

**APLIKASI PERHITUNGAN MATURITY LEVEL MENGGUNAKAN
FRAMEWORK COBIT 4.1**

SKRIPSI

Diajukan Kepada Fakultas Teknik
Universitas Muhammadiyah Yogyakarta
Untuk Memenuhi Sebagian persyaratan
Guna Memperoleh
Gelar Sarjana Teknik



Oleh

**M. Nur Aulia Rakhman
NIM 20130140020**

**PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS TEKNIK
UNIVERSITAS MUHAMMADIYAH YOGYAKARTA
2019**

HALAMAN PENGESAHAN I
TUGAS AKHIR
APLIKASI PERHITUNGAN MATURITY LEVEL MENGGUNAKAN
COBIT 4.1
UNIVERSITAS MUHAMMADIYAH YOGYAKARTA

Diajukan Oleh:

M. Nur Aulia Rakhman

20130140020

Telah disetujui oleh:

Pembimbing 1

Pembimbing 2

Asroni, S.T., M.Eng.

Aprilia Kurnianti, S.T., M. Eng.

NIDN: 0526047401

NIDN: 0518048401

HALAMAN PENGESAHAN II

**APLIKASI PERHITUNGAN MATURITY LEVEL MENGGUNAKAN
COBIT 4.1**

Diajukan oleh:

M. Nur Aulia Rakhman

NIM: 20130140020

Skripsi ini telah diperbaiki sesuai saran tim penguji

Pembimbing 1



Asroni, ST., M.Eng.

NIDN: 0526047401

Pembimbing 2



Aprilia Kurnianti, ST., M. Eng.

NIDN: 0518048401

penguji

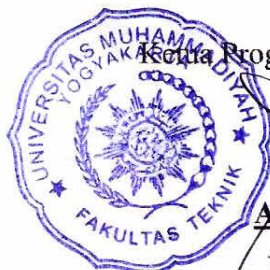


Ir. Haris Setyawan, S.T., M.Eng

NIDN. 0511116901

Mengesahkan

Rektor Program Studi Teknologi Informasi



Asroni, S.T., M.Eng.

NIDN: 0526047401

PERNYATAAN

Dengan ini saya menyatakan dengan sesungguhnya bahwa tugas akhir dengan judul “**Aplikasi Perhitungan Maturity Level Menggunakan Framework COBIT 4.1**” adalah hasil karya saya sendiri dan tidak terdapat karya yang pernah diajukan untuk memperoleh gelar kesarjanaan di suatu Perguruan Tinggi dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis dikutip dalam naskah ini dan disebutkan dalam daftar pustaka.

Yogyakarta, 18 Maret 2019



M. Nur Aulia Rakhman

PRAKATA

Alhamdulillah *rabbi'l'alamin*, puji syukur ke hadirat Allah SWT yang telah melimpahkan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan penyusunan skripsi yang berjudul “APLIKASI PERHITUNGAN *MATURITY LEVEL* MENGGUNAKAN *FRAMEWORK* COBIT 4.1”. Penulis telah menyelesaikan penyusunan skripsi ini dengan berbagai usaha dan upaya, namun penulis meminta maaf apabila terdapat kesalahan dalam pemilihan kata, susunan penulisan dan sistematika pembahasan, karena penulis masih menyadari bahwa masih banyak kekurangan dalam penyusunan skripsi ini.

Dalam penyelesaian serta penyusunan laporan skripsi ini, penulis telah mendapatkan banyak dukungan dan bantuan dari berbagai pihak. Penulis mengucapkan terima kasih banyak kepada:

1. Allah SWT yang senantiasa memberikan kekuatan dan kesehatan, serta rahmat dan hidayah-Nya.
2. Bapak Asroni, S.T., M.Eng. selaku pembimbing I dan ketua Program Studi Teknologi Informasi Universitas Muhammadiyah Yogyakarta yang telah membimbing penulis dengan penuh kesabaran dan ketulusan dalam pengembangan sistem dan penulisan laporan skripsi.
3. Ibu Aprilia Kurnianti, S.T., M.Eng. selaku pembimbing II yang telah meluangkan waktu serta dengan sabar membimbing penulis dalam pembuatan skripsi dan pengembangan sistem.
4. Bapak Ir. Haris Setyawan, S.T., M.Eng. selaku dosen penguji yang bersedia meluangkan waktu untuk memberikan kritik dan saran kepada penulis dalam pembuatan skripsi.
5. Bapak Chayadi Oktomy N S, S.T., selaku pemangku jabatan awal ketua Program Studi Teknologi Informasi yang telah membimbing secara materi penulisan dan judul skripsi.

6. Segenap dosen dan pengajar di Jurusan Teknologi Informasi Universitas Muhammadiyah Yogyakarta yang telah memberikan tambahan pengetahuan dan mengajarkan ilmunya kepada penulis selama perkuliahan.
7. Staff Tata Usaha Jurusan Teknik Informatika Universitas Muhammadiyah Yogyakarta yang senantiasa membantu penulis dalam urusan administrasi.
8. Bapak Sunaryo, Ibu Gotri Lastiti, dan Saudara-Saudara penulis, M. Amien Syafiqurrakhman, M. Faiq Abiyyu Rakhman yang selalu memberikan dukungan, motivasi, kasih sayang, dan doa kepada penulis.
9. Ide Raga Salasa yang selalu menemani dan mendukung penulis dalam menyelesaikan segala masalah yang dihadapi.
10. Semua teman-teman TI kelas A 2013, serta semua angkatan 2013 lainnya yang tidak bisa penulis cantumkan satu persatu, terimakasih banyak untuk kalian semua sebagai teman seperjuangan.
11. Teman-teman bermusik, teman-teman komunitas, dan teman sepermainan lainnya yang tidak bisa penulis sebut satu persatu yang selalu mewarnai hari-hari penulis.
12. Pihak-pihak lain yang tidak dapat disebutkan satu per satu, yang telah memberikan bantuan kepada penulis sehingga penulis dapat menyelesaikan skripsi ini.

DAFTAR ISI

HALAMAN PENGESAHAN I	i
HALAMAN PENGESAHAN II	ii
PERNYATAAN	iii
PRAKATA	iv
DAFTAR ISI	vi
INTISARI	ix
ABSTRACT	x
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	2
1.3. Batasan Masalah.....	2
1.4. Tujuan Aplikasi.....	2
1.5. Manfaat Aplikasi.....	2
1.6. Sistematika Penulisan.....	2
BAB II TINJAUAN PUSTAKA DAN LANDASAN TEORI	4
2.1. Tinjauan Pustaka	4
2.2. Landasan Teori.....	6
2.2.1. Tata Kelola Teknologi Informasi.....	6
2.2.2. COBIT (Control Objective for Information and Related Technology)	7
2.2.3. Domain dan Proses TI pada COBIT 4.1	9
2.2.4. COBIT 4.1 <i>Maturity Level</i>	12

2.2.5.	Perhitungan <i>Maturity Level</i>	14
2.3.	Teknologi Pengembangan Aplikasi	15
2.3.1.	Unifed Markup Language (UML).....	15
2.3.2.	SQL Server.....	17
2.3.3.	Bahasa Pemrograman C#.....	17
BAB III METODE PENELITIAN.....		18
3.1.	Alat dan Bahan.....	18
3.1.1.	Alat.....	18
3.1.2.	Bahan.....	18
3.2.	Alur Penelitian	18
3.3.	Rancangan Sistem dan Basis Data	20
3.3.1.	<i>Use Case Diagram</i>	21
3.3.2.	<i>Activity Diagram</i>	22
3.3.3.	ER Diagram.....	23
3.3.4.	<i>Class Diagram</i>	24
3.4.	Rancangan Antar Muka.....	25
3.4.1.	Rancangan Antar Muka Halaman <i>Login</i>	25
3.4.2.	Rancangan Antar Muka Halaman Utama	26
3.4.3.	Rancangan Antar Muka Halaman Hasil Kuisisioner.....	27
3.4.4.	Rancangan Antar Muka Halaman Rekapitulasi Data.....	28
BAB IV HASIL DAN PEMBAHASAN		29
4.1.	<i>Database</i>	29
4.1.1.	Definisi <i>Database</i>	30
4.2.	Implementasi Antarmuka.....	32
4.2.1.	Antarmuka Halaman <i>Login</i>	32
4.2.2.	Antarmuka Halaman Tambah Pengguna	33
4.2.3.	Antarmuka Halaman Utama.....	34
4.2.4.	Antarmuka Halaman Hasil Kuisisioner	35

4.2.5.	Antarmuka Halaman Rekapitulasi Data.....	36
4.3.	Implementasi Pengujian Sistem.....	37
4.3.1.	Pengujian User <i>Interface</i>	37
4.3.2.	Pengujian Validasi	38
4.3.3.	Pengujian Fungsi Dasar Sistem.....	39
4.4.	Pembahasan.....	40
BAB V KESIMPULAN DAN SARAN.....		41
5.1.	Kesimpulan	41
5.2.	Saran.....	41
DAFTAR PUSTAKA.....		42
LAMPIRAN		44

INTISARI

Dalam setiap lembaga atau instansi diperlukan adanya tata kelola teknologi informasi yang baik, salah satunya dengan mengaplikasikan metode perhitungan *maturity level* menggunakan *framework* COBIT 4.1. Dalam praktiknya, proses perhitungan *maturity level* yang ditemui saat ini masih menggunakan cara manual, yaitu dengan software Microsoft Excel. Untuk menjalankan perhitungan yang lebih efisien dan terfokus, maka dibutuhkan aplikasi yang berfungsi khusus sebagai penghitung *maturity level* yang dapat lebih mempermudah audit dalam pengumpulan data dari kuisisioner maupun proses penilaian hasil akhir *maturity level* itu sendiri. Aplikasi yang dibutuhkan dikomputerisasi dengan berbasis desktop. Aplikasi desktop dibuat dengan sistem *windows form* dan menggunakan bahasa pemrograman *c#*. Dalam hal pengelolaan data, aplikasi ini dihubungkan dengan basis data yang dibuat dengan menggunakan Microsoft SQL Server 2014. Fungsi utama aplikasi desktop ini adalah mengambil nilai dari kuisisioner dan menghitungnya sehingga menghasilkan *maturity value* dari proses TI di dalam sebuah lembaga atau instansi. Berdasarkan hasil pengujian, aplikasi desktop ini berjalan dengan baik sesuai dengan rancangan. Diharapkan aplikasi ini mampu membantu audit dalam hal perhitungan *maturity level* demi tata kelola teknologi informasi yang lebih baik.

Kata kunci: Aplikasi Desktop, *Maturity Level*, *C#*, SQL Server

ABSTRACT

In each institution or agency required the existence of corporate governance of information technology is good, one of them by applying the maturity level measurement method using framework COBIT 4.1. In real world, measurement of process maturity level found is currently still use the manual way, IE with the Microsoft Excel software. To run a more efficient and focused measurement, required applications that serve specifically as a maturity level counters that can facilitate audit in the collection of data from questionnaire or assessment of IT process result maturity level itself. The required application computerized with desktop-based. Desktop applications created with the windows form system and using the c# programming language. In terms of the management of data, this application is associated with the database that was created by using Microsoft SQL Server 2014. The main function of this desktop application is retrieving the value of the questionnaire and number them so as to produce the maturity value of the IT processes within an institution or agency. Based on the test results, this desktop application is running properly in accordance with the draft. This application is expected be able to help the audit in terms of the measurement of maturity level for the sake of information technology governance.

Keywords: Desktop application, Maturity Level, C #, SQL Server

BAB I PENDAHULUAN

1.1. Latar Belakang

Perkembangan teknologi informasi sebagai media komunikasi data hingga saat ini berkembang sangat pesat. Pengelolaan sistem informasi yang tepat dalam sebuah lembaga atau instansi akan sangat membantu lembaga atau instansi tersebut. Teknologi memiliki peranan yang sangat krusial bagi sebuah lembaga, namun yang menjadi masalah adalah sering terjadi penggunaan teknologi informasi yang ternyata tidak sesuai dengan harapan pada lembaga atau instansi, oleh karena itu diperlukan tata kelola terhadap penggunaan teknologi informasi dalam suatu lembaga atau instansi.

Tujuan utama dari tata kelola teknologi informasi yang baik adalah untuk mengetahui sejauh mana kinerja proses teknologi informasi itu berjalan dalam sebuah lembaga atau instansi. Salah satu cara yang dilakukan adalah dengan proses perhitungan *maturity* level menggunakan *framework* COBIT 4.1 yang merupakan *framework* pengelolaan teknologi informasi.

Proses tata kelola teknologi informasi dengan menggunakan *framework* COBIT 4.1 sudah dilakukan di berbagai lembaga atau instansi. Namun dalam pengaplikasiannya masih memiliki kekurangan, yaitu di sebagian lembaga atau instansi masih menggunakan aplikasi manual yang sudah tidak bisa memenuhi kebutuhan tata kelola teknologi informasi secara efisien.

Berdasarkan kondisi yang telah dipaparkan di atas, maka penulis memiliki sebuah ide untuk membuat aplikasi yang lebih memudahkan dalam memenuhi kebutuhan tata kelola teknologi informasi menggunakan *framework* COBIT 4.1 dengan lebih efisien.

1.2. Rumusan Masalah

Sebagian lembaga atau instansi belum menggunakan aplikasi yang lebih efisien dalam proses perhitungan *maturity level* menggunakan *framework* COBIT 4.1 sehingga proses perhitungan kinerja proses teknologi informasi dalam lembaga atau instansi tersebut kurang berjalan dengan efisien.

1.3. Batasan Masalah

Batasan masalah dalam aplikasi perhitungan *maturity level* yang penulis buat adalah aplikasi ini dibuat untuk perhitungan *maturity level* menggunakan *framework* COBIT 4.1. Aplikasi ini hanya dibuat sebatas pemakaian lokal karena penelitian ini terfokus pada perhitungan saja.

1.4. Tujuan Aplikasi

Tujuan aplikasi ini adalah untuk memudahkan audit dalam proses perhitungan *maturity level* dalam sebuah lembaga atau instansi.

1.5. Manfaat Aplikasi

Manfaat aplikasi ini adalah terukurnya kinerja tata kelola teknologi informasi agar lebih efisien.

1.6. Sistematika Penulisan

Sistematika penulisan dari pembuatan laporan Tugas Akhir mengacu pada format penulisan sebagai berikut:

BAB I PENDAHULUAN

Bab I berisi tentang latar belakang pemilihan judul, tujuan, batasan masalah, dan manfaat aplikasi, dan metodologi penulisan.

BAB II TINJAUAN PUSTAKA DAN LANDASAN TEORI

Bab II memaparkan dasar-dasar teori mengenai Tata Kelola Teknologi Informasi dan COBIT 4.1 yang mendukung dan melandasi dibuatnya aplikasi perhitungan *maturity level* menggunakan *framework* COBIT 4.1.

BAB III METODOLOGI PENELITIAN

Bab III megnutarakan tentang analisis kebutuhan seperti alat dan bahan dalam membangun aplikasi, dan rancangan pensusunan sistem yang berjalan pada aplikasi apakah sesuai dengan perangkat lunak yang digunakan.

BAB IV HASIL DAN PEMBAHASAN

Bab IV menguraikan hasil dari pembuatan aplikasi, kemudian menganalisa agar aplikasi berjalan sesuai dengan perancangan.

BAB V KESIMPULAN

Bab V merupakan bab penutup yang berisi tentang kesimpulan dan saran-saran yang diambil untuk perkembangan lebih lanjut.

BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1. Tinjauan Pustaka

Tata kelola TI menggunakan *framework* COBIT 4.1 telah banyak diteliti dan hasil penelitiannya telah banyak membantu instansi memperbaiki tata kelola teknologi informasi dengan baik.

Rujukan penelitian tata kelola teknologi informasi terkait yang pernah dilakukan untuk mendukung penulisan antara lain:

1. Irania Dwi Wijayanti (2016) dengan judul *Analisis Maturity Level* Proses Teknologi Informasi Menggunakan *Framework* COBIT 4.1 di BSI UMY. Hasil penelitian ini menunjukkan tingkat kematangan (*maturity level*) yang ada pada setiap proses TI yang terdapat dalam *domain Plan and Organise 3* (PO3). Penelitian tersebut dilakukan dengan pengisian kuisioner oleh responden di BSI UMY dan *maturity value* dihitung dengan menggunakan Microsoft Excel.
2. Goldi Mahardika Muhammad (2017) dengan judul *Pengukuran Maturity Level* Proses TI AI3, AI4, dan DS4 Menggunakan COBIT 4.1 Pada Dinas Kominfo Bantul. Penelitian menghasilkan kesimpulan bahwa Dinas Kominfo Bantul telah mengoperasikan infrastruktur TI dalam setiap kegiatan, namun belum membuat laporan-laporan kegiatan. Analisis pengukuran *maturity level* tersebut dihitung menggunakan *software* Microsoft Excel.
3. Jani Widiastuti (2017) dengan judul *Pengukuran Maturity Level* Proses TI di BSI UMY Menggunakan *Framework* COBIT 4.1. Kesimpulan yang dihasilkan adalah BSI UMY belum mempunyai indikator yang jelas terkait keberhasilan dari penerapan perencanaan strategi TI sebelumnya. Pengukuran tersebut dilakukan menggunakan *software* Microsoft Excel.

4. Noor Azizah (2017) yang melakukan penelitian Audit Sistem Informasi menggunakan *framework* COBIT 4.1 Pada E-Learning UNISU Jepara. Penelitian tersebut didokumentasikan dengan menggunakan *software* Microsoft Excel.
5. Umi Markhamah (2017) berjudul Pengembangan Sistem Informasi Koperasi Simpan Pinjam Berbasis Dekstop dengan melakukan implementasi secara langsung di Koperasi Sari Usaha Tani Jambi. Aplikasi desktop tersebut diintegrasikan dengan basis data sebagai penyimpanan data simpan pinjam di Koperasi Sari Usaha Tani.
6. Itha Aprliai Prihatini (2018) yang berjudul Sistem Penjadwalan Pendadaran atau Ujian Tugas Akhir Berbasis Dekstop yang mengaplikasikannya di Program Studi Teknologi Informasi UMY.

Berdasarkan beberapa tinjauan pustaka di atas, maka penulis membuat aplikasi pengukur *maturity level* menggunakan *framework* COBIT 4.1 menggunakan bahasa pemrograman C# dan berbasis desktop dengan tampilan windows form sebagai pilihan baru dari metode perhitungan sebelumnya yang masih mengukur secara manual menggunakan Microsoft Excel.

Sistem aplikasi berbasis dekstop juga telah banyak digunakan dalam pembuatan aplikasi yang mudah diakses dan digunakan pada berbagai organisasi maupun instansi. Maka, aplikasi pengukur *maturity level* menggunakan *framework* COBIT 4.1 ini dapat dengan baik dan selaras jika dibuat dengan metode dan proses berbasis desktop dengan memiliki beberapa fitur seperti:

1. Aplikasi dapat mengambil data pertanyaan kuisisioner secara otomatis.
2. Aplikasi dapat menghitung hasil perhitungan *maturity level* secara otomatis.
3. Proses pemilihan proses TI, penilaian masing-masing pertanyaan, perhitungan hasil akhir, penyampaian data rekapitulasi, dan pencetakan hasil akhir terdapat pada satu aplikasi.

2.2. Landasan Teori

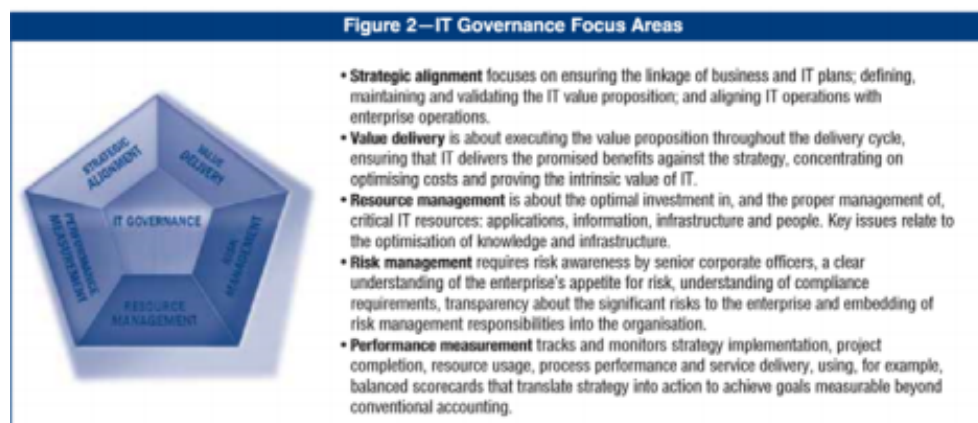
2.2.1. Tata Kelola Teknologi Informasi

Pengertian dari Tata Kelola Teknologi Informasi adalah tanggung jawab dewan direktur dan manajemen eksekutif yang terdiri atas kepemimpinan, struktur organisasi dan proses yang memastikan bahwa TI perusahaan mendukung dan memperluas strategi dan tujuan perusahaan (IT ITGI, 2007). Tata Kelola Teknologi Informasi mencakup informasi, teknologi, dan komunikasi, bisnis, dan hukum serta isu yang lainnya yang melibatkan hampir seluruh pemangku kepentingan (*stakeholders*).

Tata kelola teknologi informasi (*IT Governance*) adalah suatu struktur dan proses yang saling terkait serta mengarahkan dan mengendalikan suatu organisasi dalam mencapai tujuan organisasi dengan menyeimbangkan antara resiko dan manfaat dari teknologi informasi serta prosesnya. Maka dari penjelasan di atas dapat disimpulkan bahwa dalam rangka pencapaian tujuan organisasi pengaplikasian tata kelola teknologi informasi adalah pengelolaan yang berkaitan dengan organisasi. (IT ITGI, 2007)

Fokus area tata kelola teknologi informasi dibagi menjadi 5 bagian, yaitu:

Strategic Alignment, Value Delivery, Resource Management, Risk Management, dan Performance Measurement. Digambarkan seperti gambar 2.1 di bawah ini :



Gambar 2.1 *IT Governance Focus Area (IT ITGI, 2007)*

1. **Strategic Alignment:** memastikan hubungan relasi antara bisnis dengan rangkaian rencana teknologi informasi, pemeliharaan, dan validasi nilai-nilai teknologi informasi untuk menyelaraskan tujuan bisnis dan tujuan teknologi teknologi.

2. **Value Delivery:** menjalankan penempatan dari nilai seluruh siklus *delivery*, memastikan bahwa teknologi informasi dapat bermanfaat sesuai dengan tujuan bisnis yang direncanakan dalam strategi, serta perhitungan biaya untuk mengoptimalkan dan membuktikan nilai intrinsik dari teknologi informasi.

3. **Resource Management:** investasi yang efisien dan optimal dalam pengelolaan sumber daya teknologi informasi seperti informasi, aplikasi, SDM, infrastruktur dan pengoptimalisasian infrastruktur.

4. **Risk Management:** mengoptimalkan kesadaran dalam hal pengelolaan risiko oleh petinggi jabatan pada perusahaan, serta bagaimana mendeskripsikan persyaratan kepatuhan, keterbukaan tentang risiko yang signifikan terhadap perusahaan dalam menanamkan tanggung jawab manajemen risiko di sebuah lembaga atau organisasi.

5. **Performance Measurement:** analisa kinerja dan implementasi dari penggunaan sumber daya, penyelesaian proyek, strategi, kinerja proses, dan beberapa tugas lainnya, misalnya : *balanced scorecard* yang menerjemahkan strategi ke dalam tindakan untuk mencapai tujuan yang terukur.

2.2.2. COBIT (Control Objective for Information and Related Technology)

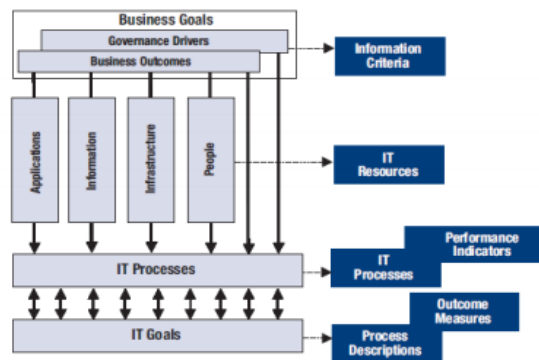
Control Objective for Information and Related Technology (COBIT) adalah seperangkat pedoman umum untuk manajemen TI yang dibuat oleh Information System Audit and Control Association (ISACA), dan IT Governance Institute (ITGI) pada tahun 1992 untuk kebutuhan-kebutuhan informasi perusahaan dalam proses mencapai tujuan yang diinginkan, oleh karena itu berikut beberapa prinsip dasar COBIT meliputi (Simonsson & Johnson, 2006):

1. *Business Information Requirement* yaitu berupa pedoman unsur-unsur yang harus ada di dalam sebuah informasi antars lain *effectiveness* (efektif), *efficiency*

(efisien), *confidentiality* (keyakinan), *integrity* (integritas), *availability* (tersedia), *compliance* (pemenuhan), *reliability* (dipercaya).

2. *IT Resource*, terdiri dari pengguna (*people*), aplikasi (*application*), teknologi (*technology*), infrastruktur (*infrastructure*), informasi (*information*), *Database Management System*, *Hardware*, *Software* dan *Multimedia*.

3. *High Level IT Process*, terdiri dari: Proses TI *Planning and Organization* (PO), *Acquisition and Implementation* (AI), *Delivery Support* (DS), dan *Monitoring and Evaluation* (ME).



Gambar 2.2 COBIT *management, control, alignment and monitoring* (IT ITGI, 2007)

Tujuan utama dari dibuatnya COBIT adalah memberikan kebijaksanaan yang jelas dan latihan yang bagus untuk Tata Kelola Teknologi Informasi, yang digunakan pada organisasi di seluruh dunia dengan harapan dapat membantu proses pengelolaan untuk memahami dan mengatur resiko yang berhubungan dengan Teknologi Informasi. Cara kerja COBIT adalah dengan menyediakan kerangka kerja IT Governance dan petunjuk control obyektif yang rinci bagi manajemen, pemilik proses bisnis, pemakaian dan auditor.

COBIT merupakan sebuah kerangka keseluruhan yang dapat membantu perusahaan dalam mencapai tujuannya pada tata kelola teknologi informasi di sebuah instansi atau perusahaan. Contohnya yaitu COBIT 4.1 membantu perusahaan menciptakan nilai optimal dari teknologi informasi dengan cara

menjaga keselarasan antara mengoptimalkan tingkat resiko penggunaan sumber daya dan pendapatan keuntungan. COBIT 4.1 bersifat umum dan digunakan untuk segala jenis ukuran perusahaan, baik itu sektor non profit, sektor komersial atau pada sektor pemerintahan dan publik.

2.2.3. Domain dan Proses TI pada COBIT 4.1

Aktivitas TI pada COBIT 4.1 didefinisikan pada 4 (empat) proses TI yaitu (IGI ITGI, 2007):

A. *Plan and Organise (PO)*

Domain PO mengutarakan proses yang harus dijalani untuk mengidentifikasi metode agar Teknologi Informasi dapat memberikan kontribusi dalam pencapaian tujuan bisnis organisasi, serta merencanakan, mengkomunikasikan, dan mengelola visi yang ingin dicapai organisasi. Proses TI PO terdiri dari 10 *control objectives*, yaitu:

1. **PO1:** *define a strategic IT plan* (menetapkan rencana strategi TI).
2. **PO2:** *define the information architecture* (menentukan arsitektur informasi).
3. **PO3:** *determine technological direction* (menentukan arah teknologi)
4. **PO4:** *define the IT processes, organization and relationships* (menetapkan proses TI, organisasi dan hubungan).
5. **PO5:** *manage the IT investement* (mengelola investasi TI).
6. **PO6:** *communicate management aims and direction* (mengkomunikasikan tujuan dan arah manajemen).
7. **PO7:** *manage IT human resource* (mengelola sumber daya manusia).
8. **PO8:** *manage quality* (mengelola kualitas)
9. **PO9:** *assess and manage IT risks* (menilai dan mengelola resiko TI)
10. **PO10:** *manage project* (mengelola proyek).

B. *Acquire and Implement (AI)*

Domain AI berisi tentang proses-proses yang dilakukan dengan cara mengidentifikasi, membangun atau menyediakan aplikasi Teknologi Informasi untuk mewujudkan rencana Teknologi Informasi. Selain itu, perancangan yang dilakukan dan pemeliharaan terhadap sistem TI juga menjadi cakupan domain ini. Proses TI *AI* terdiri dari 7 *control objectives*, yaitu:

1. **AI1:** *identify automated solutions* (identifikasi solusi otomatis)
2. **AI2:** *acquire and maintain application software* (memperoleh dan memelihara aplikasi perangkat lunak)
3. **AI3:** *acquire and maintain technology infrastructure* (memperoleh dan memelihara infrastruktur teknologi)
4. **AI4:** *enable operation and use* (pengaktifan operasi dan penggunaan)
5. **AI5:** *procure IT resource* (pengadaan sumberdaya TI)
6. **AI6:** *manage changes* (mengelola perubahan)
7. **AI7:** *install and accredit solutions and changes* (install dan akreditasi solusi dan perubahan).

C. *Deliver and Support (DS)*

Domain DS fokus pada memberikan dukungan agar pencapaian hasil sistem Teknologi Informasi sesuai dengan yang diharapkan. Proses ini secara garis besar terdiri dari keamanan, aspek kontinuitas, sampai dengan memberikan pelatihan kepada pengguna. Proses TI *DS* terdiri dari 13 *control objective*, yaitu:

1. **DS1:** *define and manage service levels* (mendefinisikan dan mengelola tingkat layanan)
2. **DS2:** *manage third-party service* (mengelola layanan pihak ketiga)
3. **DS3:** *manage performance and capacity* (mengelola kinerja dan kapasitas)
4. **DS4:** *ensure continuous service* (memastikan layanan secara terus menerus)
5. **DS5:** *ensure system security* (memastikan sistem keamanan)

6. **DS6:** *identify and allocate costs* (mengidentifikasi dan mengalokasikan biaya)
7. **DS7:** *educate and train users* (mendidik dan melatih pengguna)
8. **DS8:** *manage service desk and incidents* (mengelola pelayanan meja dan insiden)
9. **DS9:** *manage the configuration* (mengelola konfigurasi)
10. **DS10:** *manage problems* (mengelola masalah)
11. **DS11:** *manage data* (mengelola data)
12. **DS12:** *manage the physical environment* (mengelola lingkungan fisik)
13. **DS13:** *manage operations* (mengelola operasi).

D. Monitor and Evaluate (ME)

Proses TI perlu dinilai secara berkala dari waktu ke waktu untuk kualitas dan pemenuhan persyaratan dan ini berlaku untuk semua Proses TI. Proses TI membahas pemantauan pengendalian internal, manajemen kerja, kepatuhan terhadap tata kelola yang telah diatur. Proses TI ME terdiri dari 4 *control objective*, yaitu:

1. **ME1:** *monitor and evaluate IT performance* (memonitor dan mengevaluasi kinerja TI)
2. **ME2:** *monitor and evaluate internal control* (memonitor dan mengevaluasi pengendalian internal)
3. **ME3:** *ensure compliance with external requirements* (memastikan kepatuhan terhadap persyaratan eksternal)
4. **ME4:** *provide IT Governance* (menyediakan pengelolaan TI).

2.2.4. COBIT 4.1 *Maturity Level*

(IGI ITGI, 2007) *Maturity Model* adalah sebuah metode yang digunakan untuk mengukur sejauh mana kapabilitas manajemen dalam tingkat pengembangan manajemen proses. Tercapainya tujuan-tujuan COBIT yang sudah diterapkan berguna untuk melihat bagus atau tidaknya suatu pengembangan atau kapabilitas manajemen. Salah satu alat perhitungan kinerja di suatu sistem teknologi informasi adalah level kematangan (*Maturity Level*).

Untuk menentukan *maturity level* dalam pengelolaan dan pengendalian pada proses teknologi informasi harus didasarkan pada metode evaluasi organisasi sehingga dapat mengevaluasi sendiri dari level 0 (tidak ada) hingga level 5 (optimal). Penggunaan *maturity level* yang dikembangkan untuk setiap 34 proses teknologi informasi memungkinkan manajemen dapat mengidentifikasi:

1. Kondisi instansi atau perusahaan saat ini
2. Kondisi industri sekarang untuk perbandingan
3. Kondisi instansi atau perusahaan dari standar internasional sebagai perbandingan tambahan
4. Kondisi yang diinginkan instansi atau perusahaan.



Gambar 2.3 Grafik *Maturity Model*

Berikut ini merupakan tabel pengelompokan berdasarkan nilai *maturity level* yang telah terinci:

Tabel 2.1 Index *Maturity Level*

<i>Maturity Indexes</i>	<i>Maturity Level</i>
0-49	0-Non Existent
0.50-1.49	1-Initial / Ad Hoc
1.50-2.49	2-Repeateble But Intuitive
2.50-3.49	3-Defined Process
3.50-4.49	4-Managed And Measurable
4.50-5.00	5-Optimized

Penjelasan setiap *maturity level* adalah sebagai berikut:

1. **Non-Exsistent** (0 = *Management process are not applied at all*), pernyataan minimal dan menyeluruh terhadap semua proses yang dapat dikenali. Bahkan perusahaan tidak mengetahui bahwa terhadap permasalahan yang harus diatasi dan solusi yang harus diambil.
2. **Ad-Hoc** (1 = *Processes are ad hoc and disorganized*), Terdapat bukti bahwa perusahaan mengetahui adanya permasalahan yang harus diatasi. Bagaimanapun juga tidak terdapat proses standar, namun menggunakan pendekatan *ad hoc* yang cenderung diperlakukan secara individu atau per kasus. Secara umum pendekatan kepada pengelolaan proses tidak terorganisasi.
3. **Reapetable** (2 = *Process or alow a regular pattern*), Proses dikembangkan kedalam tahapan dimana prosedur serupa diikuti oleh pihak-pihak yang berbeda untuk pekerjaan yang sama. Tidak terdapat pelatihan formal atau pengkomunikasian prosedur standard an tanggung jawab diserahkan kepada individu masing-masing.
4. **Defined** (3 = *Processes are documented and communicated*), Prosedur distandarisasi dan didokumentasikan kemudian dikomunikasikan melalui pelatihan. Kemudian diamanatkan bahwa proses-proses tersebut harus diikuti. Namun penyimpangan tidak mungkin dapat terdeteksi. Prosedur sendiri tidak lengkap namun sudah menformalkan praktek yang berjalan.

5. **Managed** (4 = *Processes are monitored and measured*), Manajemen telah mengukur dan mengawasi ketaatan terhadap prosedur dan mengambil tindakan jika proses tidak dapat dikerjakan secara efektif, proses ini berada dibawah penyediaan praktek yang baik dan peningkatan yang konstan. Otomatisasi dan perangkat digunakan dalam batasan tertentu.
6. **Optimized** (5 = *Best practices are followed and automated*), Proses telah dipilih ke dalam tingkat praktek yang baik, berdasarkan hasil dari perbaikan berkelanjutan dan permodelan kedewasaan dengan perusahaan lain. Teknologi Informasi digunakan sebagai proses integrasi untuk mengotomatisasi alur kerja, penyediaan alat untuk peningkatan kualitas dan efektifitas, serta mewujudkan sebuah perusahaan yang dapat beradaptasi.

2.2.5. Perhitungan *Maturity Level*

Perhitungan level kematangan pada setiap proses Ti di dalam COBIT 4.1 menggunakan skala penilaian yang tertera pada tabel 2.2 (Laksito & Luthfi, 2013):

Tabel 2.2 Skala Penilaian

<i>Value Index</i>	Statements Compliance Values
Tidak benar sama sekali	0
Ada benarnya	0,33
Sebagian besar benar	0,66
Sepenuhnya benar	1

Rumusan perhitungan total nilai *maturity level* di dalam COBIT 4.1 adalah dengan cara seperti pada gambar 2.4.

$$\begin{array}{l}
 \text{Maturity level compliance value (a)} \\
 = \frac{\text{Sum of statements compliance values}}{\text{Number of Maturity level statements}} \\
 \\
 \text{Normalized compliance value (b)} \\
 = \frac{\text{(a)}}{\text{Sum of Maturity level compliance value}}
 \end{array}$$

Gambar 2.4 Rumusan Perhitungan *Maturity Level*

2.3. Teknologi Pengembangan Aplikasi

2.3.1. Unifed Markup Language (UML)

Unifed Markup Language (UML) merupakan bahasa pemodelan secara grafis yang digunakan untuk mespesifikasikan, memvisualkan, membangun, dan mendokumentasikan seluruh rancangan aplikasi perangkat lunak, menurut (Booch & Ivar, n.d.). UML adalah bahasa standar untuk membuat rancangan software. UML biasanya digunakan untuk menggambarkan dan membangun dokumen artifak dari software.

Penggunaan model ini bertujuan untuk mengidentifikasi bagian-bagian yang termasuk dalam lingkup sistem di dalam aplikasi. Model UML yang dipakai dalam pengembangan aplikasi ini antara lain *Use Case Diagram*, *ER Diagram*, dan *Class Diagram*.

a. *Use Case Diagram*

Use Case merupakan salah satu UML yang digunakan dalam membuat desain sebuah aplikasi. *Use Case* menjelaskan interaksi yang terjadi antara “aktor” yang mana inisiator dari interaksi sistem itu sendiri dengan sistem yang ada (Brady & Loonam, 2010). *Use Case* memiliki beberapa komponen yaitu:

1. Aktor

Menggambarkan seseorang yang berinteraksi dengan sistem, di mana hanya dapat meng-*input*-kan informasi dan menerima informasi dari sistem dan tidak memegang kendali pada *Use Case*.

2. *Use Case*

Gambaran fungsional sistem yang akan dibuat, agar pengguna lebih mengerti penggunaan sistem.

b. Entity Relationship Diagram

Entity Relationship Diagram (ERD) adalah suatu Teknik yang digunakan untuk memodelkan diagram untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang memiliki relasi, biasanya oleh *system analysis* dalam tahap analisis persyaratan proyek pengembangan sistem (Brady & Loonam, 2010). ERD memiliki tiga komponen yaitu:

1. Entitas

Entitas memiliki simbol yang biasanya digambarkan dengan persegi panjang. Entitas merupakan objek yang mewakili elemen yang nyata dan dapat dibedakan dari satu sama lain.

2. Atribut

Setiap entitas memiliki elemen yang disebut dengan atribut yang berfungsi untuk mendeskripsikan karakteristik pada entitas tersebut. Isi dari atribut mempunyai sesuatu yang dapat mengidentifikasi elemen satu dengan yang lain.

3. Relasi

Relasi merupakan hubungan antara kumpulan dari sebuah entitas yang berasal dari himpunan entitas yang berbeda.

c. Class Diagram

Class Diagram adalah salah satu model statis yang menggambarkan struktur dan deskripsi *class* serta hubungan antar *class*. *Class diagram* memberikan pandangan secara luas dari suatu sistem dengan menunjukkan kelas dan hubungan mereka (“4_Class Diagram.pdf,” n.d.). *Class diagram* mirip dengan ER Diagram pada perancangan database, bedanya pada ER diagram tidak terdapat operasi atau *method* tapi hanya atribut. *Class diagram* memiliki empat elemen yaitu *Class*, *Attributes*, *Operations*, dan *Relationships*. (“The class diagram,” 2004).

2.3.2. SQL Server

Microsoft SQL Server merupakan sistem manajemen basis data relasional yang dirancang untuk aplikasi dengan arsitektur *client server*. Fitur pada Microsoft SQL Server yaitu mempunyai kemampuan untuk membuat basis data mirroring dan clustering. Microsoft SQL Server juga mendukung SQL sebagai bahasa untuk memproses *query* ke dalam basis data dan kita tahu bahwa SQL ini sudah digunakan secara umum pada semua produk *database server* (Delano & George, 2005).

2.3.3. Bahasa Pemrograman C#

C# adalah sebuah bahasa pemrograman yang diciptakan oleh microsoft dibawah kepemimpinan Andres Hejlsberg yang sebelumnya juga telah menciptakan berbagai macam bahasa pemrograman termasuk C++, Borland Turbo dan Borland Dephi. Bahasa C# juga telah menjadi standar yang diakui internasional oleh ECMA. C# bisa digunakan untuk membangun berbagai macam jenis aplikasi, seperti aplikasi berbasis Windows (desktop) dan aplikasi berbasis *website* serta aplikasi berbasis web services seperti halnya pemrograman yang lain, (Enterprise, 2016).

C# didesain untuk membangun jangkauan aplikasi *enterprise* yang berjalan di atas framework .NET.

BAB III

METODE PENELITIAN

3.1. Alat dan Bahan

Dalam pembuatan aplikasi ini digunakan alat dan bahan untuk mendukung berjalannya perancangan dan pembuatan aplikasi.

3.1.1. Alat

Alat yang digunakan dalam pembuatan aplikasi ini berupa perangkat keras dan perangkat lunak.

a. Perangkat keras

Personal Computer (PC) atau Laptop dengan spesifikasi:

- Processor Intel(R) Core(TM) i7-3632QM CPU @ 2.20GHz
- 4GB installed memory RAM

b. Perangkat lunak

1. Visual Studio 2015
2. Microsoft SQL Server 2014
3. Microsoft SQL Management Studio 2014
4. Microsoft Visio 2016
5. Sistem Operasi Windows 10

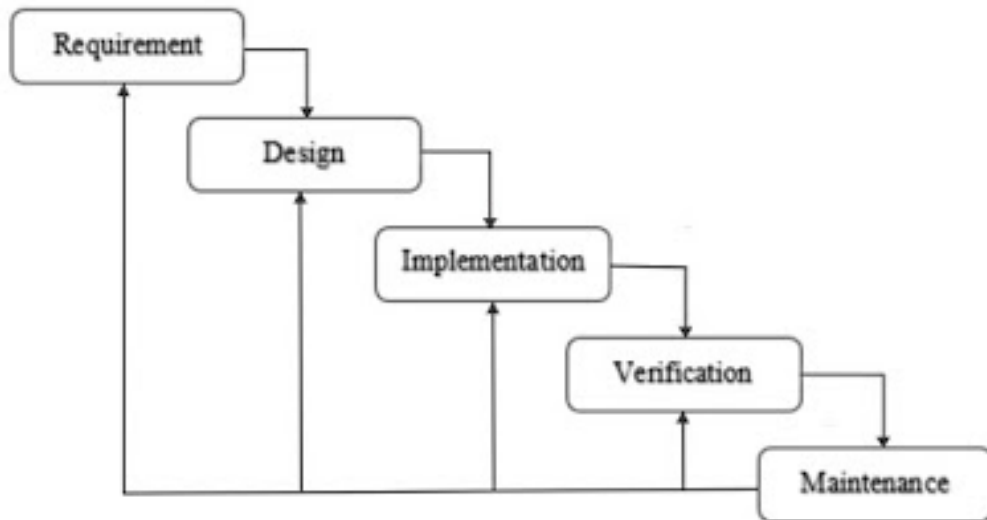
3.1.2. Bahan

Bahan yang digunakan dalam pembuatan aplikasi ini adalah data pertanyaan kuisisioner dari Proses TI PO1.

3.2. Alur Penelitian

Alur penelitian yang digunakan dalam pembuatan aplikasi ini adalah model SDLC (*Software Development Life Cycle*). Model SDLC dipilih sebagai model untuk melakukan proses pembuatan serta pemeliharaan sistem.

Metode SDLC yang digunakan untuk pembuatan sistem adalah model *Waterfall* atau *Classic Life Cycle*. Model ini adalah model di mana tahap pengembangan diharuskan menunggu tahap-tahap sebelumnya selesai untuk melanjutkan ke tahap-tahap berikutnya.



Gambar 3.1 *Waterfall Model* (“Metode Waterfall,” n.d.)

Berikut tahap-tahap pengembangan sistem berdasarkan metode *Waterfall*:

1. Perancangan arsitektur

Pada fase perancangan arsitektur ini merupakan fase awal dalam pembuatan sistem informasi yang mendefinisikan kebutuhan-kebutuhan sumber daya. Dalam fase ini dilakukan langkah sebagai berikut:

- Mendefinisikan masalah dan menentukan tujuan sistem.
- Mengidentifikasi berbagai kendala sistem dan membuat studi kelayakan.

2. Analisa kebutuhan

Proses pengumpulan kebutuhan dilakukan secara intensif untuk menspesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan pengguna.

3. Perancangan menggunakan UML

Perancangan menggunakan UML merupakan proses perancangan sistem seperti apa yang akan dibangun. Penelitian pada sistem yang berjalan dengan tujuan untuk merencanakan sistem yang baru menggunakan tools atau UML (*Unified Modelling Language*) dengan software visio 2016.

4. Pengembangan aplikasi

Fase pengembangan aplikasi merupakan fase mengimplementasikan design sistem pada fase-fase sebelumnya. Di dalam implementasi terdapat beberapa aktivitas yaitu:

- Pembuatan database sesuai skema rancangan
- Proses pembuatan aplikasi berdasarkan design sistem
- Proses debugging atau pengujian dan perbaikan suatu aplikasi

5. Pengujian

Pada tahap ini software hasil produksi harus diuji coba, termasuk semua fungsi-fungsinya, agar software sesuai dengan kebutuhan dan harus bebas dari error.

6. Implementasi

Pada tahap ini, software siap digunakan.

3.3. Rancangan Sistem dan Basis Data

Pada rancangan aplikasi digunakan *Unified Markup Language* (UML) sebagai bahasa perancang desain. Model yang dipakai dalam pada perancangan aplikasi adalah *Use Case Diagram*, *Activity Diagram*, *Entity Relationship Diagram*, dan *Class Diagram*.

3.3.1. Use Case Diagram

Rangkaian Use Case Diagram yang digunakan dalam pembuatan aplikasi dapat dilihat pada gambar 3.2.



Gambar 3.2 Use Case Diagram

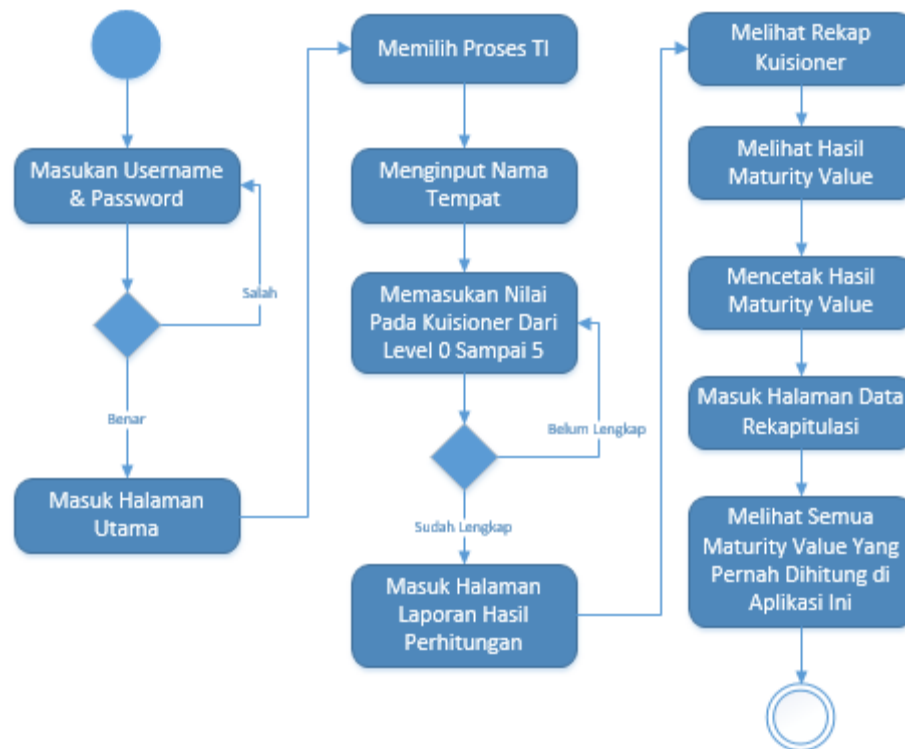
Berdasarkan analisis kebutuhan maka di dalam *use case* diagram terdapat satu aktor yaitu audit atau pengguna.

Berikut merupakan penjelasan dari *use case* diagram di atas

1. Pengguna harus registrasi dan login terlebih dahulu untuk dapat mengakses aplikasi
2. Pengguna dapat menambah pengguna
3. Pengguna dapat memasukan nilai pada tiap pertanyaan di dalam kuisisioner
4. Pengguna dapat melihat hasil perhitungan *Maturity Level*
5. Pengguna dapat mencetak hasil perhitungan *Maturity Level*
6. Pengguna dapat melihat semua data *Maturity Level* yang pernah dihitung di dalam aplikasi

3.3.2. Activity Diagram

Gambaran Activity Diagram yang digunakan dalam pembuatan aplikasi dapat dilihat pada gambar 3.3



Gambar 3.3 Activity Diagram

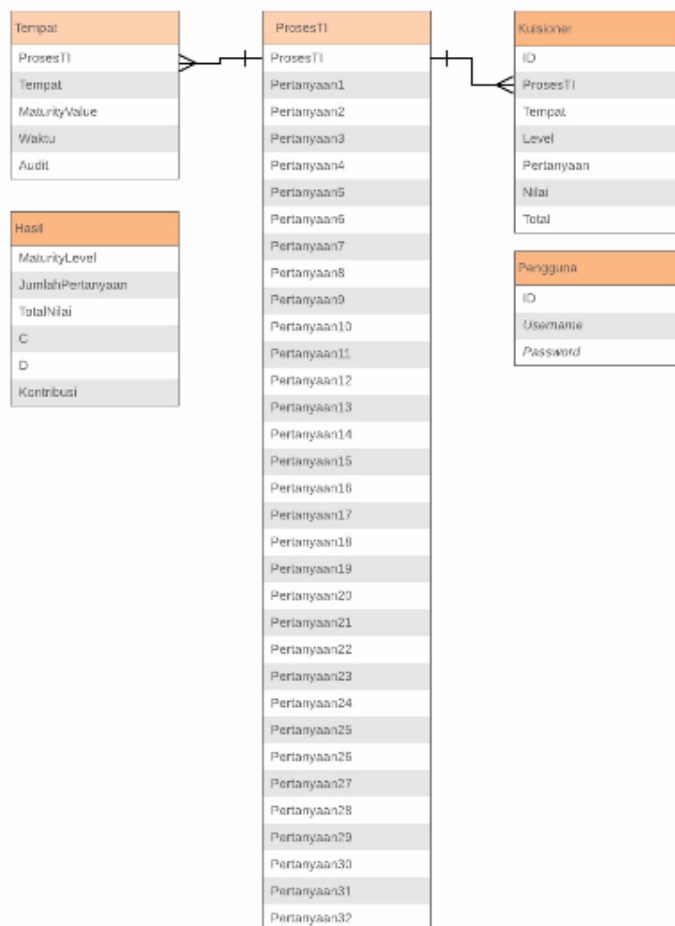
Activity Diagram pada gambar 3.3 menjelaskan mengenai pengaksesan ke dalam aplikasi, yaitu pengguna hanya dapat masuk ke dalam aplikasi ketika *username* dan *password* pengguna tersebut sudah terdaftar di dalam basis data.

Pengguna juga dapat menambahkan pengguna yang bisa mengakses ke dalam aplikasi. Pengguna masuk ke dalam halaman tambah pengguna dan memasukkan *username* dan *password* pengguna yang baru.

Pada halaman utama pengguna memilih proses TI, nama tempat atau instansi, lalu mengisi nilai pada setiap pertanyaan pada setiap level yaitu level 0 sampai 5. Apabila data telah terisi lengkap, maka dapat melanjutkan ke halaman yang berisi hasil dari kuisisioner perhitungan *maturity level* tersebut.

Setelah masuk pada halaman hasil kuisisioner yang berisi hasil perhitungan *maturity level*, pengguna dapat mencetak hasil perhitungan tersebut, dan pengguna juga dapat masuk ke dalam menu rekapitulasi data yang berisi semua data perhitungan *maturity level* yang pernah dihitung di dalam aplikasi ini. Tampilan data bisa diatur berdasarkan nama tempat atau instansi.

3.3.3. ER Diagram



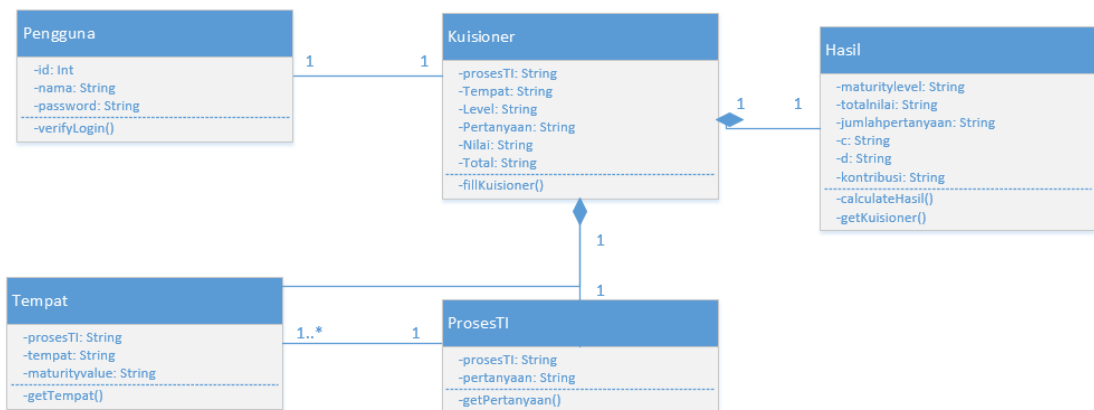
Gambar 3.4 ER Diagram

Berikut ini merupakan penjelasan dari ER Diagram pada gambar 3.4.

1. Entitas Pengguna tidak memiliki relasi dengan entitas manapun dan bersifat berdiri sendiri. Entitas Pengguna memiliki atribut antara lain ID, *Username* dan *Password*.

2. Entitas ProsesTI memiliki relasi dengan entitas Kuisisioner dan entitas Tempat, masing-masing memiliki relasi *one-to-many*. Entitas ProsesTI memiliki atribut ProsesTI dan atribut Pertanyaan yang berjumlah sebanyak pertanyaan yang diperlukan kuisisioner tergantung pada proses TI yang digunakan. Pada aplikasi ini jumlah pertanyaan ada 32 sebanyak pertanyaan dari prosesTI PO1.
3. Entitas Hasil tidak memiliki relasi dengan entitas manapun dan bersifat berdiri sendiri. Entitas hasil memiliki beberapa atribut antara lain MaturityLevel, JumlahPertanyaan, TotalNilai, C, D, dan Kontribusi.
4. Entitas Kuisisioner memiliki beberapa atribut antara lain ID, ProsesTI, Tempat, Level, Pertanyaan, Nilai, dan Total.
5. Entitas Tempat memiliki beberapa atribut antara lain ProsesTI, Tempat, MaturityValue, Waktu, Audit.

3.3.4. Class Diagram



Gambar 3.5 *Class Diagram*

Berikut merupakan penjelasan fungsi dari class diagram yang ada pada gambar 3.5

1. *Class* Pengguna

Pada *class* Pengguna terdapat `verifyLogin()` untuk mengecek apakah username dan password telah sesuai untuk masuk ke dalam halaman utama.

2. *Class* Kuisisioner

Pada *class* Kuisisioner terdapat fungsi `fillKuisisioner()` yang berfungsi untuk menambahkan data kuisisioner.

3. *Class* Hasil

Pada *class* Hasil terdapat fungsi `getHasil()` yang berfungsi untuk menampilkan hasil dari perhitungan *maturity value* yang telah dihitung berdasarkan kuisisioner dan fungsi `getKuisisioner()` yang berfungsi untuk menampilkan data kuisisioner yang telah diisi sebelumnya.

4. *Class* Proses TI

Pada *class* Proses TI terdapat fungsi `getPertanyaan()` yang berfungsi untuk menampilkan pertanyaan kuisisioner berdasarkan proses TI yang dipilih.

5. *Class* Tempat

Pada *class* Tempat terdapat fungsi `getTempat()` yang berfungsi untuk menampilkan data *maturity value* berdasarkan tempat yang dipilih.

3.4. Rancangan Antar Muka

Rancangan antar muka ini adalah bagian dari aplikasi yang menjadi wadah untuk melakukan komunikasi antara sistem dan pengguna dengan fungsi yang dapat memudahkan penggunaan aplikasi bagi pengguna.

3.4.1. Rancangan Antar Muka Halaman *Login*

Antar muka halaman *login* merupakan tampilan yang pertama kali saat aplikasi diakses. Pada halaman *login* pengguna harus memasukan nama pengguna dan *password* terlebih dahulu. Halaman ini memiliki dua *textbox* yaitu *textbox* nama pengguna dan *textbox* password. Halaman ini juga memiliki satu tombol yaitu tombol masuk, seperti yang terlihat pada gambar 3.6

The image shows a simple login interface within a window titled "Halaman Login". The window has a thin border. Inside, the text "Nama" is positioned to the left of a rectangular input box. Below it, the text "Password" is positioned to the left of another rectangular input box. Centered below these two input boxes is a rectangular button with the text "Masuk" inside it.

Gambar 3.6 Rancangan Antar Muka Halaman *Login*

3.4.2. Rancangan Antar Muka Halaman Utama

Antar muka halaman utama merupakan halaman pertama setelah pengguna berhasil masuk melewati halaman *login*. Pada halaman ini berisi pengisian kuisioner *maturity level* dari *level 0* sampai *level 5*. Pada halaman ini terdapat satu *combobox* yaitu *combobox* proses TI untuk memilih *maturity value* pada proses TI apa yang akan dihitung, serta satu *textbox* tempat untuk memasukan di mana tempat yang akan dihitung *maturity level* nya. Pada halaman ini terdapat *sub-menu* yang berisi daftar pertanyaan kuisioner dari *level 0* sampai *level 5*. Selain itu terdapat pula tiga tombol yaitu tombol tambah pengguna, tombol hitung dan tombol reset seperti pada gambar 3.7

Halaman Utama

Proses TI Tempat

0 1 2 3 4 5

1	<input type="text" value="Pertanyaan 1"/>	<input type="text" value="3"/> <input type="button" value="↑"/> <input type="button" value="↓"/>
2	<input type="text" value="Pertanyaan 2"/>	<input type="text" value="3"/> <input type="button" value="↑"/> <input type="button" value="↓"/>
3	<input type="text" value="Pertanyaan 3"/>	<input type="text" value="3"/> <input type="button" value="↑"/> <input type="button" value="↓"/>
4	<input type="text" value="Pertanyaan 4"/>	<input type="text" value="3"/> <input type="button" value="↑"/> <input type="button" value="↓"/>

Gambar 3.7 Rancangan Antar Muka Halaman Utama

3.4.3. Rancangan Antar Muka Halaman Hasil Kuisisioner

Antar muka halaman hasil kuisisioner merupakan tampilan yang muncul ketika tombol hitung pada halaman utama ditekan. Pada halaman hasil kuisisioner terdapat dua data *grid* yaitu data *grid* kuisisioner yang berisi daftar pertanyaan dan total nilai masing-masing level yang telah diisi pada halaman utama, dan data *grid* perhitungan yang berisi tentang perhitungan kuisisioner yang menghasilkan hasil akhir yaitu *maturity value* dari proses TI yang telah diisi pada halaman utama. Gambaran halaman hasil kuisisioner bisa dilihat pada gambar 3.8

Halaman Hasil Kuisisioner

Level

Proses TI	Level	Pertanyaan	Nilai

Maturity Level	Total Nilai	Jumlah Pertanyaan	C	D	Kontribusi

Maturity Value = ...

Gambar 3.8 Antar Muka Halaman Hasil Kuisisioner

3.4.4. Rancangan Antar Muka Halaman Rekapitulasi Data

Antar muka halaman rekapitulasi data merupakan tampilan yang berisi satu buah data *grid* yang menampilkan *maturity* dari setiap proses TI dan tempat yang pernah dihitung dalam aplikasi ini. Pada halaman rekapitulasi data ini juga mempunyai satu *combobox* yaitu *combobox* tempat yang berfungsi untuk mengubah susunan data pada data *grid view* berdasarkan tempat. Gambaran antar muka halaman rekapitulasi data bisa dilihat pada gambar 3.9

Halaman Hasil Kuisisioner

Tempat

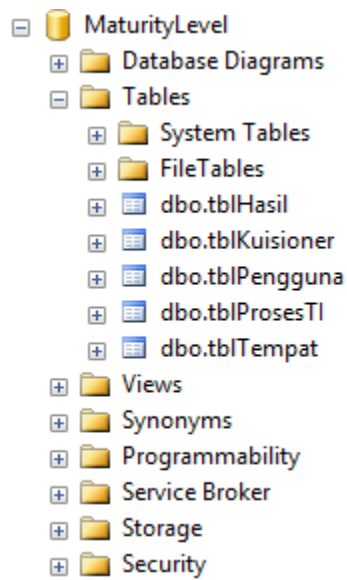
Proses TI	Tempat	Maturity Value

Gambar 3.9 Antar Muka Halaman Rekapitulasi Data

BAB IV HASIL DAN PEMBAHASAN

4.1. Database

Dalam pembuatan aplikasi diperlukan sebuah basis data yang digunakan sebagai sumber data untuk kemudian disimpan dalam sebuah server. Dalam aplikasi ini menggunakan Microsoft SQL Server 2014 sebagai basis data server. Pembuatan basis data merupakan hasil pengembangan dari ER Diagram yang dapat dilihat pada gambar 3.4. Hasil dan penjelasan mengenai pembuatan basis data “MaturityLevel” dapat dilihat pada gambar 4.1.



Gambar 4.1 Database Aplikasi

4.1.1. Definisi *Database*

Berikut ini merupakan penjelasan dari masing-masing tabel yang ada pada sistem aplikasi ini.

a. Tabel Pengguna

Tabel pengguna berisi data pengguna aplikasi; admin atau audit seperti nama dan password.

Tabel 4.1 Tabel Pengguna

Kolom	Tipe Data	Panjang Data	Keterangan
ID	Int	-	Primary Key
NAMA	Varchar	50	
PASSWORD	Varchar	50	

b. Tabel Kuisisioner

Tabel kuisisioner berisi data-data yang dibutuhkan untuk perhitungan maturity level seperti nama proses TI, nama tempat, level, pertanyaan-pertanyaan kuisisioner, nilai tiap pertanyaan, dan total nilai tiap level.

Tabel 4.2 Tabel Kuisisioner

Kolom	Tipe Data	Panjang Data	Keterangan
ID	Integer	-	Primary Key
PROSESTI	Varchar	5	Foreign Key
TEMPAT	Varchar	50	
LEVEL	Integer	-	
PERTANYAAN	Varchar	500	
NILAI	Float	-	
TOTAL	Float	-	

c. Tabel ProsesTI

Tabel prosesTI berisi semua pertanyaan dari masing-masing proses TI yang dibutuhkan dalam proses memasukan data kuisioner. Tabel proses TI memiliki

Tabel 4.3 Tabel ProsesTI

Kolom	Tipe Data	Panjang Data	Keterangan
ProsesTI	Varchar	5	Primary Key
Pertanyaan 1 - 32	Varchar	50	Jumlah Kolom Sebanyak Jumlah Pertanyaan

d. Tabel Hasil

Tabel hasil berisi tentang data perhitungan yang menghasilkan maturity value.

Tabel 4.4 Tabel Hasil

Kolom	Tipe Data	Panjang Data	Keterangan
Maturity Level	Float	-	
TotalNilai	Float	-	
JumlahPertanyaan	Integer	2	
C	Varchar	50	
D	Varchar	50	
Kontribusi	Varchar	50	

e. Tabel Tempat

Tabel tempat berisi tentang data maturity value yang pernah dihitung dari setiap tempat.

Tabel 4.5 Tabel Tempat

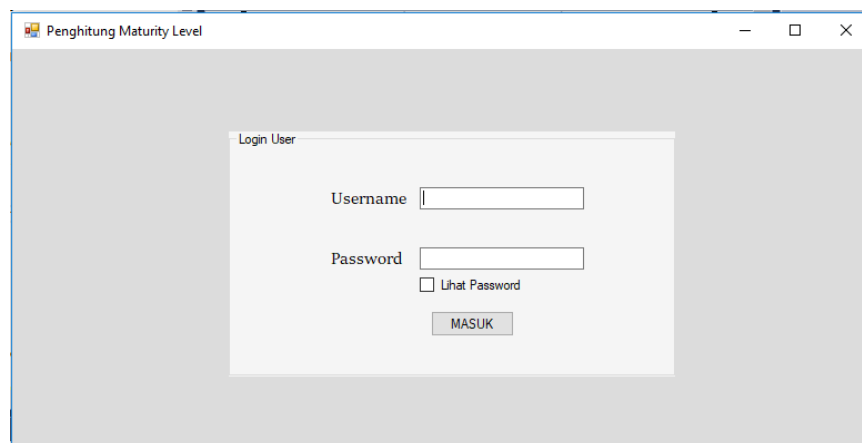
Kolom	Tipe Data	Panjang Data	Keterangan
ProsesTI	Varchar	5	Foreign Key
Tempat	Varchar	50	
MaturityValue	Float	-	
Waktu	Varchar	50	
Audit	Varchar	50	

4.2. Implementasi Antarmuka

Dalam pengembangan implementasi halaman antarmuka, rancangan antarmuka yang telah dibuat harus diubah kedalam bahasa pemrograman yaitu dalam aplikasi ini menggunakan bahasa *c#*. Perangkat yang digunakan sebagai pendukung untuk mengimplementasi rancangan antarmuka aplikasi penghitung *maturity level* ini adalah Microsoft Visual Studio 2015.

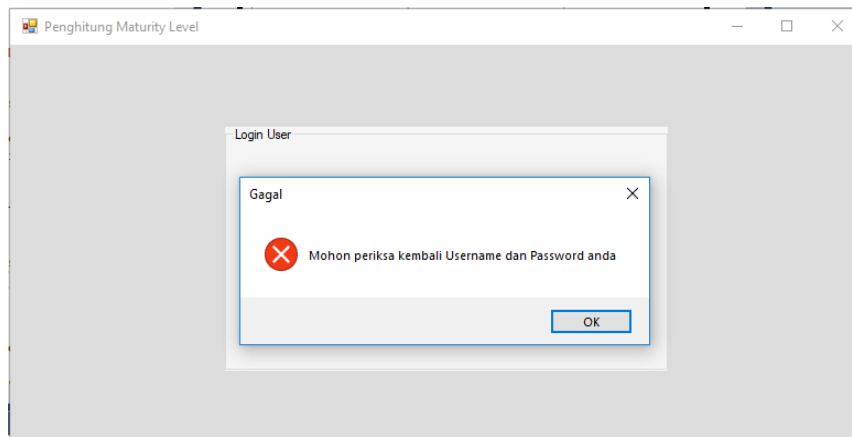
4.2.1. Antarmuka Halaman *Login*

Halaman *login* adalah halaman yang muncul pertama kali pada saat aplikasi mulai dijalankan. Fungsi dari halaman login adalah untuk memberikan keamanan ke dalam sistem. Sebagai validasi, admin atau audit diwajibkan mengisi textbox Username dan textbox Password dengan benar sebelum menekan tombol “MASUK”. Antarmuka halaman *login* dapat dilihat pada gambar 4.2.



Gambar 4.2 Halaman *Login*

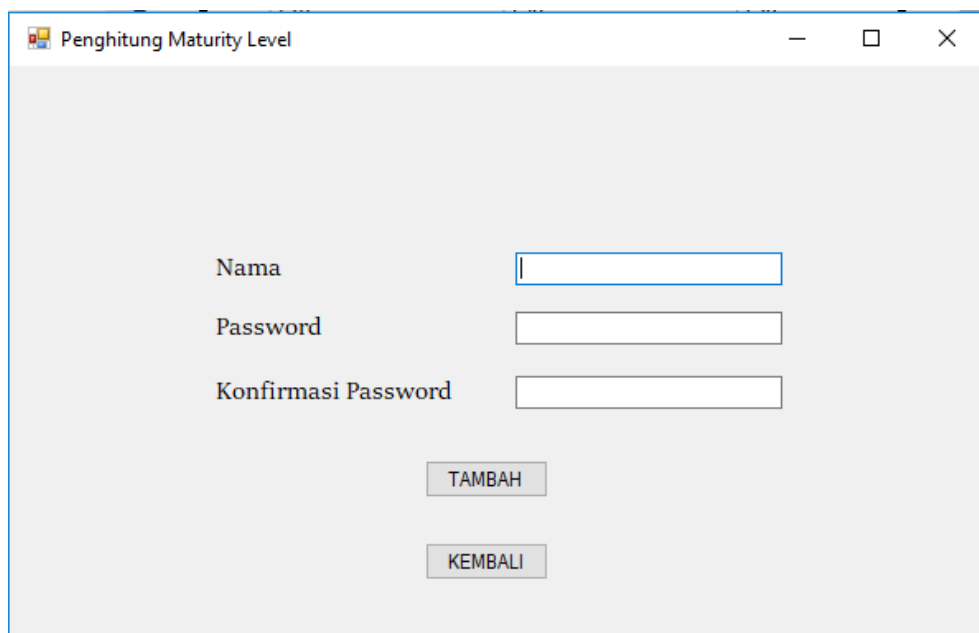
Apabila ketika tombol “MASUK” ditekan sedangkan textbox username dan textbox password masih kosong atau tidak sesuai dengan data pengguna dalam basis data yang telah diregistrasi, maka akan muncul peringatan seperti pada gambar 4.3.



Gambar 4.3 Peringatan *Login*

4.2.2. Antarmuka Halaman Tambah Pengguna

Halaman tambah pengguna merupakan halaman yang digunakan ketika akan menambah pengguna baru ke dalam basis data. Untuk masuk ke halaman tambah pengguna, admin harus masuk ke dalam halaman utama terlebih dahulu. Admin diharuskan memasukan nama, password, dan konfirmasi password dengan benar sebelum menekan tombol "TAMBAH". Antarmuka halaman tambah pengguna dapat dilihat pada gambar 4.4.



Gambar 4.4 Halaman Tambah Pengguna

4.2.3. Antarmuka Halaman Utama

Setelah pengguna berhasil memasukkan username dan password, maka pengguna masuk ke halaman utama. Pada halaman utama pengguna melakukan tahapan-tahapan yang akan dilewati antara lain

- Memilih proses TI pada combobox proses TI
- Memasukkan nama tempat pada textbox tempat
- Mengisi nilai pada combobox nilai dengan input “0”, “0,33”, “0,66”, “1” kolom kuisioner dari level
- Menekan tombol “LANJUT” apabila kuisioner sudah terisi dengan benar

Dalam halaman utama juga terpaat beberapa tombol seperti tombol “LANJUT”, tombol “RESET”, dan tombol “TAMBAH PENGGUNA”. Antarmuka halaman utama dapat dilihat pada gambar 4.5.

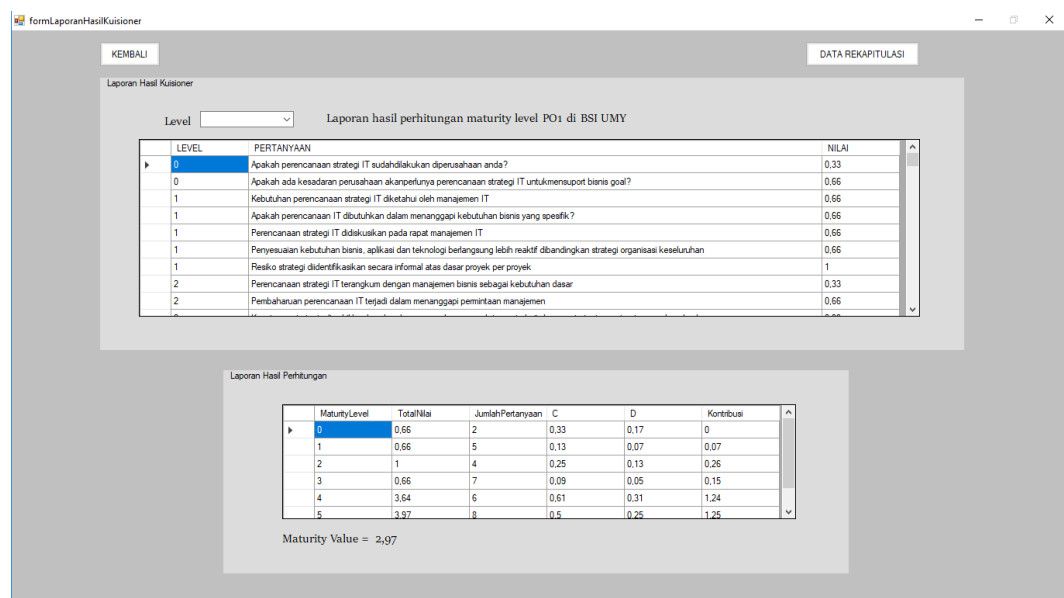
Level	Question	Value
1	Perencanaan strategi IT menggunakan acuan yang standar dan hal diluar dari itu akan diperhatikan oleh manajemen	1
2	Perencanaan strategi IT mendefinisikan fungsi manajemen dengan tanggung jawab tingkat senior	0,66
3	Manajemen dapat mengawasi proses perencanaan strategi IT serta menyampaikannya berdasarkan ukuranukuran yang jelas	0,66
4	Apakah perencanaan IT dalam jangka pendek maupun jangka panjang terjadi dan dijalankan dengan update yang akan dilakukan	0,33
5	Strategi IT dan strategi keseluruhan organisasi berkembang menjadi lebih terkoordinasi dengan menempatkan proses bisnis dan	0,66
6	Proses yang baik untuk menentukan penggunaan atas sumber daya internal dan eksternal yang dibutuhkan dalam pengembangan dan	0,33
TOTAL		3,64

Gambar 4.5 Halaman Utama

4.2.4. Antarmuka Halaman Hasil Kuisisioner

Halaman hasil kuisisioner merupakan halaman yang menampilkan hasil dari perhitungan nilai maturity yang sudah dihitung setelah pengguna sebelumnya memasukan nilai pada halaman utama.

Pada halaman hasil kuisisioner terdapat dua data grid view yaitu data grid kuisisioner berisi pertanyaan dan nilai yang telah dimasukan oleh pengguna pada halaman utama dan dapat ditampilkan berdasarkan level dengan memilih level 0 sampai 5 pada combobox level, sedangkan data grid ke dua berisi mengenai data perhitungan beserta hasil maturity value yang dihasilkan. Antarmuka halaman hasil kuisisioner dapat dilihat pada gambar 4.6.

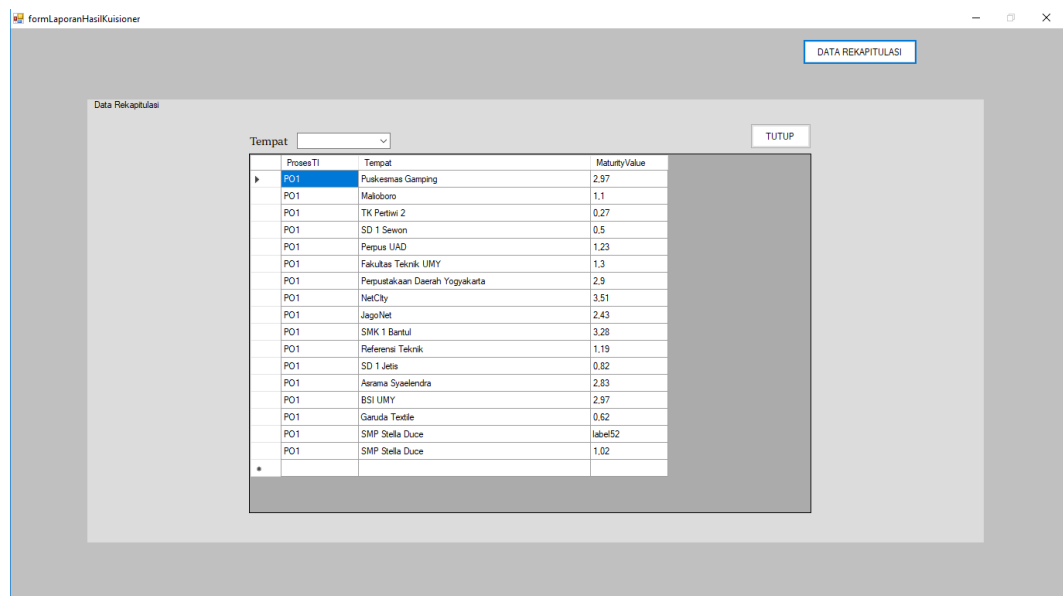


Gambar 4.6 Halaman Hasil Kuisisioner

4.2.5. Antarmuka Halaman Rekapitulasi Data

Halaman rekapitulasi data terdapat satu buah data grid view yang menampilkan data maturity level dari semua tempat yang pernah dinilai maturity level nya oleh aplikasi ini.

Data yang ditampilkan dapat dipilih sesuai tempat dengan memilih tempat yang telah masuk ke dalam basis data di dalam combobox tempat. Antarmuka halaman rekapitulasi data dapat dilihat pada gambar 4.7.



Gambar 4.7 Halaman Rekapitulasi Data

4.3. Implementasi Pengujian Sistem

Pengujian ini bertujuan untuk membuktikan bahwa setiap fungsi yang ada di dalam aplikasi telah berjalan dengan baik serta terbebas dari error. Proses pengujian meliputi tiga hal yaitu pengujian user *interface*, pengujian validasi, dan pengujian fungsi dasar sistem.

4.3.1. Pengujian User *Interface*

Pengujian fungsi *interface* memiliki tujuan untuk menguji fungsi-fungsi dari elemeng yang ada pada sistem aplikasi. Elemen-elemen yang diujikan dapat dilihat pada table 4.6.

Tabel 4.6 Hasil Pengujian

NO	Kasus Uji	Hasil	Status
1	Halaman <i>Login</i>	Sistem dapat masuk ke dalam aplikasi dan muncul halaman <i>login</i>	Berhasil
2	Tombol <i>Login</i>	Ketika tombol <i>login</i> diklik, maka sistem masuk ke dalam halaman utama	Berhasil
3	Halaman Utama	Sistem dapat masuk ke dalam halaman utama setelah <i>username</i> dan <i>password</i> terverifikasi dengan baik pada halaman <i>login</i>	Berhasil
4	<i>Combobox</i> Proses TI	Ketika <i>combobox</i> proses TI diakses, maka muncul pilihan proses TI yang terdaftar dalam basis data	Berhasil
5	<i>Groupbox</i> Kuisisioner	Pertanyaan - pertanyaan muncul dalam <i>groubox</i> kuisisioner sesuai dengan proses TI yang dipilih pada <i>combobox</i> proses TI	Berhasil
6	<i>Combobox</i> Nilai	<i>Combobox</i> nilai tiap pertanyaan muncul berisi nilai dari 0 – 0,33 – 0,66 - 1	Berhasil
7	Tombol Lanjut	Ketika tombol lanjut diklik, maka sistem akan masuk ke dalam halaman hasil kuisisioner	Berhasil
8	Halaman Hasil Kuisisioner	Pada halaman hasil kuisisioner terdapat dua data <i>grid view</i> yaitu data <i>grid</i> kuisisioner dan data <i>grid</i> hasil	Berhasil

9	Halaman Rekapitulasi Data	Pada halaman rekapitulasi data terdapat satu data <i>grid view</i> yaitu data <i>grid</i> tempat	
10	Halaman Tambah Pengguna	Pada halaman utama, jika tombol tambah pengguna diklik, maka sistem akan masuk ke dalam halaman tambah pengguna	Berhasil

4.3.2. Pengujian Validasi

Pengujian validasi memiliki tujuan agar fungsi-fungsi validasi setiap elemen di dalam sistem aplikasi berjalan dengan baik. Hasil pengujian dapat dilihat pada table 4.7

Table 4.7 Pengujian Validasi

NO	Kasus Uji	Hasil	Status
1	Validasi <i>Login</i>	Sistem akan memberikan peringatan ketika <i>username</i> dan <i>password</i> tidak sesuai dengan basis data	Berhasil
2	Validasi <i>Combobox</i> Proses TI	<i>Combobox</i> proses TI tidak dapat diisi secara manual	Berhasil
3	Validasi Tombol Lanjut	Sistem akan memberi peringatan ketika tombol lanjut diklik sedangkan <i>textbox</i> tempat belum terisi	Berhasil
4	Validasi Tambah Pengguna	Sistem akan memberi peringatan ketika pada halaman tambah pengguna ketika <i>textbox password</i> dan <i>textbox konfirmasi password</i> tidak sama	Berhasil
5	Validasi <i>Combobox</i> Nilai	<i>Combobox</i> nilai tidak dapat diisi secara manual	Berhasil

4.3.3. Pengujian Fungsi Dasar Sistem

Pengujian fungsi dasar sistem memiliki tujuan untuk memastikan fungsi-fungsi dasar dari sistem berjalan dengan baik. Hasil pengujian dapat dilihat pada table 4.8.

Tabel 4.8 Pengujian Fungsi Dasar Sistem

NO	Kasus Uji	Hasil	Status
1	Fungsi <i>Login</i>	Sistem dapat masuk ke dalam aplikasi dan muncul halaman <i>login</i> dan mengecek apakah <i>username</i> dan <i>password</i> yang dimasukkan sesuai dengan yang ada pada basis data	Berhasil
2	Fungsi <i>Groupbox</i> Kuisisioner	Pada halaman utama, pertanyaan pada kuisisioner sesuai dengan proses TI yang dipilih pada <i>combobox proses TI</i>	Berhasil
3	Fungsi <i>textbox</i> total	<i>Textbox</i> total sesuai dengan jumlah dari nilai yang diinputkan pada <i>combobox</i> nilai di setiap pertanyaan	Berhasil
4	Fungsi tombol lanjut	Ketika tombol lanjut diklik, data proses TI, tempat, pertanyaan, nilai, dan total akan ditambahkan ke dalam basis data	Berhasil
5	Fungsi <i>combobox</i> level	Pada halaman hasil kuisisioner, <i>combobox</i> level berfungsi untuk merubah tampilan data <i>grid view</i> sesuai dengan level yang dipilih	Berhasil
6	Fungsi Halaman Rekapitulasi Data	Data <i>grid view</i> pada halaman rekapitulasi data mengambil data dari basis data berisi proses TI, tempat, dan <i>maturity value</i>	Berhasil
7	Fungsi Tambah Pengguna	Pada halaman tambah pengguna, <i>username</i> dan <i>password</i> baru masuk ke dalam basis data	Berhasil

4.4. Pembahasan

Berdasarkan tiga pengujian sistem yang telah dilakukan, yaitu pengujian *user interface* yang dapat dilihat hasilnya pada tabel 4.6, lalu pengujian validasi yang dapat dilihat hasilnya pada tabel 4.7, dan pengujian fungsi dasar sistem yang dapat dilihat hasilnya pada tabel 4.8, hasil yang didapatkan sudah sesuai dengan hasil rancangan.

Sehingga dapat disimpulkan bahwa sistem aplikasi yang dibuat sudah memenuhi hal- hal yang diharapkan dan sesuai dengan rancangan karena telah berhasil melewati pengujian-pengujian yang dilakukan.

BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Berdasarkan hasil dan pembahasan yang telah dilakukan dalam pembuatan aplikasi penghitung *maturity level* menggunakan *framework* COBIT 4.1 maka diperoleh kesimpulan sebagai berikut:

1. Aplikasi penghitung *maturity level* menggunakan *framework* COBIT 4.1 telah berhasil dibuat.
2. Aplikasi dapat menghitung *maturity value* dengan baik dan benar sesuai dengan nilai yang dimasukkan oleh audit pada masing-masing pertanyaan di setiap level.
3. Aplikasi dapat menyimpan data-data perhitungan *maturity value* ke dalam basis data.

5.2. Saran

Berikut berupa saran yang dapat digunakan sebagai pengembangan sistem aplikasi pengukur *maturity level* menggunakan *framework* COBIT 4.1:

1. Menambah pertanyaan kuisisioner dari proses TI yang berbeda ke dalam aplikasi yang mana sementara hanya proses TI *Plan and Organize* 1 (PO1).
2. Menambah fitur di dalam aplikasi berupa print laporan berbentuk pdf.
3. Penyempurnaan sistem admin dimana diketahui waktu pada saat pengguna melakukan penilaian proses TI.
4. Menambah fitur ke dalam aplikasi berupa kendali data seperti edit dan delete data.
5. Apabila aplikasi ini akan dikembangkan secara *online*, maka dapat dikembangkan untuk penelitian selanjutnya.

DAFTAR PUSTAKA

- 4_Class Diagram.pdf. (n.d.). Retrieved from http://desy.lecturer.pens.ac.id/Workshop%20Pengembangan%20Perangkat%20Lunak/4_Class%20Diagram.pdf
- Booch, G. J., & Ivar, R. (n.d.). J. 2005. *The Unified Modeling Language User Guide Second Edition. United State: Addison Wesley Professional.*
- Brady, M., & Loonam, J. (2010). Exploring the use of entity-relationship diagramming as a technique to support grounded theory inquiry. *Qualitative Research in Organizations and Management: An International Journal*, 5(3), 224–237.
- Delano, L., & George, R. (2005). *Wrox's SQL Server 2005 Express Edition Starter Kit (Programmer to Programmer)*. Microsoft Press.
- Enterprise, J. (2016). *Belajar Sendiri Visual C# dan C++ untuk Pemula*. Elex Media Komputindo.
- ITGI, IGI. (2007). COBIT 4.1. *Framework Control Objective Management Guidelines Maturity Model*.
- ITGI, IT. (2007). Governance institute. *About IT Governance Framework: Information Systems Audit and Control Foundation-CobiT 4^o Edition. Rolling Meadows: ISACA.*
- Laksito, A. D., & Luthfi, E. T. (2013). PENGUKURAN TINGKAT MODEL KEMATANGAN PROSES COBIT MENGGUNAKAN APLIKASI BERBASIS WEB (Studi Kasus di STMIK AMIKOM Yogyakarta), 6.

Metode Waterfall : Definisi, Tahapan, Kelebihan dan Kekurangan. (n.d.). Retrieved March 14, 2019, from <http://www.pengetahuandanteknologi.com/2016/09/metode-waterfall-definisi-tahapan.html>

Simonsson, M. arten, & Johnson, P. (2006). Defining IT governance-a consolidation of literature. In *The 18th conference on advanced information systems engineering* (Vol. 6). Citeseer.

The class diagram. (2004, September 15). Retrieved March 14, 2019, from <http://www.ibm.com/developerworks/rational/library/content/RationalEdge/sep04/bell/index.html>

LAMPIRAN

FORM LOGIN

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.SqlClient;

namespace PENGHITUNG_MATURITY_LEVEL
{
    public partial class FormLogin : Form
    {
        public FormLogin()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            verifyLogin();
        }

        String ConString = "Data Source=DESKTOP-97NKBL3\\AULIARAKH;Initial
Catalog=MaturityLevel;Integrated Security=True";

        public bool verifyLogin()
        {
            bool hasil = false;
            try
            {
                SqlConnection con = new SqlConnection(ConString);
                SqlCommand cmd = new SqlCommand("Select PASSWORD from
tblPengguna where NAMA = '" + txtUsername.Text + "' ", con);
                con.Open();
                string pass = (string)cmd.ExecuteScalar();
                if (pass == txtPassword.Text)
                {
                    hasil = true;
                    //MessageBox.Show("Silahkan masuk, " +
txtUsername.Text, "Masuk", MessageBoxButtons.OK,
MessageBoxIcon.Information);
                    this.Hide();
                    formUtama next = new formUtama(txtUsername.Text);
                    next.Show();
                }
                else
                {
                    hasil = false;
                }
            }
            catch { }
        }
    }
}
```

```

        MessageBox.Show("Mohon periksa kembali Username dan
        Password anda", "Gagal", MessageBoxButtons.OK, MessageBoxIcon.Error);
        txtPassword.Text = "";
        txtPassword.Select();
    }
}
catch (Exception x)
{
    MessageBox.Show(x.Message);
}

return hasil;
}

e) private void checkPassword_CheckedChanged(object sender, EventArgs
{
    if (checkPassword.Checked == true)
    {
        txtPassword.UseSystemPasswordChar = false;
    }
    else if (checkPassword.Checked == false)
    {
        txtPassword.UseSystemPasswordChar = true;
    }
}

e) private void txtPassword_KeyPress(object sender, KeyPressEventArgs
{
    if (e.KeyChar != 13)
    {
    }
    else
    {
        verifyLogin();
    }
}
}
}
}

```


FORM HASIL

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.SqlClient;

namespace PENGHITUNG_MATURITY_LEVEL
{
    public partial class formLaporanHasilKuisisioner : Form
    {
        String lemparmaturity;
        String lempartempat;
        String lemparproses;
        String lemparwaktu;
        String lemparnama;
        String level0;
        String level1;
        String level2;
        String level3;
        String level4;
        String level5;
        String ConString = "Data Source=DESKTOP-97NKBL3\\AULIARAKH;Initial
Catalog=MaturityLevel;Integrated Security=True";

        public formLaporanHasilKuisisioner(string maturity, string tempat,
string proses, string waktu,string nama,string level0,string level1,string
level2, string level3,string level4,string level5)
        {
            InitializeComponent();
            this.lemparmaturity = maturity;
            lblMaturity.Text = "Maturity Value = "+lemparmaturity;

            this.lempartempat = tempat;
            lbltempat.Text = lempartempat;

            this.lemparproses = proses;
            lblProses.Text = lemparproses;

            this.lemparwaktu = waktu;
            this.lemparnama = nama;
            this.level0 = level0;
            this.level1 = level1;
            this.level2 = level2;
            this.level3 = level3;
            this.level4 = level4;
            this.level5 = level5;
        }

        void getKuisisioner()
        {
```

```

this.tblTempatTableAdapter1.Fill(this.maturityLevelDataSet3.tblTempat);
this.tblHasilTableAdapter.Fill(this.maturityLevelDataSet1.tblHasil);
this.tblKuisisionerTableAdapter1.Fill(this.maturityLevelDataSet2.tblKuisisioner
);
    this.tblTempatTableAdapter2.Fill(this.dataSetTempat.tblTempat);
}
private void formLaporanHasilKuisisioner_Load(object sender,
EventArgs e)
{
    getKuisisioner();
}

private void txtLv1_TextChanged(object sender, EventArgs e)
{
    try
    {
        SqlConnection con = new SqlConnection(ConString);
        SqlCommand cmd = new SqlCommand("Select * from
dbo.tblKuisisioner where Tempat = '" + lbltempat.Text + "' AND Lv1 like('%" +
comboLv1.Text + "%')", con);
        SqlDataAdapter da = new SqlDataAdapter(cmd);
        DataTable dt = new DataTable();
        da.Fill(dt);
        gridHasilKuisisioner.DataSource = dt;
        con.Open();
        cmd.ExecuteNonQuery();
        con.Close();
    }
    catch (Exception x)
    {
        MessageBox.Show(x.Message);
    }
}

private void comboLv1_KeyPress_1(object sender, KeyPressEventArgs
e)
{
    if (e.KeyChar != 13)
    {
        MessageBox.Show("Pilihlah 'Level' dari daftar yang telah
disediakan!", "Peringatan", MessageBoxButtons.OK, MessageBoxIcon.Warning);
        e.KeyChar = (char)0;
    }
}

private void btnReset_Click(object sender, EventArgs e)
{
    formUtama back = new formUtama(lemparnama.ToString());
    this.Hide();
    back.Show();
}

void getTempat()
{

```

```

        groupBox3.Visible = true;
        groupBox1.Visible = false;
        groupBox2.Visible = false;
        btnKembali.Visible = false;

        try
        {
            SqlConnection con = new SqlConnection(ConString);
            SqlCommand cmd = new SqlCommand("Select * from
dbo.tblTempat", con);
            con.Open();
            SqlDataReader dr = cmd.ExecuteReader();
            comboTempat.Items.Clear();
            while (dr.Read())
            {
                string namatempat = dr.GetString(1);
                comboTempat.Items.Add(namatempat);
            }
            con.Close();
        }
        catch (Exception x)
        {
            MessageBox.Show(x.Message);
        }

        try
        {
            SqlConnection con1 = new SqlConnection(ConString);
            SqlCommand cmd1 = new SqlCommand("Select * from
dbo.tblTempat ORDER BY Tempat ASC", con1);
            SqlDataAdapter da1 = new SqlDataAdapter(cmd1);
            DataTable dt1 = new DataTable();
            da1.Fill(dt1);
            gridRekapitulasi.DataSource = dt1;
            con1.Open();
            cmd1.ExecuteNonQuery();
            con1.Close();
        }
        catch (Exception x1)
        {
            MessageBox.Show(x1.Message);
        }
    }

    private void button2_Click(object sender, EventArgs e)
    {
        getTempat();
    }

    private void button1_Click(object sender, EventArgs e)
    {
        groupBox3.Visible = false;
        groupBox1.Visible = true;
        groupBox2.Visible = true;
        btnKembali.Visible = true;
    }
}

```

```

private void comboBox2_SelectedIndexChanged(object sender,
EventArgs e)
{
    if(comboTempat.Text != "")
    {
        comboLv1.Enabled = true;
    }
    else
    {
        comboLv1.Enabled = false;
    }

    try
    {
        SqlConnection con = new SqlConnection(ConString);
        SqlCommand cmd = new SqlCommand("Select * from
dbo.tblTempat where Tempat = '" + comboTempat.Text + "'", con);
        SqlDataAdapter da = new SqlDataAdapter(cmd);
        DataTable dt = new DataTable();
        da.Fill(dt);
        gridRekapitulasi.DataSource = dt;
        con.Open();
        cmd.ExecuteNonQuery();
        con.Close();
    }
    catch (Exception x)
    {
        MessageBox.Show(x.Message);
    }
}

private void btnLihatSemua_Click(object sender, EventArgs e)
{
    comboTempat.Text="";
    try
    {
        SqlConnection con1 = new SqlConnection(ConString);
        SqlCommand cmd1 = new SqlCommand("Select * from
dbo.tblTempat ORDER BY Tempat ASC", con1);
        SqlDataAdapter da1 = new SqlDataAdapter(cmd1);
        DataTable dt1 = new DataTable();
        da1.Fill(dt1);
        gridRekapitulasi.DataSource = dt1;
        con1.Open();
        cmd1.ExecuteNonQuery();
        con1.Close();
    }
    catch (Exception x1)
    {
        MessageBox.Show(x1.Message);
    }
}

private void printDocument1_PrintPage(object sender,
System.Drawing.Printing.PrintPageEventArgs e)
{

```

```

        e.Graphics.DrawString("HASIL PENGUKURAN MATURITY LEVEL", new
Font("Arial", 26, FontStyle.Regular), Brushes.Black, new Point(100, 100));
        //e.Graphics.DrawImage bmp, 0, 0);
        e.Graphics.DrawString("Proses TI = "+lblProses.Text, new
Font("Arial", 22, FontStyle.Regular), Brushes.Black, new Point(100, 200));
        e.Graphics.DrawString("Tempat = " + lbltempat.Text, new
Font("Arial", 22, FontStyle.Regular), Brushes.Black, new Point(100, 240));
        e.Graphics.DrawString("Nilai Level 0 = " + level0.ToString(),
new Font("Arial", 22, FontStyle.Regular), Brushes.Black, new Point(100,
280));
        e.Graphics.DrawString("Nilai Level 1 = " + level1.ToString(),
new Font("Arial", 22, FontStyle.Regular), Brushes.Black, new Point(100,
320));
        e.Graphics.DrawString("Nilai Level 2 = " + level2.ToString(),
new Font("Arial", 22, FontStyle.Regular), Brushes.Black, new Point(100,
360));
        e.Graphics.DrawString("Nilai Level 3 = " + level3.ToString(),
new Font("Arial", 22, FontStyle.Regular), Brushes.Black, new Point(100,
400));
        e.Graphics.DrawString("Nilai Level 4 = " + level4.ToString(),
new Font("Arial", 22, FontStyle.Regular), Brushes.Black, new Point(100,
440));
        e.Graphics.DrawString("Nilai Level 5 = " + level5.ToString(),
new Font("Arial", 22, FontStyle.Regular), Brushes.Black, new Point(100,
480));
        e.Graphics.DrawString(lblMaturity.Text, new Font("Arial", 22,
FontStyle.Regular), Brushes.Black, new Point(100, 520));
        e.Graphics.DrawString("Waktu = " + lemparwaktu.ToString(), new
Font("Arial", 22, FontStyle.Regular), Brushes.Black, new Point(100, 560));
        e.Graphics.DrawString("Audit = " + lemparnama.ToString(), new
Font("Arial", 22, FontStyle.Regular), Brushes.Black, new Point(100, 600));
        e.Graphics.DrawString("_____ ", new Font("Arial",
22, FontStyle.Regular), Brushes.Black, new Point(100, 680));
    }
    //Bitmap bmp;
    private void btnCetak_Click(object sender, EventArgs e)
    {
        //Graphics g = this.CreateGraphics();
        //bmp = new Bitmap(this.Size.Width, this.Size.Height, g);
        //Graphics mg = Graphics.FromImage(bmp);

        //mg.CopyFromScreen(this.Location.X, this.Location.Y, 0, 0, this.Size);
        //
        printPreviewDialog1.ShowDialog();
        // using (MaturityLevelDataSet2 db = new
MaturityLevelDataSet2())
        // {
            //tblKuisisionerBindingSource1.DataSource =
db.tblKuisisioner.ToList();
        //}
    }
}

```

FORM TAMBAH PENGGUNA

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.SqlClient;

namespace PENGHITUNG_MATURITY_LEVEL
{
    public partial class FormTambahPengguna : Form
    {
        public FormTambahPengguna()
        {
            InitializeComponent();

            String ConString = "Data Source=DESKTOP-97NKBL3\\AULIARAKH;Initial
Catalog=MaturityLevel;Integrated Security=True";

            private void btnTambah_Click(object sender, EventArgs e)
            {
                if (txtKonfirmasi.Text != txtPassword.Text)
                {
                    MessageBox.Show("Password tidak sama", "Peringatan",
                    MessageBoxButtons.OK, MessageBoxIcon.Error);
                    txtPassword.Text = "";
                    txtKonfirmasi.Text = "";
                    txtKonfirmasi.Enabled = false;
                    txtPassword.Select();
                }
                else
                {
                    try
                    {
                        SqlConnection con = new SqlConnection(ConString);
                        con.Open();
                        SqlCommand cmd = new SqlCommand("insert into
                        dbo.tblUser values('" + txtNama.Text + "',''+ txtPassword.Text + ')", con);
                        cmd.ExecuteNonQuery();
                        con.Close();
                        MessageBox.Show("Berhasil menambah admin", "Berhasil",
                        MessageBoxButtons.OK, MessageBoxIcon.Information);
                        txtNama.Text = "";
                        txtPassword.Text = "";
                        txtPassword.Enabled = false;
                        txtKonfirmasi.Text = "";
                        txtKonfirmasi.Enabled = false;
                        txtNama.Select();
                    }
                    catch (Exception x)
                    {

```

```
        MessageBox.Show("Gagal menambah admin", "Gagal",  
        MessageBoxButtons.OK, MessageBoxIcon.Error);  
        MessageBox.Show(x.Message);  
    }  
}  
  
private void btnKembali_Click(object sender, EventArgs e)  
{  
    formUtama back = new formUtama();  
    this.Hide();  
    back.Show();  
}  
}
```

FORM UTAMA

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.SqlClient;

namespace PENGHITUNG_MATURITY_LEVEL
{
    public partial class formUtama : Form
    {
        String lemparnama;

        String ConString = "Data Source=DESKTOP-97NKBL3\\AULIARAKH;Initial
Catalog=MaturityLevel;Integrated Security=True";

        public formUtama(string nama)
        {
            InitializeComponent();
            this.lemparnama = nama;
            lblNama.Text = lemparnama;
        }

        //masuk ke tambah pengguna
        private void btnTambah_Click(object sender, EventArgs e)
        {
            this.Hide();
            FormTambahPengguna next = new FormTambahPengguna();
            next.Show();
        }

        void getPertanyaan()
        {
            lblProses.Text = comboProsesTI.Text;

            if (comboProsesTI.Text == "PO1")
            {
                txtTempat.Enabled = true;
                SqlConnection con = new SqlConnection(ConString);
                con.Open();
                SqlCommand cmd = new SqlCommand("SELECT * FROM tblProsesTI
WHERE PROSESTI = 'PO1'", con);
                SqlDataReader dr = cmd.ExecuteReader();
                if(dr.Read())
                {
                    Pertanyaan1Lv10.Text = (dr.GetValue(1).ToString());
                    Pertanyaan2Lv10.Text = (dr.GetValue(2).ToString());

                    Pertanyaan1Lv11.Text = (dr.GetValue(3).ToString());
                    Pertanyaan2Lv11.Text = (dr.GetValue(4).ToString());
                    Pertanyaan3Lv11.Text = (dr.GetValue(5).ToString());
                }
            }
        }
    }
}
```



```

        Pertanyaan4Lv11.Text = (dr.GetValue(6).ToString());
        Pertanyaan5Lv11.Text = (dr.GetValue(7).ToString());

        Pertanyaan1Lv12.Text = (dr.GetValue(8).ToString());
        Pertanyaan2Lv12.Text = (dr.GetValue(9).ToString());
        Pertanyaan3Lv12.Text = (dr.GetValue(10).ToString());
        Pertanyaan4Lv12.Text = (dr.GetValue(11).ToString());

        Pertanyaan1Lv13.Text = (dr.GetValue(12).ToString());
        Pertanyaan2Lv13.Text = (dr.GetValue(13).ToString());
        Pertanyaan3Lv13.Text = (dr.GetValue(14).ToString());
        Pertanyaan4Lv13.Text = (dr.GetValue(15).ToString());
        Pertanyaan5Lv13.Text = (dr.GetValue(16).ToString());
        Pertanyaan6Lv13.Text = (dr.GetValue(17).ToString());
        Pertanyaan7Lv13.Text = (dr.GetValue(18).ToString());

        Pertanyaan1Lv14.Text = (dr.GetValue(19).ToString());
        Pertanyaan2Lv14.Text = (dr.GetValue(20).ToString());
        Pertanyaan3Lv14.Text = (dr.GetValue(21).ToString());
        Pertanyaan4Lv14.Text = (dr.GetValue(22).ToString());
        Pertanyaan5Lv14.Text = (dr.GetValue(23).ToString());
        Pertanyaan6Lv14.Text = (dr.GetValue(24).ToString());

        Pertanyaan1Lv15.Text = (dr.GetValue(25).ToString());
        Pertanyaan2Lv15.Text = (dr.GetValue(26).ToString());
        Pertanyaan3Lv15.Text = (dr.GetValue(27).ToString());
        Pertanyaan4Lv15.Text = (dr.GetValue(28).ToString());
        Pertanyaan5Lv15.Text = (dr.GetValue(29).ToString());
        Pertanyaan6Lv15.Text = (dr.GetValue(30).ToString());
        Pertanyaan7Lv15.Text = (dr.GetValue(31).ToString());
        Pertanyaan8Lv15.Text = (dr.GetValue(32).ToString());
    }
    cmd.Dispose();
    con.Close();
}

if (comboProsesTI.Text == "")
{
    RefreshForm();
    Pertanyaan1Lv10.Text = "";
    Pertanyaan2Lv10.Text = "";

    Pertanyaan1Lv11.Text = "";
    Pertanyaan2Lv11.Text = "";
    Pertanyaan3Lv11.Text = "";
    Pertanyaan4Lv11.Text = "";
    Pertanyaan5Lv11.Text = "";

    Pertanyaan1Lv12.Text = "";
    Pertanyaan2Lv12.Text = "";
    Pertanyaan3Lv12.Text = "";
    Pertanyaan4Lv12.Text = "";

    Pertanyaan1Lv13.Text = "";
    Pertanyaan2Lv13.Text = "";
    Pertanyaan3Lv13.Text = "";
    Pertanyaan4Lv13.Text = "";
}

```

```

        Pertanyaan5Lv13.Text = "";
        Pertanyaan6Lv13.Text = "";
        Pertanyaan7Lv13.Text = "";

        Pertanyaan1Lv14.Text = "";
        Pertanyaan2Lv14.Text = "";
        Pertanyaan3Lv14.Text = "";
        Pertanyaan4Lv14.Text = "";
        Pertanyaan5Lv14.Text = "";
        Pertanyaan6Lv14.Text = "";

        Pertanyaan1Lv15.Text = "";
        Pertanyaan2Lv15.Text = "";
        Pertanyaan3Lv15.Text = "";
        Pertanyaan4Lv15.Text = "";
        Pertanyaan5Lv15.Text = "";
        Pertanyaan6Lv15.Text = "";
        Pertanyaan7Lv15.Text = "";
        Pertanyaan8Lv15.Text = "";
    }
}

//jika combo proses TI diubah
private void comboProsesTI_SelectedIndexChanged(object sender,
EventArgs e)
{
    getPertanyaan();
}

//validasi combo proses TI
private void comboProsesTI_KeyPress(object sender,
KeyPressEventArgs e)
{
    if (e.KeyChar != 13)
    {
        MessageBox.Show("Pilihlah 'Proses TI' dari daftar yang
telah disediakan!", "Peringatan", MessageBoxButtons.OK,
MessageBoxIcon.Warning);
        e.KeyChar = (char)0;
    }

    if (comboProsesTI.Text == "")
    {
        groupBox1.Enabled = false;
    }
    else
    {
        groupBox1.Enabled = true;
    }
}

private void comboProsesTI_TextChanged(object sender, EventArgs e)
{
    if (comboProsesTI.Text == "")
    {
        groupBox1.Enabled = false;
    }
}

```

```

        groupBox1.Enabled = false;
        label6.Visible = false;
        lblLevel.Visible = false;
        label8.Visible = false;
        lblProses.Visible = false;
    }
    else
    {
        groupBox1.Enabled = true;
    }
}

//lanjut ke perhitungan

void fillKuisisioner()
{
    if (lblTotalLv10.Text == "TOTAL" || lblTotalLv11.Text ==
"TOTAL" || lblTotalLv12.Text == "TOTAL" ||
        lblTotalLv13.Text == "TOTAL" || lblTotalLv14.Text ==
"TOTAL" || lblTotalLv15.Text == "TOTAL" || txtTempat.Text == "")
    {
        MessageBox.Show("Pastikan kuisisioner sudah terisi
lengkap!", "Peringatan", MessageBoxButtons.OK, MessageBoxIcon.Warning);
    }
    else
    {
        MessageBox.Show("Sudah
lengkap", "Information", MessageBoxButtons.OK, MessageBoxIcon.Information);

        //memasukan hasil nilai-nilai kuisisioner dan pertanyaannya
        try
        {
            SqlConnection con = new SqlConnection(ConString);
            con.Open();

            //lv10
            SqlCommand cmd = new SqlCommand("Insert into
dbo.tblKuisisioner values('"+ comboProsesTI.Text + "','"+ txtTempat.Text
+ "','0','"+ Pertanyaan1Lv10.Text + "','"+ comboNilai1Lv10.Text + "','"+
lblTotalLv10.Text + "')", con);
            cmd.ExecuteNonQuery();
            SqlCommand cmd1 = new SqlCommand("Insert into
dbo.tblKuisisioner values('" + comboProsesTI.Text + "',' + txtTempat.Text +
 "','0','" + Pertanyaan2Lv10.Text + "',' + comboNilai2Lv10.Text + "',' +
lblTotalLv10.Text + "')", con);
            cmd1.ExecuteNonQuery();

            //lv11
            SqlCommand cmd3 = new SqlCommand("Insert into
dbo.tblKuisisioner values('" + comboProsesTI.Text + "',' + txtTempat.Text +
 "','1','" + Pertanyaan1Lv11.Text + "',' + comboNilai1Lv11.Text + "',' +
lblTotalLv11.Text + "')", con);
            cmd3.ExecuteNonQuery();
            SqlCommand cmd4 = new SqlCommand("Insert into
dbo.tblKuisisioner values('" + comboProsesTI.Text + "',' + txtTempat.Text +

```

```

'', '1', '' + Pertanyaan2Lv11.Text + ', ' + comboNilai2Lv11.Text + ', ' +
lblTotalLv11.Text + ''", con);
        cmd4.ExecuteNonQuery();
        SqlCommand cmd5 = new SqlCommand("Insert into
dbo.tblKuisisioner values('" + comboProsesTI.Text + ', ' + txtTempat.Text +
'', '1', '' + Pertanyaan3Lv11.Text + ', ' + comboNilai3Lv11.Text + ', ' +
lblTotalLv11.Text + ''")", con);
        cmd5.ExecuteNonQuery();
        SqlCommand cmd6 = new SqlCommand("Insert into
dbo.tblKuisisioner values('" + comboProsesTI.Text + ', ' + txtTempat.Text +
'', '1', '' + Pertanyaan4Lv11.Text + ', ' + comboNilai4Lv11.Text + ', ' +
lblTotalLv11.Text + ''")", con);
        cmd6.ExecuteNonQuery();
        SqlCommand cmd7 = new SqlCommand("Insert into
dbo.tblKuisisioner values('" + comboProsesTI.Text + ', ' + txtTempat.Text +
'', '1', '' + Pertanyaan5Lv11.Text + ', ' + comboNilai5Lv11.Text + ', ' +
lblTotalLv11.Text + ''")", con);
        cmd7.ExecuteNonQuery();

        //lv12
        SqlCommand cmd8 = new SqlCommand("Insert into
dbo.tblKuisisioner values('" + comboProsesTI.Text + ', ' + txtTempat.Text +
'', '2', '' + Pertanyaan1Lv12.Text + ', ' + comboNilai1Lv12.Text + ', ' +
lblTotalLv12.Text + ''")", con);
        cmd8.ExecuteNonQuery();
        SqlCommand cmd9 = new SqlCommand("Insert into
dbo.tblKuisisioner values('" + comboProsesTI.Text + ', ' + txtTempat.Text +
'', '2', '' + Pertanyaan2Lv12.Text + ', ' + comboNilai2Lv12.Text + ', ' +
lblTotalLv12.Text + ''")", con);
        cmd9.ExecuteNonQuery();
        SqlCommand cmd10 = new SqlCommand("Insert into
dbo.tblKuisisioner values('" + comboProsesTI.Text + ', ' + txtTempat.Text +
'', '2', '' + Pertanyaan3Lv12.Text + ', ' + comboNilai3Lv12.Text + ', ' +
lblTotalLv12.Text + ''")", con);
        cmd10.ExecuteNonQuery();
        SqlCommand cmd11 = new SqlCommand("Insert into
dbo.tblKuisisioner values('" + comboProsesTI.Text + ', ' + txtTempat.Text +
'', '2', '' + Pertanyaan4Lv12.Text + ', ' + comboNilai4Lv12.Text + ', ' +
lblTotalLv12.Text + ''")", con);
        cmd11.ExecuteNonQuery();

        //lv13
        SqlCommand cmd12 = new SqlCommand("Insert into
dbo.tblKuisisioner values('" + comboProsesTI.Text + ', ' + txtTempat.Text +
'', '3', '' + Pertanyaan1Lv13.Text + ', ' + comboNilai1Lv13.Text + ', ' +
lblTotalLv13.Text + ''")", con);
        cmd12.ExecuteNonQuery();
        SqlCommand cmd13 = new SqlCommand("Insert into
dbo.tblKuisisioner values('" + comboProsesTI.Text + ', ' + txtTempat.Text +
'', '3', '' + Pertanyaan2Lv13.Text + ', ' + comboNilai2Lv13.Text + ', ' +
lblTotalLv13.Text + ''")", con);
        cmd13.ExecuteNonQuery();
        SqlCommand cmd14 = new SqlCommand("Insert into
dbo.tblKuisisioner values('" + comboProsesTI.Text + ', ' + txtTempat.Text +
'', '3', '' + Pertanyaan3Lv13.Text + ', ' + comboNilai3Lv13.Text + ', ' +
lblTotalLv13.Text + ''")", con);
        cmd14.ExecuteNonQuery();

```

```

        SqlCommand cmd15 = new SqlCommand("Insert into
dbo.tblKuisisioner values('" + comboProsesTI.Text + "',''" + txtTempat.Text +
 "','3','" + Pertanyaan4Lv13.Text + "',''" + comboNilai4Lv13.Text + "',''" +
 lblTotalLv13.Text + "')", con);
        cmd15.ExecuteNonQuery();
        SqlCommand cmd16 = new SqlCommand("Insert into
dbo.tblKuisisioner values('" + comboProsesTI.Text + "',''" + txtTempat.Text +
 "','3','" + Pertanyaan5Lv13.Text + "',''" + comboNilai5Lv13.Text + "',''" +
 lblTotalLv13.Text + "')", con);
        cmd16.ExecuteNonQuery();
        SqlCommand cmd17 = new SqlCommand("Insert into
dbo.tblKuisisioner values('" + comboProsesTI.Text + "',''" + txtTempat.Text +
 "','3','" + Pertanyaan6Lv13.Text + "',''" + comboNilai6Lv13.Text + "',''" +
 lblTotalLv13.Text + "')", con);
        cmd17.ExecuteNonQuery();
        SqlCommand cmd18 = new SqlCommand("Insert into
dbo.tblKuisisioner values('" + comboProsesTI.Text + "',''" + txtTempat.Text +
 "','3','" + Pertanyaan7Lv13.Text + "',''" + comboNilai7Lv13.Text + "',''" +
 lblTotalLv13.Text + "')", con);
        cmd18.ExecuteNonQuery();

        //lv14
        SqlCommand cmd19 = new SqlCommand("Insert into
dbo.tblKuisisioner values('" + comboProsesTI.Text + "',''" + txtTempat.Text +
 "','4','" + Pertanyaan1Lv14.Text + "',''" + comboNilai1Lv14.Text + "',''" +
 lblTotalLv14.Text + "')", con);
        cmd19.ExecuteNonQuery();
        SqlCommand cmd20 = new SqlCommand("Insert into
dbo.tblKuisisioner values('" + comboProsesTI.Text + "',''" + txtTempat.Text +
 "','4','" + Pertanyaan2Lv14.Text + "',''" + comboNilai2Lv14.Text + "',''" +
 lblTotalLv14.Text + "')", con);
        cmd20.ExecuteNonQuery();
        SqlCommand cmd21 = new SqlCommand("Insert into
dbo.tblKuisisioner values('" + comboProsesTI.Text + "',''" + txtTempat.Text +
 "','4','" + Pertanyaan3Lv14.Text + "',''" + comboNilai3Lv14.Text + "',''" +
 lblTotalLv14.Text + "')", con);
        cmd21.ExecuteNonQuery();
        SqlCommand cmd22 = new SqlCommand("Insert into
dbo.tblKuisisioner values('" + comboProsesTI.Text + "',''" + txtTempat.Text +
 "','4','" + Pertanyaan4Lv14.Text + "',''" + comboNilai4Lv14.Text + "',''" +
 lblTotalLv14.Text + "')", con);
        cmd22.ExecuteNonQuery();
        SqlCommand cmd23 = new SqlCommand("Insert into
dbo.tblKuisisioner values('" + comboProsesTI.Text + "',''" + txtTempat.Text +
 "','4','" + Pertanyaan5Lv14.Text + "',''" + comboNilai5Lv14.Text + "',''" +
 lblTotalLv14.Text + "')", con);
        cmd23.ExecuteNonQuery();
        SqlCommand cmd24 = new SqlCommand("Insert into
dbo.tblKuisisioner values('" + comboProsesTI.Text + "',''" + txtTempat.Text +
 "','4','" + Pertanyaan6Lv14.Text + "',''" + comboNilai6Lv14.Text + "',''" +
 lblTotalLv14.Text + "')", con);
        cmd24.ExecuteNonQuery();

        //lv15
        SqlCommand cmd25 = new SqlCommand("Insert into
dbo.tblKuisisioner values('" + comboProsesTI.Text + "',''" + txtTempat.Text +

```

```

'', '5', '' + Pertanyaan1Lv15.Text + ', ' + comboNilai1Lv15.Text + ', ' +
lblTotalLv15.Text + ''", con);
        cmd25.ExecuteNonQuery();
        SqlCommand cmd26 = new SqlCommand("Insert into
dbo.tblKuisisioner values('" + comboProsesTI.Text + ', ' + txtTempat.Text +
'', '5', '' + Pertanyaan2Lv15.Text + ', ' + comboNilai2Lv15.Text + ', ' +
lblTotalLv15.Text + ''", con);
        cmd26.ExecuteNonQuery();
        SqlCommand cmd27 = new SqlCommand("Insert into
dbo.tblKuisisioner values('" + comboProsesTI.Text + ', ' + txtTempat.Text +
'', '5', '' + Pertanyaan3Lv15.Text + ', ' + comboNilai3Lv15.Text + ', ' +
lblTotalLv15.Text + ''", con);
        cmd27.ExecuteNonQuery();
        SqlCommand cmd28 = new SqlCommand("Insert into
dbo.tblKuisisioner values('" + comboProsesTI.Text + ', ' + txtTempat.Text +
'', '5', '' + Pertanyaan4Lv15.Text + ', ' + comboNilai4Lv15.Text + ', ' +
lblTotalLv15.Text + ''", con);
        cmd28.ExecuteNonQuery();
        SqlCommand cmd29 = new SqlCommand("Insert into
dbo.tblKuisisioner values('" + comboProsesTI.Text + ', ' + txtTempat.Text +
'', '5', '' + Pertanyaan5Lv15.Text + ', ' + comboNilai5Lv15.Text + ', ' +
lblTotalLv15.Text + ''", con);
        cmd29.ExecuteNonQuery();
        SqlCommand cmd30 = new SqlCommand("Insert into
dbo.tblKuisisioner values('" + comboProsesTI.Text + ', ' + txtTempat.Text +
'', '5', '' + Pertanyaan6Lv15.Text + ', ' + comboNilai6Lv15.Text + ', ' +
lblTotalLv15.Text + ''", con);
        cmd30.ExecuteNonQuery();
        SqlCommand cmd31 = new SqlCommand("Insert into
dbo.tblKuisisioner values('" + comboProsesTI.Text + ', ' + txtTempat.Text +
'', '5', '' + Pertanyaan7Lv15.Text + ', ' + comboNilai7Lv15.Text + ', ' +
lblTotalLv15.Text + ''", con);
        cmd31.ExecuteNonQuery();
        SqlCommand cmd32 = new SqlCommand("Insert into
dbo.tblKuisisioner values('" + comboProsesTI.Text + ', ' + txtTempat.Text +
'', '5', '' + Pertanyaan8Lv15.Text + ', ' + comboNilai8Lv15.Text + ', ' +
lblTotalLv15.Text + ''", con);
        cmd32.ExecuteNonQuery();

        con.Close();

        //perhitungan
        C1.Text = Math.Round((Double.Parse(lblTotalLv10.Text)
/ Double.Parse("2")), 2).ToString();
        C2.Text = Math.Round((Double.Parse(lblTotalLv11.Text)
/ Double.Parse("5")), 2).ToString();
        C3.Text = Math.Round((Double.Parse(lblTotalLv12.Text)
/ Double.Parse("4")), 2).ToString();
        C4.Text = Math.Round((Double.Parse(lblTotalLv13.Text)
/ Double.Parse("7")), 2).ToString();
        C5.Text = Math.Round((Double.Parse(lblTotalLv14.Text)
/ Double.Parse("6")), 2).ToString();
        C6.Text = Math.Round((Double.Parse(lblTotalLv15.Text)
/ Double.Parse("8")), 2).ToString();

```

```

TOTALC.Text = Math.Round((Double.Parse(C1.Text) +
Double.Parse(C1.Text) + Double.Parse(C1.Text) +
Double.Parse(C1.Text) + Double.Parse(C1.Text)), 2).ToString();

D1.Text = Math.Round((Double.Parse(C1.Text) /
Double.Parse(TOTALC.Text)), 2).ToString();
D2.Text = Math.Round((Double.Parse(C2.Text) /
Double.Parse(TOTALC.Text)), 2).ToString();
D3.Text = Math.Round((Double.Parse(C3.Text) /
Double.Parse(TOTALC.Text)), 2).ToString();
D4.Text = Math.Round((Double.Parse(C4.Text) /
Double.Parse(TOTALC.Text)), 2).ToString();
D5.Text = Math.Round((Double.Parse(C5.Text) /
Double.Parse(TOTALC.Text)), 2).ToString();
D6.Text = Math.Round((Double.Parse(C6.Text) /
Double.Parse(TOTALC.Text)), 2).ToString();

Kontribusi1.Text = Math.Round((Double.Parse(D1.Text) *
Double.Parse("0")), 2).ToString();
Kontribusi2.Text = Math.Round((Double.Parse(D2.Text) *
Double.Parse("1")), 2).ToString();
Kontribusi3.Text = Math.Round((Double.Parse(D3.Text) *
Double.Parse("2")), 2).ToString();
Kontribusi4.Text = Math.Round((Double.Parse(D4.Text) *
Double.Parse("3")), 2).ToString();
Kontribusi5.Text = Math.Round((Double.Parse(D5.Text) *
Double.Parse("4")), 2).ToString();
Kontribusi6.Text = Math.Round((Double.Parse(D6.Text) *
Double.Parse("5")), 2).ToString();

MaturityValue.Text =
Math.Round((Double.Parse(Kontribusi1.Text) + Double.Parse(Kontribusi2.Text)
+ Double.Parse(Kontribusi3.Text) + Double.Parse(Kontribusi4.Text) +
Double.Parse(Kontribusi5.Text) + Double.Parse(Kontribusi6.Text)),
2).ToString();

//memasukan hasil dan perumusan perhitungan maturity
value ke database
try
{
SqlConnection connection = new
SqlConnection(ConString);
connection.Open();

SqlCommand cmd33 = new SqlCommand("Insert into
dbo.tblHasil values('0','" + lblTotalLvl0.Text + "','2','" + C1.Text + "',''"
+ D1.Text + "','" + Kontribusi1.Text + "')", connection);
cmd33.ExecuteNonQuery();

SqlCommand cmd34 = new SqlCommand("Insert into
dbo.tblHasil values('1','" + lblTotalLvl1.Text + "','5','" + C2.Text + "',''"
+ D2.Text + "','" + Kontribusi2.Text + "')", connection);
cmd34.ExecuteNonQuery();

SqlCommand cmd35 = new SqlCommand("Insert into
dbo.tblHasil values('2','" + lblTotalLvl2.Text + "','4','" + C3.Text + "',''"
+ D3.Text + "','" + Kontribusi3.Text + "')", connection);

```

```

        cmd35.ExecuteNonQuery();

        SqlCommand cmd36 = new SqlCommand("Insert into
        dbo.tblHasil values('3','" + lblTotalLv13.Text + "','7','" + C4.Text + "',''"
        + D4.Text + "',''" + Kontribusi4.Text + "')", connection);
        cmd36.ExecuteNonQuery();

        SqlCommand cmd37 = new SqlCommand("Insert into
        dbo.tblHasil values('4','" + lblTotalLv14.Text + "','6','" + C5.Text + "',''"
        + D5.Text + "',''" + Kontribusi5.Text + "')", connection);
        cmd37.ExecuteNonQuery();

        SqlCommand cmd38 = new SqlCommand("Insert into
        dbo.tblHasil values('5','" + lblTotalLv15.Text + "','8','" + C6.Text + "',''"
        + D6.Text + "',''" + Kontribusi6.Text + "')", connection);
        cmd38.ExecuteNonQuery();

        connection.Close();
    }
    catch (Exception x)
    {
        MessageBox.Show("Error");
        MessageBox.Show(x.Message);
    }

    formLaporanHasilKuisisioner next = new
    formLaporanHasilKuisisioner(MaturityValue.Text,txtTempat.Text,comboProsesTI.T
    ext,lblWaktu.Text,lblNama.Text,lblTotalLv10.Text,lblTotalLv11.Text,lblTotal
    Lv12.Text,lblTotalLv13.Text,lblTotalLv14.Text,lblTotalLv15.Text);
    this.Hide();
    next.Show();

    SqlConnection conn = new SqlConnection(ConString);
    conn.Open();

    SqlCommand cmdtempat = new SqlCommand("insert into
    dbo.tblTempat values('" + comboProsesTI.Text + "',''" + txtTempat.Text + "',''"
    + MaturityValue.Text + "',''" + lblWaktu.Text + "',''" + lblNama.Text + "')",
    conn);

    cmdtempat.ExecuteNonQuery();
    conn.Close();
    }
    catch (Exception x)
    {
        MessageBox.Show("Error");
        MessageBox.Show(x.Message);
    }
}

private void btnLanjut_Click(object sender, EventArgs e)
{
    fillKuisisioner();
}

private void formUtama_Load(object sender, EventArgs e)
{

```



```

String a = DateTime.Now.ToLongDateString();

lblWaktu.Text = a + " " + DateTime.Now.ToShortTimeString();
lblProses.Text = comboProsesTI.Text;
SqlConnection con = new SqlConnection(ConString);
con.Open();

SqlCommand cmd2 = new SqlCommand("truncate table tblHasil",
con);

cmd2.ExecuteNonQuery();

}

private void comboNilai1_KeyPress(object sender, KeyPressEventArgs
e)
{
    if (e.KeyChar != 13)
    {
        MessageBox.Show("Pilihlah 'Nilai' dari daftar yang telah
disediakan!", "Peringatan", MessageBoxButtons.OK, MessageBoxIcon.Warning);
        e.KeyChar = (char)0;
    }
}

void RefreshForm()
{
    comboNilai1Lv10.Text = "0";
    comboNilai2Lv10.Text = "0";

    comboNilai1Lv11.Text = "0";
    comboNilai2Lv11.Text = "0";
    comboNilai3Lv11.Text = "0";
    comboNilai4Lv11.Text = "0";
    comboNilai5Lv11.Text = "0";

    comboNilai1Lv12.Text = "0";
    comboNilai2Lv12.Text = "0";
    comboNilai3Lv12.Text = "0";
    comboNilai4Lv12.Text = "0";

    comboNilai1Lv13.Text = "0";
    comboNilai2Lv13.Text = "0";
    comboNilai3Lv13.Text = "0";
    comboNilai4Lv13.Text = "0";
    comboNilai5Lv13.Text = "0";
    comboNilai6Lv13.Text = "0";
    comboNilai7Lv13.Text = "0";

    comboNilai1Lv14.Text = "0";
    comboNilai2Lv14.Text = "0";
    comboNilai3Lv14.Text = "0";
    comboNilai4Lv14.Text = "0";
    comboNilai5Lv14.Text = "0";
    comboNilai6Lv14.Text = "0";

    comboNilai1Lv15.Text = "0";
    comboNilai2Lv15.Text = "0";

```

```

        comboNilai3Lv15.Text = "0";
        comboNilai4Lv15.Text = "0";
        comboNilai5Lv15.Text = "0";
        comboNilai6Lv15.Text = "0";
        comboNilai7Lv15.Text = "0";
        comboNilai8Lv15.Text = "0";

        lblTotalLv10.Visible = false;
        lblTotalLv11.Visible = false;
        lblTotalLv12.Visible = false;
        lblTotalLv13.Visible = false;
        lblTotalLv14.Visible = false;
        lblTotalLv15.Visible = false;

        lblTotalLv10.Text = "";
        lblTotalLv11.Text = "";
        lblTotalLv12.Text = "";
        lblTotalLv13.Text = "";
        lblTotalLv14.Text = "";
        lblTotalLv15.Text = "";

        txtTempat.Enabled = false;
        txtTempat.Text = "";
    }

    private void btnReset_Click(object sender, EventArgs e)
    {
        RefreshForm();
    }

    private void tabLevel0_Enter(object sender, EventArgs e)
    {
        lblLevel.Text = "0";
    }

    private void tabLevel1_Enter(object sender, EventArgs e)
    {
        lblLevel.Text = "1";
    }

    private void comboNilai1_SelectedIndexChanged(object sender,
    EventArgs e)
    {
        lblTotalLv10.Visible = true;
        //lblTotal.Text = comboNilai1.Text + comboNilai2.Text;

        lblTotalLv10.Text = (Double.Parse(comboNilai1Lv10.Text) +
        Double.Parse(comboNilai2Lv10.Text)).ToString();
    }

    private void comboNilai1Lv11_SelectedIndexChanged(object sender,
    EventArgs e)
    {
        lblTotalLv11.Visible = true;
    }

```

```

        lblTotalLv11.Text = (Double.Parse(comboNilai1Lv11.Text) +
Double.Parse(comboNilai3Lv11.Text) + Double.Parse(comboNilai5Lv11.Text) +
Double.Parse(comboNilai4Lv11.Text) +
Double.Parse(comboNilai2Lv11.Text)).ToString();
    }

    private void comboNilai1Lv12_SelectedIndexChanged(object sender,
EventArgs e)
    {
        lblTotalLv12.Visible = true;

        lblTotalLv12.Text = (Double.Parse(comboNilai1Lv12.Text) +
Double.Parse(comboNilai3Lv12.Text) + Double.Parse(comboNilai4Lv12.Text) +
Double.Parse(comboNilai2Lv12.Text)).ToString();

    }

    private void tabLevel2_Enter(object sender, EventArgs e)
    {
        lblLevel.Text = "2";
    }

    private void tabLevel3_Enter(object sender, EventArgs e)
    {
        lblLevel.Text = "3";
    }

    private void tabLevel4_Enter(object sender, EventArgs e)
    {
        lblLevel.Text = "4";
    }

    private void tabLevel5_Enter(object sender, EventArgs e)
    {
        lblLevel.Text = "5";
    }

    private void comboNilai1Lv13_SelectedIndexChanged(object sender,
EventArgs e)
    {
        lblTotalLv13.Visible = true;

        lblTotalLv13.Text = (Double.Parse(comboNilai1Lv13.Text) +
Double.Parse(comboNilai3Lv13.Text) + Double.Parse(comboNilai5Lv13.Text) +
Double.Parse(comboNilai4Lv13.Text) + Double.Parse(comboNilai6Lv13.Text) +
Double.Parse(comboNilai7Lv13.Text) +
Double.Parse(comboNilai2Lv13.Text)).ToString();

    }

    private void comboNilai1Lv15_SelectedIndexChanged(object sender,
EventArgs e)
    {
        lblTotalLv15.Visible = true;

        lblTotalLv15.Text = (Double.Parse(comboNilai1Lv15.Text) +
Double.Parse(comboNilai3Lv15.Text) + Double.Parse(comboNilai5Lv15.Text) +

```

```

Double.Parse(comboNilai4Lv15.Text) + Double.Parse(comboNilai6Lv15.Text) +
Double.Parse(comboNilai7Lv15.Text) + Double.Parse(comboNilai8Lv15.Text) +
Double.Parse(comboNilai2Lv15.Text)).ToString();

    }

    private void comboNilai1Lv14_SelectedIndexChanged(object sender,
EventArgs e)
    {
        lblTotalLv14.Visible = true;

        lblTotalLv14.Text = (Double.Parse(comboNilai1Lv14.Text) +
Double.Parse(comboNilai3Lv14.Text) + Double.Parse(comboNilai5Lv14.Text) +
Double.Parse(comboNilai4Lv14.Text) + Double.Parse(comboNilai6Lv14.Text) +
Double.Parse(comboNilai2Lv14.Text)).ToString();

    }
}
}

```