENDATION dan ID dari rekomendasi tersebut.



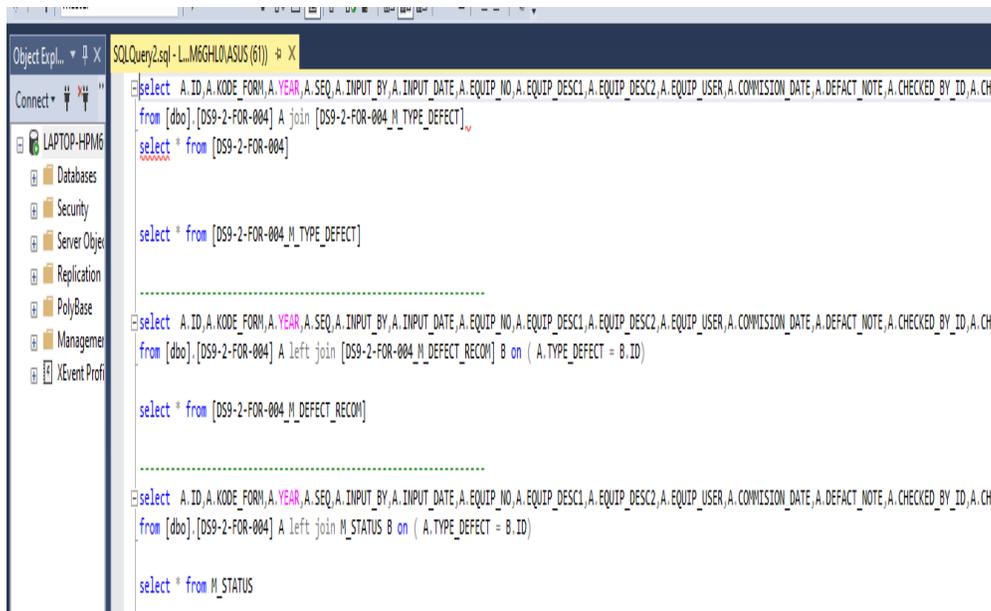
Gambar 3.10 Table M_STATUS

Pada gambar 3.10 merupakan *table* untuk status dari pengajuan laporan kerusakan peralatan. Pada STATUS terdapat 10 tipe status dari perbaikan yang akan dilakukan. Di dalam *table* ini memiliki 2 komponen yaitu ID yang merupakan tipe *id* perbaikan dan STATUS merupakan kondisi dan respon dari pengajuan laporan kerusakan peralatan.

3.3.4 Pengembangan pada Rancangan Database

(Minggu 4)

Pada minggu keempat, mahasiswa memiliki tugas untuk melakukan pengembangan kembali terhadap *database* yang telah dibuat pada minggu sebelumnya dengan melakukan *research* mengenai teknik yang akan digunakan dalam mengolah setiap data pada *table*. Dalam pengembangan ini sering terjadi *error* atau teknik yang digunakan tidak sesuai namun mahasiswa menemukan metode teknik yang mampu diterapkan pada *table database* yaitu *join table*. Konsep *join table* sendiri adalah melakukan suatu penggabungan data-data yang terdapat pada *table*. Selain itu percobaan dilakukan dengan memberikan setiap data-data label berupa huruf untuk memudahkan dalam penggabungan data. Berikut merupakan keterangan gambar dari pengembangan yang dilakukan pada *database*.

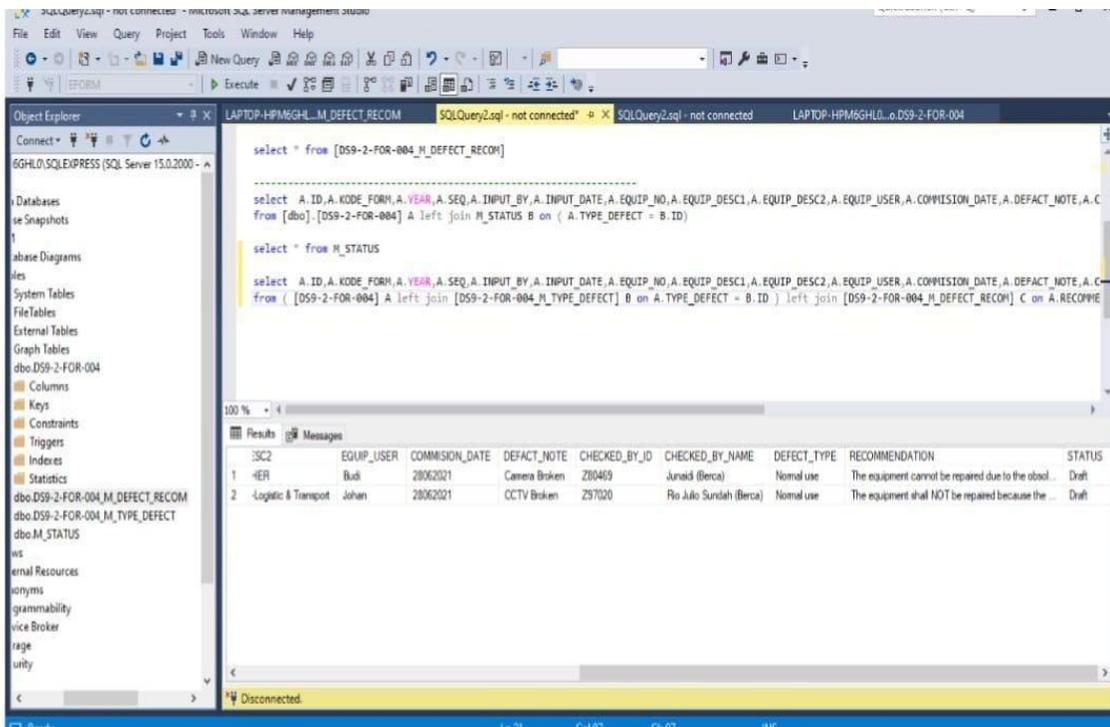


```
select A.ID,A.KODE_FORM,A.YEAR,A.SEQ,A.INPUT_BY,A.INPUT_DATE,A.EQUIP_NO,A.EQUIP_DESC1,A.EQUIP_DESC2,A.EQUIP_USER,A.COMMISSION_DATE,A.DEFACT_NOTE,A.CHECKED_BY_ID,A.CH  
from [dbo].[DS9-2-FOR-004] A join [DS9-2-FOR-004_M_TYPE_DEFECT]  
select * from [DS9-2-FOR-004]  
  
select * from [DS9-2-FOR-004_M_TYPE_DEFECT]  
  
-----  
select A.ID,A.KODE_FORM,A.YEAR,A.SEQ,A.INPUT_BY,A.INPUT_DATE,A.EQUIP_NO,A.EQUIP_DESC1,A.EQUIP_DESC2,A.EQUIP_USER,A.COMMISSION_DATE,A.DEFACT_NOTE,A.CHECKED_BY_ID,A.CH  
from [dbo].[DS9-2-FOR-004] A left join [DS9-2-FOR-004_M_DEFECT_RECUM] B on ( A.TYPE_DEFECT = B.ID)  
  
select * from [DS9-2-FOR-004_M_DEFECT_RECUM]  
  
-----  
select A.ID,A.KODE_FORM,A.YEAR,A.SEQ,A.INPUT_BY,A.INPUT_DATE,A.EQUIP_NO,A.EQUIP_DESC1,A.EQUIP_DESC2,A.EQUIP_USER,A.COMMISSION_DATE,A.DEFACT_NOTE,A.CHECKED_BY_ID,A.CH  
from [dbo].[DS9-2-FOR-004] A left join M_STATUS B on ( A.TYPE_DEFECT = B.ID)  
  
select * from M_STATUS
```

Gambar 3.11 Bentuk Pengolahan dan Pengembangan Data Table Database

3.3.5 Uji Database dan Pengabungan Table (Minggu 5)

Pada minggu kelima mahasiswa melakukan pengujian kembali terhadap database dengan menyesuaikan kembali koneksi yang ada pada database agar dapat dihubungkan ke dalam Visual Studio. Berikut merupakan proses testing database dan pengabungan pada table yang berisi data-data kerusakan pada peralatan.

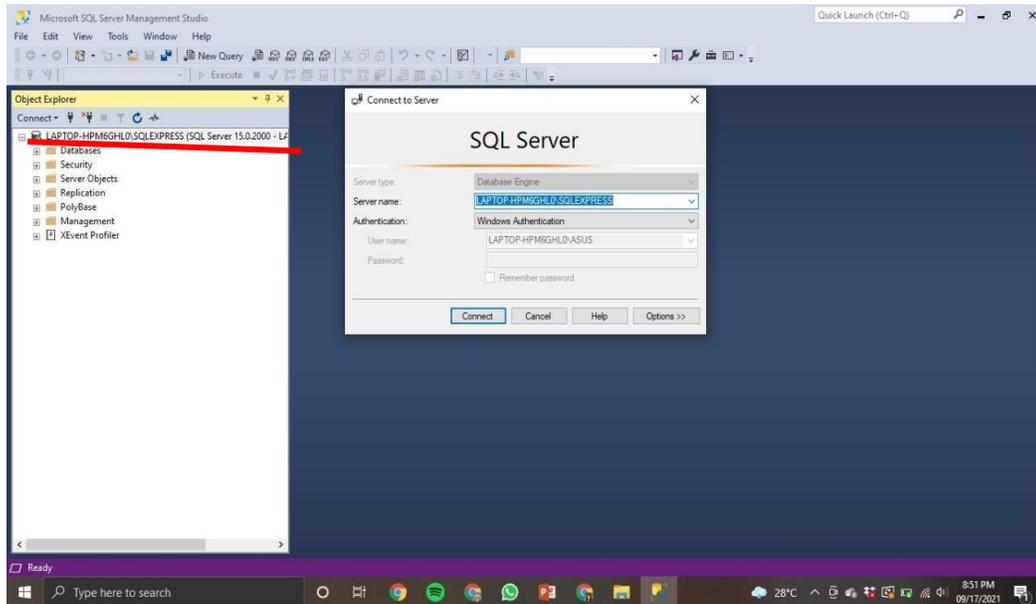


Gambar 3.12 Pengabungan Table

Pada gambar 3.12 merupakan proses dari pengabungan table. Table-table pada bentuk awal sebelum dilakukannya pemrograman masih terbagi menjadi 4 bagian table yaitu table utama DS9-2-FOR-004, kemudian table DS9-2-FOR-004_M_TYPE_DEFAC_T yang merupakan table untuk tipe dari bentuk kerusakan peralatan, lalu selanjutnya ada table DS9-2-FOR-

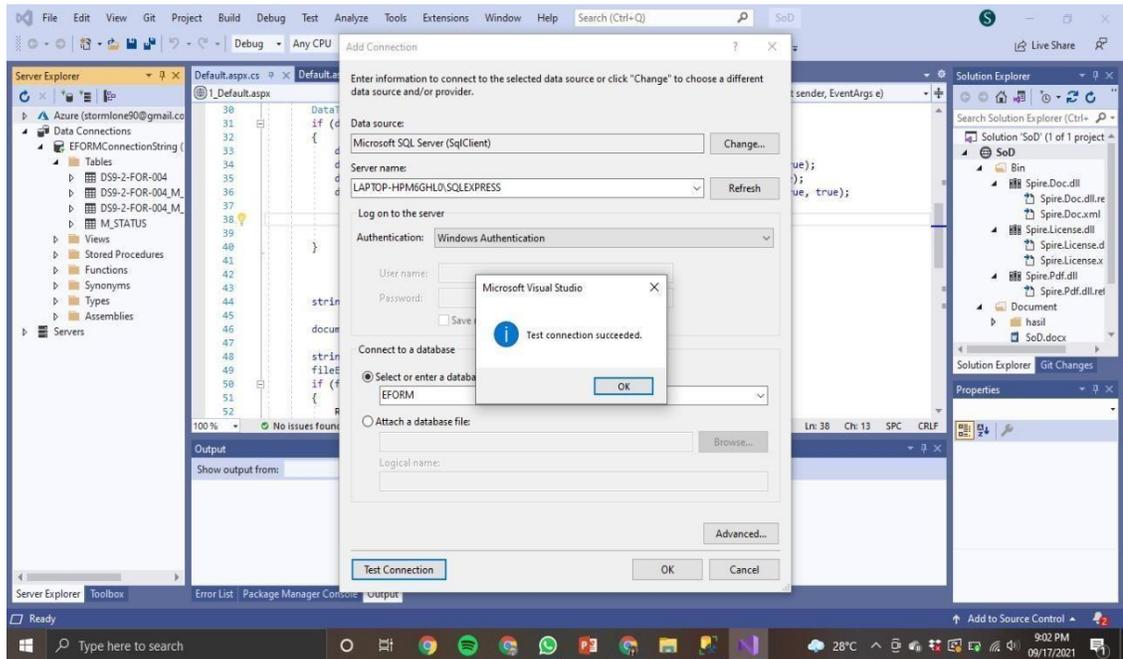
004_M_DEFECT_RECOMMENDATION yang merupakan *table* untuk rekomendasi dari perlakuan apa yang harus dilakukan terhadap peralatan yang rusak, dan yang terakhir adalah *table* M_STATUS yang merupakan *table* untuk mengetahui status dari pengajuan laporan. Metode pengabungan *table* dilakukan dengan cara *JoinTable*. *JoinTable* berfungsi untuk mengabungan keempat *table* menjadi satu pada *table* DS9-2-FOR-004 sebagai *table* utama dengan *primary key* berada pada ID.

Table yang diambil pada 3 kolom lainnya adalah DEFECT_TYPE, RECOMMENDATION, dan STATUS. Data-data pada kolom dibuat menjadi 4 variabel. Untuk variabel pada data kolom DS9-2-FOR-004 semuanya diinisialkan menjadi A, lalu pada data kolom DS9-2-FOR-004_M_TYPE_DEFACT semuanya diinisialkan menjadi B, pada data *table* DS9-2-FOR-004_M_DEFECT_RECOMMENDATION setiap data diinisialkan menjadi C, dan yang terakhir adalah M_STATUS yang diinisialkan menjadi D. Variabel inisial B, C, dan D dimasukkan kedalam data pada *table* inisial A dengan metode *join table from table-table* asal dari inisial-inisial data. Apabila setiap data telah masuk maka pada *table* utama akan muncul kolom baru dari hasil *join table* 3 inisial *table* yang telah dibuat.



Gambar 3.13 *Connection to Server Device*

Pada gambar 3.13 merupakan bentuk dari *connection* program kepada server yang terdapat pada *device*. Server name sebagai alamat *database* yang akan menjadi koneksi utama dari Visual Studio menuju *database* utama.



Gambar 3.14 Koneksi Database dan Visual Studio Berhasil

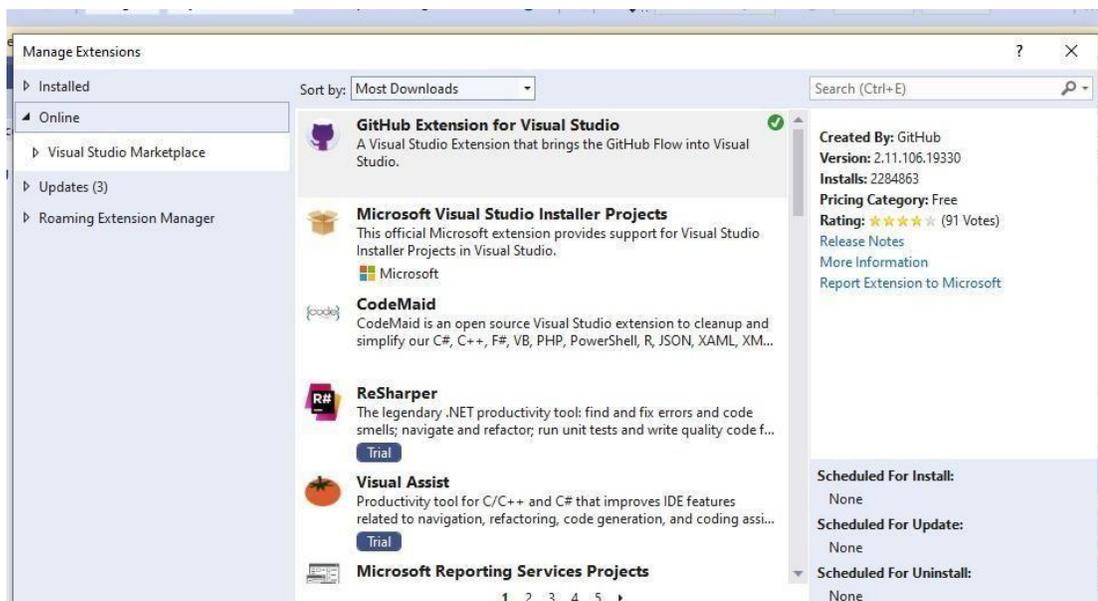
Pada gambar 3.14 adalah hasil dari koneksi yang telah dilakukan. Alamat server name yang telah dicopy pada SSMS kemudian dimasukkan ke dalam server name pada fitur *connect to database* pada Visual Studio. Lalu pada *add connection* alamat yang telah dimasukkan selanjutnya diuji dengan mencoba mencari nama *database* yang terdapat pada *server name*. Apabila nama *database* tidak ada yang muncul maka proses penyambungan gagal, namun apabila daftar *table* maupun *database* yang ada pada server muncul itu menandakan koneksi berhasil 80%. Pilih *database* EFORM sebab *database* tersebut merupakan server utama dari *table-table* yang akan diolah dan dibentuk ke dalam *form* digital di Visual Studio. *Test connection* dan apabila muncul notifikasi *Test Connection Succeeded* berarti menandakan bahwa *database* dan Visual Studio telah terhubung dan berhasil *connect*. *Database*

akan muncul pada *Server Explorer* dan selanjutnya data-data *table* akan disesuaikan dengan bentuk *e-form* yang akan dibuat.

3.3.6 Persiapan Konfigurasi dan Instalasi *Tools* Visual Studio

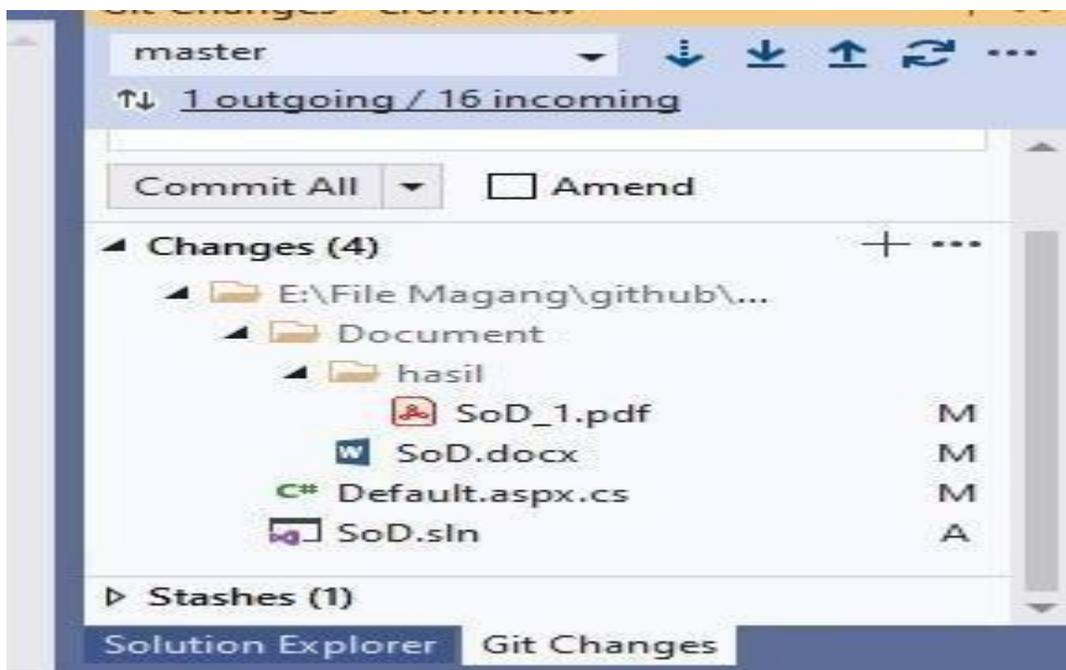
(Minggu 6)

Pada minggu keenam, mahasiswa memiliki tugas untuk melakukan konfigurasi dan *instalasi* dari komponen-komponen *tools* yang akan digunakan pada Visual Studio. Konfigurasi yang dilakukan adalah dengan menyambungkan jaringan pada *database* kedalam sistem dari Visual Studio. Pada Visual Studio kemudian dilakukan *instalasi tools* yang akan digunakan dalam pengambilan *file code* yang akan diberikan oleh pembimbing. *Tools* yang digunakan adalah Github Extension for Visual Studio.



Gambar 3.15 Github Extension for Visual Studio

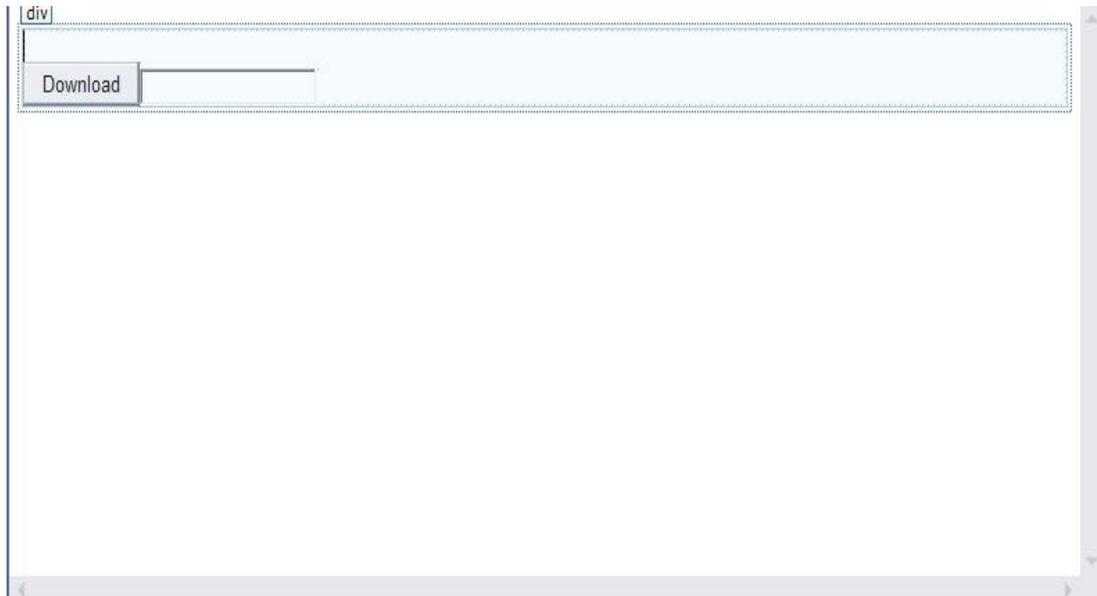
Pada gambar 3.15 merupakan *tools* yang digunakan dalam persiapan untuk mengerjakan *project* pada Visual Studio. *Tools* ini berfungsi sebagai penghubung antara akun pembimbing dengan akun mahasiswa dalam melakukan kegiatan *sharing file* yang akan dikerjakan oleh mahasiswa.



Gambar 3.16 Git Changes

Pada gambar 3.16 adalah hasil dari fungsi *tools* yang digunakan. Pada option *Git Changes* terdapat *folder-folder* yang dibutuhkan oleh mahasiswa dalam proses pengerjaan *project* yang diberikan oleh pembimbing. *Folder* yang dibagikan oleh pembimbing dilakukan melalui *website* github yang kemudian akan diunduh oleh mahasiswa. Untuk mendapatkan akses *folder*, mahasiswa

terlebih dahulu membuat *git repository* dan *add* pembimbing ke dalam *repository* tersebut agar pembimbing dapat melakukan *sharing file*.



Gambar 3.17 Komponen pada *Folder*

Pada gambar 3.17 merupakan salah satu komponen pada *folder* yang berhasil masuk ke dalam Visual Studio mahasiswa. Komponen ini bernama *Default.aspx* yang merupakan tampilan awal pada *project* yang berada di dalam *website*.

3.3.7 Membuat *Form* dan *Download File Code Script*

(Minggu 7)

Pada minggu ini mahasiswa melakukan aktifitas pembuatan *form* yang dilakukan pada *software* Ms.Word sebagai formulir yang akan diisi secara otomatis dengan data-data yang telah diolah pada *database*. Ketentuan *form* telah disesuaikan dengan keadaan *form* fisik sehingga hanya memerlukan pengisian berupa data yang ada. Lalu kegiatan selanjutnya adalah melakukan pengambilan *file code script* pada *website* github.

Statement of Equipment Defect

This is to advise the defect(s) of the following equipment and parts (as described in details):
[EQUIP_NO] [EQUIP_DESC1] [EQUIP_DESC2]
[EQUIP_USER]
[COMMISSION_DATE]

The statement is given after checking the condition of the equipment.

Checked by : [CHECKED_BY_NAME]
Badge No : [CHECKED_BY_ID]

The defect(s) involves the following parts of the equipment:

[DEFACT_NOTE]

Type of defect:

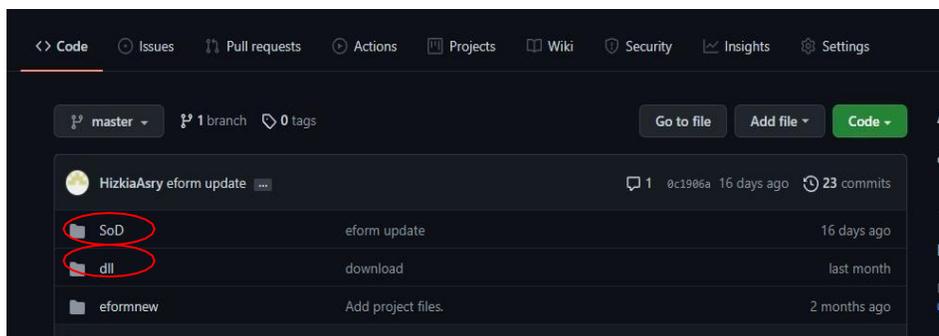
Normal use.

Other than normal use, (please specify) _____

Considering the defect(s) as mentioned above, our recommendation is as follows:

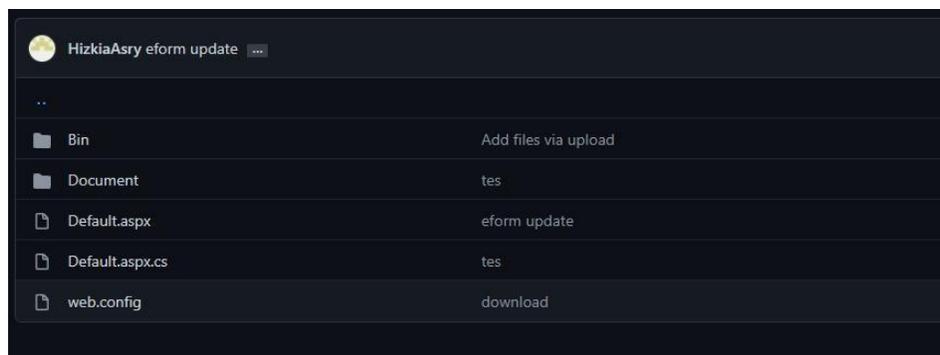
Gambar 3.18 *Form* dan *Input Label Data*

Pada gambar 3.18 merupakan *form* yang akan digunakan dan dijadikan tempat untuk memasukan data secara otomatis. Pada *form* tersebut telah diberikan label data yang akan dipanggil pada sistem sehingga data akan secara otomatis masuk ke dalam *form* secara otomatis.



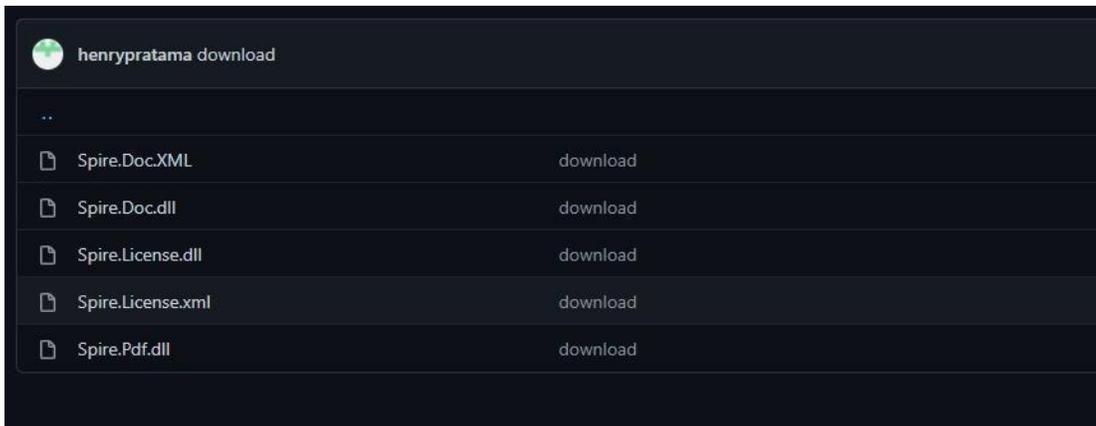
Gambar 3.19 Folder pada Github

Pada gambar 3.19 merupakan *folder* pada github yang telah diberikan oleh pembimbing untuk mahasiswa rancang kembali konsep *project* yang telah ditentukan pada *folder-folder* tersebut.



Gambar 3.20 Folder SoD

Pada gambar 3.20 merupakan isi dari *folder* SoD yang berisi atribut yang akan digunakan dalam proses perancangan sistem *e-form*. *Folder* tersebut berisi *Folder* Bin, *Folder* Document, Default.aspx, Default.aspx.cs, web.config.



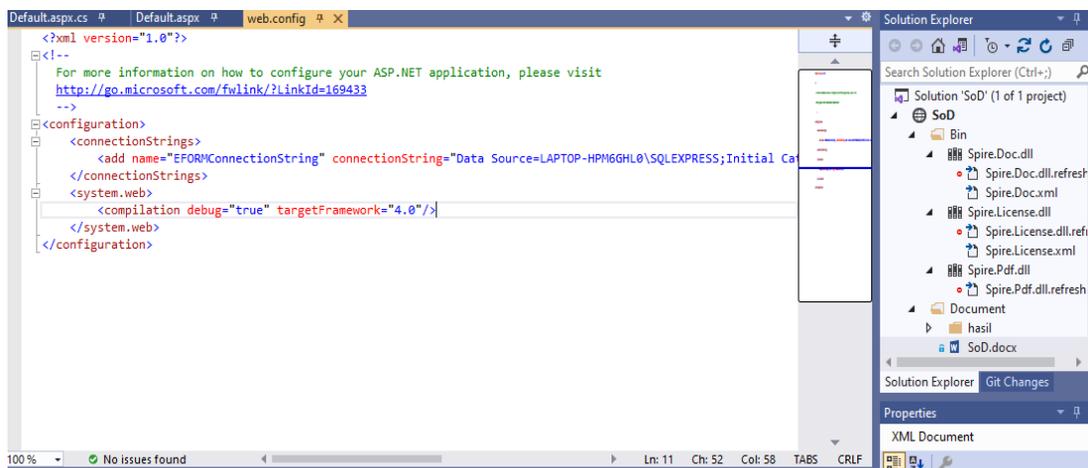
Gambar 3.21 *Folder* dll

Pada gambar 3.21 merupakan isi dari *folder* dll yang merupakan komponen yang akan digunakan pada Visual Studio sebagai atribut yang akan digunakan pada saat proses pengerjaan *project*. *Folder* tersebut berisi Spire.Doc.XML, Spire.Doc.dll, Spire.License.dll, Spire.License.xml, Spire.Pdf.dll.

3.3.8 Pembuatan Sistem

(Minggu 8)

Pada minggu kedelapan mahasiswa melakukan pembuatan sistem *e-form* dengan memulai *coding* dan memasukan data pada konsep rancangan *script code* yang telah ditentukan. Namun, sebelum dimulainya proses pengerjaan sistem *e-form*, mahasiswa diarahkan dan dipandu oleh pembimbing terlebih dahulu agar sesuai dengan ketentuan yang telah diberikan oleh pembimbing.



Gambar 3.22 Cek Koneksi Database

Pada gambar 3.22 merupakan aktifitas cek koneksi *database* kembali agar data pada *database* dapat masuk dan diolah pada sistem *project* yang akan dikerjakan pada Visual Studio. Tahap ini merupakan tahap pertama sebelum melakukan proses pengerjaan *project*.

```

1 reference
protected DataTable getData(string id)
{
    SqlConnection sqlConn = new SqlConnection(koneksi());
    SqlCommand sqlCmd = new SqlCommand(" select * from [DS9-2-FOR-004] where ID = '"+id+"'", sqlConn);
    SqlDataAdapter sqlDA = new SqlDataAdapter();
    DataTable dt = new DataTable();
    sqlDA.SelectCommand = sqlCmd;
    sqlDA.Fill(dt);
    return dt;
}

```

Gambar 3.23 Fungsi Pengambilan Data pada Database

Pada gambar 3.23 merupakan fungsi pengambilan data pada *database* yang telah dibuat sebelumnya pada SQL Server Management. *Table* pada fungsi disesuaikan dengan yang terdapat pada *database* yaitu *table* DS9-2-FOR-004 yang merupakan *table* utama dari semua kumpulan *table* pada *database*. Lalu pada fungsi *where* mengarah kepada ID yang merupakan *primary key* pada *table* utama.

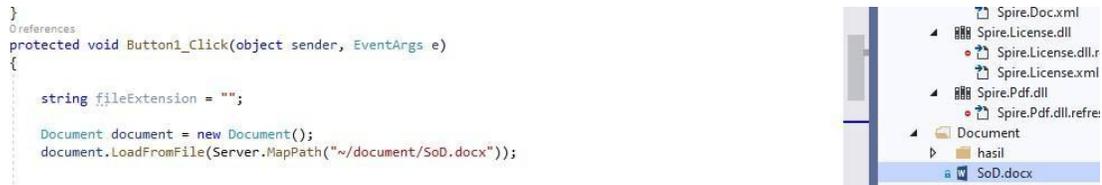
```

1 reference
public string koneksi()
{
    return ConfigurationManager.ConnectionStrings["EFORMConnectionString"].ConnectionString;
}

```

Gambar 3.24 Fungsi Koneksi Database

Pada gambar 3.24 adalah fungsi untuk melakukan koneksi pada *database* yang telah disambungkan sebelumnya pada *web.config*. *Public string* merupakan fungsi untuk memanggil koneksi *database* dari *web.config*.



Gambar 3.25 Load From File SoD

Pada gambar 3.25 merupakan fungsi untuk melakukan load *form* SoD. Pada fungsi `document.LoadFromFile` akan melakukan *direct* instruksi kepada *directory* menuju *folder document* yang selanjutnya fungsi tersebut akan memasukan data yang ditentukan ke dalam `SoD.docx`.

```

// Get From DB
DataTable dt = getData(TextBox1.Text);
if (dt.Rows.Count > 0)
{
    document.Replace("[EQUIP_NO]", dt.Rows[0]["EQUIP_NO"].ToString(), true, true);
    document.Replace("[EQUIP_DESC1]", dt.Rows[0]["EQUIP_DESC1"].ToString(), true, true);
    document.Replace("[EQUIP_DESC2]", dt.Rows[0]["EQUIP_DESC2"].ToString(), true, true);
    document.Replace("[EQUIP_USER]", dt.Rows[0]["EQUIP_USER"].ToString(), true, true);
    document.Replace("[COMMISSION_DATE]", dt.Rows[0]["COMMISSION_DATE"].ToString(), true, true);
    document.Replace("[CHECKED_BY_NAME]", dt.Rows[0]["CHECKED_BY_NAME"].ToString(), true, true);
    document.Replace("[CHECKED_BY_ID]", dt.Rows[0]["CHECKED_BY_ID"].ToString(), true, true);
    document.Replace("[DEFACT_NOTE]", dt.Rows[0]["DEFACT_NOTE"].ToString(), true, true);
}

```

Gambar 3.26 Get Data From DB and Replace

Pada gambar 3.26 merupakan fungsi dalam melakukan *input* data kedalam fungsi `LoadFromFile` sebelumnya. Data-data tersebut menyesuaikan dengan keadaan *form* pada `SoD.docx` dan memasukan data yang diperlukan pada *form* tersebut. Lalu dilakukan metode *replace* data pada fungsi menuju *form*

SoD.docx. Namun, sebelum dilakukan *replace* pada *form* SoD diberikan label terlebih dahulu sesuai dengan data yang akan dimasukkan. TextBox1 merupakan ID dari fungsi pengambilan data.

```
string loc = Server.MapPath("~/document/hasil/SoD_" + TextBox1.Text + ".pdf");  
document.SaveToFile(loc, Spire.Doc.FileFormat.PDF);
```

Gambar 3.27 Fungsi Hasil Replace SoD

Pada gambar 3.27 merupakan fungsi untuk melakukan *print* hasil dari *input* data pada fungsi sebelumnya. Hasil tersebut kemudian akan tersimpan secara otomatis menjadi *form* dengan file *format* PDF.

```
string filePath = Server.MapPath("~/document/hasil/SoD_" + TextBox1.Text + ".pdf");  
fileExtension = System.IO.Path.GetExtension(filePath).ToLower();  
if (fileExtension == ".pdf")  
{  
    Response.ContentType = "application/pdf";  
    Response.AppendHeader("Content-Disposition", "attachment; filename=" + Path.GetFileName(filePath));  
    Response.WriteFile(filePath);  
    Response.End();  
}
```

Gambar 3.28 Fungsi Download Form

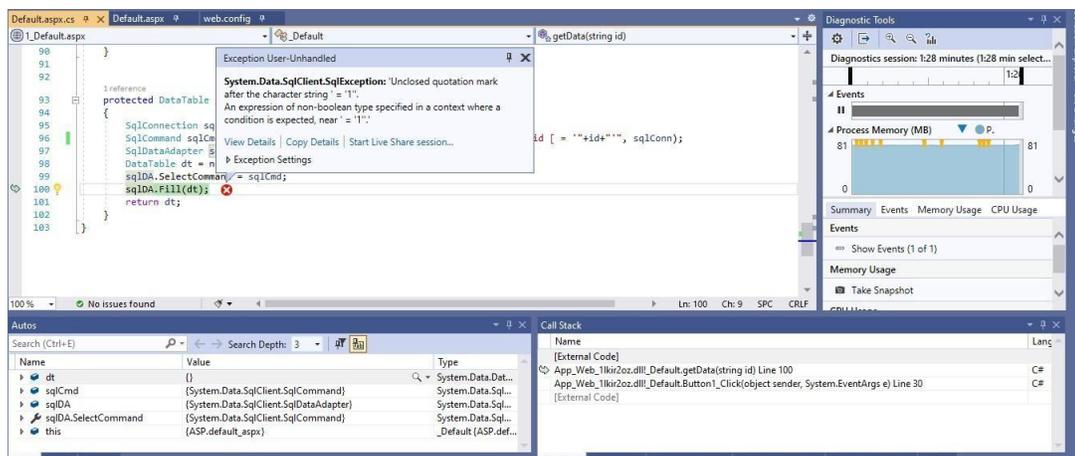
Pada gambar 3.28 merupakan fungsi untuk melakukan *download form* yang telah berbentuk pdf. Fungsi tersebut akan melakukan penulisan secara otomatis

pada *file extension* yang terdapat dalam *document* dan menjadi *e-form* secara otomatis.

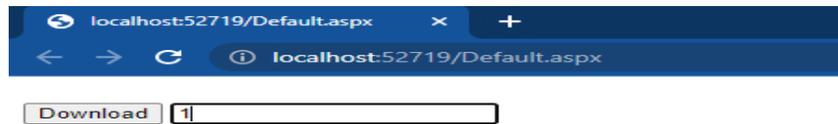
3.3.9 Pengujian Sistem

(Minggu 9)

Pada minggu ini dilakukan kegiatan pengujian kembali pada sistem *project* yang telah dibuat. Aktifitas yang dilakukan adalah pencarian *bug* dan kesalahan atau *error* pada sistem yang telah dirancang sebelum diimplementasikan oleh perusahaan. Kemudian dilakukan pengujian kembali sistem yang telah diperiksa dan dilakukan perbaikan kembali pada sistem. Setelah dilakukan pengujian dan perbaikan maka sistem yang diinginkan telah berjalan sesuai dengan yang diharapkan dan ditentukan ekspetasinya.

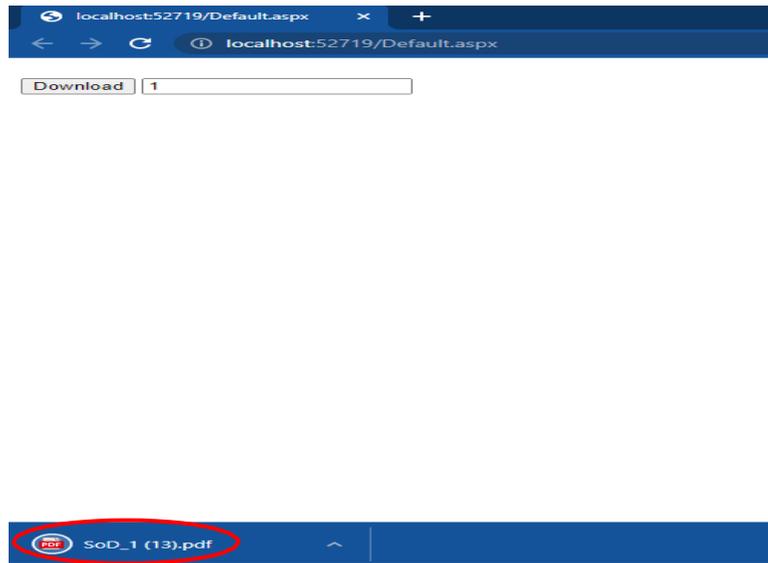


Gambar 3.29 Error pada Sistem



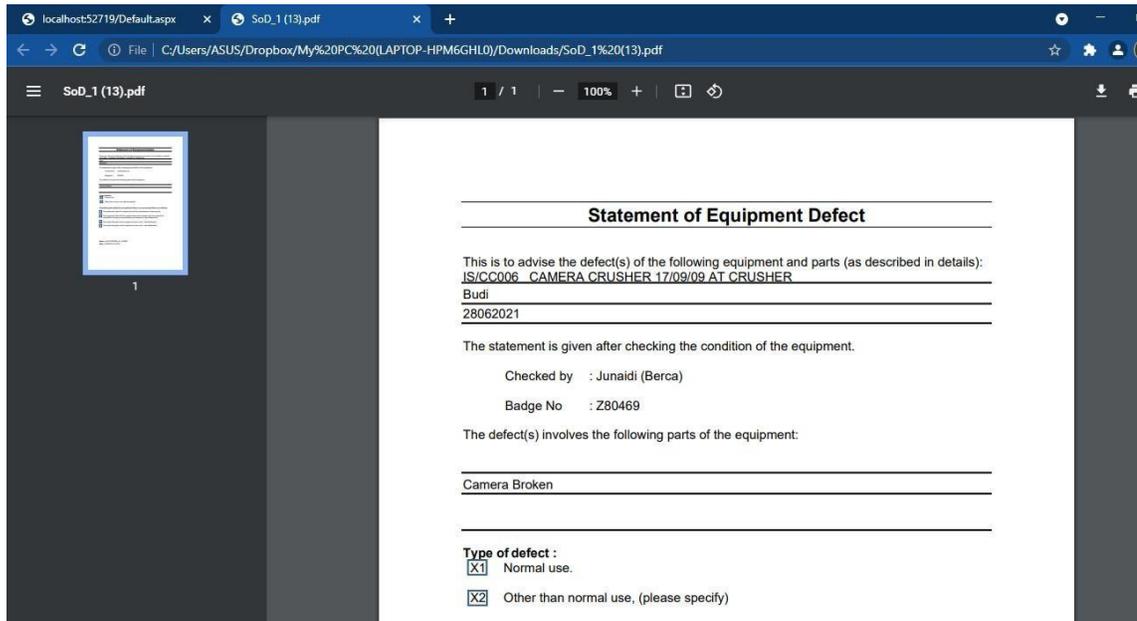
Gambar 3.31 Searching Data

Pada gambar 3.31 merupakan tampilan untuk *searching* data yang ingin dicari. Pada menu *searching* dimasukan data yang ingin dicari pada *database* dan ingin dilakukan penulisan otomatis data tersebut pada *form*. Contoh *sample* data yang diambil adalah data ID 1. Sistem akan secara otomatis mengambil data yang berada pada *table* di dalam *database* dan mencatat data tersebut secara otomatis kedalam file docx.



Gambar 3.32 *Form Otomatis Format PDF*

Pada gambar 3.32 merupakan respon otomatis pada sistem untuk melakukan pencatatan ke dalam *form*. Setelah semua data yang diinginkan masuk selanjutnya *form* tersebut akan menjadi *format pdf* sesuai dengan fungsi yang telah ditentukan sebelumnya pada pengujian dan pengerjaan *code* di dalam sistem Visual Studio.



Gambar 3.33 Hasil Form

Pada gambar 3.33 merupakan hasil dari *form* dan telah terisi secara otomatis setiap data yang berkaitan dengan ID nomor 1 pada *database*. *Form* ini dapat langsung dikirim kepada pihak *admin* untuk dilakukan evaluasi kembali terhadap *equipment* yang rusak dan mengalami kendala. Konsep *e-Form* telah dapat diterapkan pada sistem perusahaan namun masih perlu dilakukan pengembangan kembali agar sistem yang saat ini dikerjakan dapat memberikan manfaat dan keunggulan yang lebih baik.

Pada pembuatan sistem tersebut masih merupakan konsep awal dari bagaimana sistem *e-Form* dapat bekerja. Namun, telah dilakukan pengembangan kembali pada sistem agar tampilan pada sistem lebih menarik dan fungsi pada sistem menjadi lebih baik. Berikut merupakan bentuk dari pengembangan pada sistem.



e-Form IT Portal

Silahkan login untuk masuk ke dalam sistem

Username

Password

Compatible with Google Chrome or Mozilla Firefox Browser

© 2021 | IT AppS - PT Kaltim Prima Coal

Gambar 3.34 Login e-Form

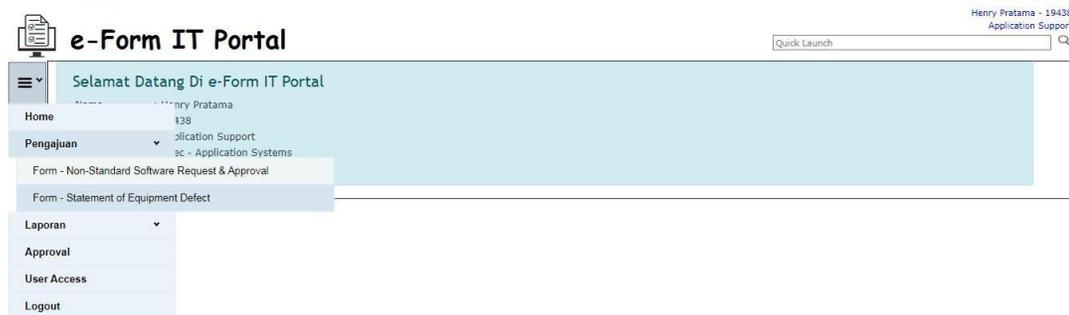
Pada gambar 3.34 merupakan tampilan awal sebelum masuk ke dalam sistem *e-Form*. *User* diharapkan untuk memasukkan data diri berupa *username* dan *password* yang telah terdaftar pada sistem secara otomatis.



Gambar 3.35 Menu Utama e-Form

Pada gambar 3.35 adalah menu utama dari *e-Form*. Di bagian menu utama terdapat notifikasi bahwa *user* telah berhasil masuk berupa teks “

Selamat Datang Di *e-Form IT Portal* “ dan menampilkan biodata dari *user* yaitu nama *user*, *Employee ID*, *Department*, Posisi *user*.



Gambar 3.36 Fitur-fitur *e-Form*

Pada gambar 3.36 merupakan fitur-fitur yang terdapat pada *e-Form*. Fitur-fitur tersebut adalah *Pengajuan*, *Laporan*, *Approval*, *User Access*, dan *logout*. Pada fitur *Pengajuan* memiliki fungsi untuk melakukan permintaan dalam membuat *form Statement of Equipment Defect*. Kemudian pada fitur laporan berisi tentang bentuk-bentuk dari pengajuan laporan yang dilakukan sesuai dengan ID laporan dan data yang dimasukkan sebelumnya pada fitur *Pengajuan*. Lalu terdapat fitur *Approval* yang memiliki fungsi sebagai tempat untuk mengetahui apakah laporan telah disetujui atau belum oleh *admin*. Kemudian terdapat fitur *User Access* yang memiliki fungsi untuk meminta akses kepada *admin* untuk melakukan pengajuan laporan SoD. Fitur yang terakhir adalah *logout* yang memiliki fungsi untuk keluar dari sistem.

Gambar 3.37 Form Request

Pada gambar 3.37 merupakan aktifitas dalam melakukan *request* laporan yang menjadi proses awal untuk melakukan pembuatan *e-Form*. Hal yang perlu dilakukan terlebih dahulu adalah memasukkan nomor *form*, kemudian menentukan *commission date* dari *form* yang ingin diajukan, lalu memasukan nomor dari *equipment*, dan yang terakhir adalah memasukan *user* dari *equipment*.

Gambar 3.38 Attachment Form

Pada gambar 3.38 merupakan fitur *attachment form* yang berfungsi untuk menambahkan lampiran pada *form* yang ingin dibuat. Terdapat *File Name* untuk memasukkan nama dari *file* yang ingin dilampirkan dan kemudian ada *File Upload* yang berfungsi untuk memasukkan lampiran yang terkait dengan *form* yang ingin dibuat.



The screenshot shows the 'e-Form IT Portal' interface. At the top, there is a header with the portal name and a 'Quick Launch' field. Below the header, a navigation bar displays the current form: 'DS9-2-FOR-0004 - Statement of Equipment Defect'. A series of tabs below the navigation bar indicates the workflow steps: 'Form Request', 'Attachment', 'Assign' (which is currently selected and highlighted in blue), 'Equipment Check', 'Approval', 'Review', and 'Update Status & Progress'. The main content area shows the 'Assign' step, featuring a label 'Assigned To' followed by a dropdown menu with the text 'Select User'. Below the dropdown menu is a 'Submit' button.

Gambar 3.39 Assign Form

Pada gambar 3.39 merupakan fitur *Assign Form* dalam pengajuan laporan yang ingin dibuat oleh *user*. Fitur ini berfungsi untuk memberikan keterangan pada laporan bahwa laporan tersebut akan diserahkan atau ditujukan kepada *user* yang akan menerima laporan *e-Form*. Ketika nama penerima telah terisi maka selanjutnya menekan tombol *submit*.

DS9-2-FOR-0004 - Statement of Equipment Defect

Form Request | Attachment | Assign | **Equipment Check** | Approval | Review | Update Status & Progress

Form No. :

Checked by : 19438 - Henry Pratama

The defect involves the following parts of equipment :

Type of Defect : Normal Use
 Other than normal use

Recommendation : The equipment cannot be repaired due to the obsolescence of spare-parts.
 The equipment shan't be repaired because the repair cost is too expensive than purchasing new
 The equipment parts can be repaired at user's cost.
 The equipment parts can be replaced at user's cost.

Gambar 3.40 *Equipment Check*

Pada gambar 3.40 merupakan fitur *Equipment Check* yang memiliki fungsi untuk memberikan keterangan pada laporan tentang aktifitas pemeriksaan pada *equipment* yang mengalami kerusakan.

e-Form IT Portal

DS9-2-FOR-0004 - Statement of Equipment Defect

Form Request | Attachment | Assign | Equipment Check | **Approval** | Review | Update Status & Progress

Supervisor PC-Support : 19438 - Henry Pratama

Gambar 3.41 *Approval*

Pada gambar 3.41 adalah fitur *Approval* untuk mengarahkan kepada pihak yang akan melakukan evaluasi terhadap laporan yang telah dibuat.



Gambar 3.42 Download e-Form

Pada gambar 3.42 merupakan keterangan bahwa *e-Form* telah selesai dibuat dan dapat di *download* lalu kemudian diserahkan kepada *admin* yang akan menerima laporan *e-Form* tersebut untuk selanjutnya dilakukan peninjauan dan evaluasi kembali terhadap *equipment* yang mengalami kerusakan.

Pada sistem yang telah dikerjakan masih memiliki kekurangan yang perlu dilakukan pengembangan kembali agar sistem menjadi lebih baik lagi. Kekurangan tersebut diantaranya adalah :

- Tampilan pada sistem masih perlu perbaikan lagi sehingga diperlukan pengembangan pada UI agar sistem menjadi lebih menarik.
- Masih perlu adanya fitur-fitur tambahan lagi pada sistem agar sistem memiliki banyak fungsi pada pembuatan *e-form*. Salah satu fitur yang dapat ditambahkan ke dalam *e-Form* adalah *Send e-Form* secara otomatis kepada *admin*.

3.4 Kendala yang Dihadapi

Pada proses berjalannya kegiatan praktek kerja magang, terdapat beberapa kendala yang dihadapi mahasiswa yaitu :

1. Waktu kegiatan kerja magang dengan waktu kerja pembimbing yang berbeda sehingga untuk melakukan bimbingan mahasiswa masih sulit menyesuaikan waktu bimbingan dengan pembimbing.
2. Terdapat software yang tidak sesuai dengan ketentuan yang diberikan oleh pembimbing yaitu Visual Studio 10.
3. Pemadaman listrik yang sering terjadi 2 kali dalam sebulan dan jaringan internet wifi portable di rumah yang terkadang terganggu.
4. Kesulitan dalam berkomunikasi secara langsung kepada pembimbing apabila mahasiswa ingin bertanya.

3.5 Solusi Atas Kendala

Untuk mengatasi setiap kendala yang terdapat pada kegiatan praktek kerja magang, terdapat solusi yang dapat diterapkan dalam memudahkan mahasiswa untuk melewati setiap kendala yang ada. Solusi yang dapat diterapkan adalah sebagai berikut :

1. Sebelum mahasiswa melakukan bimbingan, mahasiswa terlebih dahulu menyesuaikan waktu bimbingan dengan jadwal kesediaan dari pembimbing dengan menanyakan waktu yang tepat untuk melakukan bimbingan.
2. Menanyakan kepada pembimbing software *alternative* yang dapat digunakan. Kemudian pembimbing memberikan saran *software* yaitu Visual Studio 2019.
3. Menyesuaikan waktu pemadaman listrik agar proses pengerjaan *project* tidak terganggu dan menyediakan paket data seluler sebagai sumber jaringan internet cadangan apabila jaringan *wifi portable* yang terdapat di rumah mengalami gangguan.

4. Membuat jadwal tatap muka kepada pembimbing melalui media Zoom. Mahasiswa terlebih dahulu mencatat poin-poin penting yang ingin ditanyakan sebelum melakukan bimbingan tatap muka dengan pembimbing.