

Sistem Informasi Antropometri Terintegrasi Dengan Sistem Tertanam Sebagai Pengukur Berat Dan Tinggi Balita

Nugroho Budi Wicaksono¹⁾, Rinta Kridalukmana²⁾, Ike Pertiwi Windasari²⁾
Program Studi Sistem Komputer, Fakultas Teknik, Universitas Diponegoro
Jalan Prof. Sudharto, Tembalang, Semarang, Indonesia
nugrohobudi@ce.undip.ac.id

Abstract - Indonesian children's growth are monitored monthly and can be seen by the KMS graphic and posyandu cadre's book. The data from posyandu activity will be evaluated by the government health department. In practice, there are many obstacles to make posyandu works ideally. First, the cadres of posyandu have a difficulty to use the conventional anthropometric equipment, so the measurement data may not be valid and precise. Second, noting the KMS graphic needs an accuracy. Third, the measurement result in posyandu is not rapidly integrated to the central health data. A technology innovation of automatization anthropometry measurement called OTOTIM designed for infants that supported by an integrated information system is needed to solve those problems.

OTOTIM designed to make anthropometric measurement for infants more fast, valid, and accurate. The measurement result will be stored in the database and transferred to an image that describe the nutrition status in the KMS. Microcontroller will process the data from load sensor that measure baby's weight and ping sensor that measure baby's height. Then those raw data will be sent to main component of OTOTIM, which is the software itself to gain informative result of measurement, that consist of right suggestion on what kind of treatment used to take care various nutrition condition of infant and comparison against the previous measurement. OTOTIM will also generate statistics which group the infant by nutritional condition for each month. Those statistics will also include information on how percentage of male and female babies which have certain nutritional condition. However those statistic will provide information which is relevant only for specific district as OTOTIM used in regional manner. In order to create statistic which is relevant to consumed by public, those measurement data must be processed in central server which can calculate whole data and grouped them by nutritional condition. OTOTIM introduces the central server that bring synchronization service between OTOTIM software which is installed locally on client PC and the server itself. OTOTIM software have to establish a connection to server to publish subdistrict-specific statistic, however this connection is not mandatory to run primary function. So if client have trouble to connect, this will not interrupt the primary function as OTOTIM supposed to run in less-signal environment.

Keywords : Anthropometry, KMS, Anthropometric Measurement, Automation, Information System, Desktop Application, Web Service

I. PENDAHULUAN

Masa-masa emas tumbuh kembang seorang anak saat ia balita menentukan kesehatan anak saat ia dewasa. Balita yang memiliki tumbuh kembang yang baik akan menjadi generasi yang kuat, cerdas dan berkualitas. Asupan gizi yang baik tentu diperlukan untuk mewujudkan hal tersebut. Kekurangan asupan gizi dapat memicu timbulnya masalah tumbuh kembang anak yang berujung pada gizi buruk. Data Riset Kesehatan Dasar (Riskesdas) 2013 menunjukkan prevalensi balita gizi buruk dan kurang di Indonesia mencapai 19,6 persen. Angka tersebut meningkat dibandingkan dengan data Riskesdas 2010 sebesar 17,9 persen dan Riskesdas 2007 sebesar 18,4 persen.^[1]

Pada tahun 2010, Indonesia mengadopsi Standar Baru WHO 2005 sebagai pengganti rujukan WHO/NCHS melalui Kepmenkes RI No.1995/MENKES/SK/XII/2010 tentang penerapan Standar Antropometri Penilaian Status Gizi Anak. Sebagai konsekuensinya maka seluruh Tenaga Pelaksana Gizi (TPG) harus mendapatkan Pelatihan Penilaian Pertumbuhan Balita agar mampu melakukan penilaian pertumbuhan balita menggunakan minimal 3 indikator yaitu TB/U, BB/U, dan BB/TB. Hingga saat ini pelatihan tersebut masih terus dilaksanakan dengan harapan setelah dilatih mereka segera mengaplikasikannya di wilayah masing-masing.^[3]

Pelatihan Penilaian Pertumbuhan Balita dititikberatkan pada kemampuan TPG untuk melakukan penilaian terhadap status gizi balita dan bagaimana memantau pola pertumbuhan balita serta tindak lanjut yang dilakukan bila terjadi masalah atau gangguan pertumbuhan dengan standard dan prosedur yang tepat yaitu (a) menghitung umur balita; (b) melihat bila terdapat tanda klinis marasmus, kwashiorkor, atau gabungan keduanya; (c) melakukan penimbangan berat badan dan pengukuran panjang/tinggi badan dengan benar; (d) melakukan interpretasi terhadap status gizi balita minimal menggunakan 3 indikator serta melihat trend perkembangannya; (d) melakukan konseling pemberian makan sesuai masalah atau gangguan pertumbuhan yang dialami balita.^[3]

Berdasarkan penelusuran yang dilakukan melalui evaluasi pasca pelatihan penilaian pertumbuhan balita akhir tahun 2013 pada beberapa wilayah di Provinsi Jawa Barat, Lampung, NTB, dan Sulawesi Tengah, diketahui

bahwa penerapan penilaian pertumbuhan balita secara umum masih banyak yang belum sesuai standar dan prosedur sebagaimana yang diajarkan pada pelatihan. Namun hal ini bukan semata-mata karena TPG, tetapi juga dikarenakan minimnya kualitas dan kuantitas sarana dan prasarana di tempat kerja.^[3]

Tujuan penelitian adalah untuk menghasilkan sistem informasi yang mampu menyediakan layanan pengukuran berat dan tinggi badan balita menggunakan sensor dari modul pengukuran, menyediakan layanan analisa status gizi balita, menyediakan layanan basis data status gizi balita, dan mampu menyediakan akses data status gizi balita bagi pihak-pihak yang berwenang.

Untuk menghindari pembahasan yang meluas maka dalam Tugas Akhir ini ditetapkan batasan-batasan masalah sebagai berikut:

- a. Keluaran dari modul pengukuran dibaca melalui antarmuka serial port
- b. Sistem hanya menggunakan 3 indeks antropometri gizi, yaitu indeks berat badan menurut umur (BB/U), tinggi badan menurut umur (TB/U) dan berat badan menurut tinggi badan (BB/TB)
- c. Aplikasi pengolahan data di sisi klien dibangun dengan bahasa pemrograman C# untuk platform windows desktop dengan framework Windows Presentation Foundation (WPF) dan pola desain/perancangan Model-View-ViewModel (MVVM)
- d. Basis data yang digunakan sebagai sumber data aplikasi di sisi klien berasal dari 2 lokasi yaitu lokal yang dibangun menggunakan SQL Server Compact Edition; dan server yang dibangun menggunakan SQL Server Express
- e. Basis data di sisi server dengan aplikasi server IIS diakses melalui Web API yang dibangun menggunakan bahasa pemrograman C# dengan framework ASP.NET MVC dan ORM Entity Framework.
- f. Tahap distribusi pada proses pengembangan sistem tidak dilakukan, sehingga dalam pembahasan, sistem dijalankan pada lingkungan *localhost*, yang juga digunakan sebagai *development server*.

II. LANDASAN TEORI

A. Aplikasi KMS

Penelitian mengenai penggunaan aplikasi pada posyandu telah dilakukan oleh beberapa peneliti. aplikasi E-KMS untuk mencatat hasil pengukuran berat dan tinggi badan balita, catatan pemberian imunisasi dan pemberian Vitamin A. Aplikasi dibuat menggunakan bahasa pemrograman Visual Basic dan merupakan aplikasi stand alone, yaitu hanya dapat diakses dari komputer tempat aplikasi dipasang. Saran yang diberikan responden pada aplikasi tersebut adalah pemberian grafik berat badan balita. Aplikasi ini mendapatkan tanggapan positif dari kader posyandu, dimana sebagian besar berpendapat bahwa E-KMS dapat memperkecil kesalahan perhitungan umur balita, dan kader dapat langsung melihat catatan riwayat pemberian imunisasi dan Vitamin A. Aplikasi ini hanya bisa diakses oleh kader posyandu yang memiliki komputer untuk mengakses aplikasi tersebut, sehingga data dan riwayat kesehatan balita tidak bisa diakses sewaktu-

waktu oleh ibu balita.^[6]

Terdapat hubungan yang bermakna antara pemanfaatan KMS dengan tumbuh kembang balita. Oleh karena itu, fungsi dari KMS sebenarnya bukan hanya sebagai tempat mencatat hasil penimbangan berat badan anak saja tetapi yang paling penting adalah penggunaannya untuk memantau tumbuh kembang dan status kesehatan anak. Pencatatan riwayat kesehatan secara elektronik yang bisa diakses oleh ibu balita sebagai pengasuh menjadi tujuan utama penelitian ini. Penggunaan aplikasi mobile pada perangkat bergerak berbasis Android menjadi solusi yang diusulkan pada penelitian ini.^[5]

B. Antropometri

Antropometri adalah ilmu yang mempelajari berbagai ukuran tubuh manusia. Dalam bidang ilmu gizi digunakan untuk menilai status gizi. Ukuran yang sering digunakan adalah berat badan dan tinggi badan. Selain itu juga ukuran tubuh lainnya seperti lingkaran lengan atas, lapisan lemak bawah kulit, tinggi lutut, lingkaran perut, lingkaran pinggul. Ukuran-ukuran antropometri tersebut bisa berdiri sendiri untuk menentukan status gizi dibanding baku atau berupa indeks dengan membandingkan ukuran lainnya seperti BB/U, BB/TB, TB/U.^[8]

Secara umum antropometri artinya ukuran tubuh manusia. Ditinjau dari sudut pandang gizi, maka antropometri gizi berhubungan berbagai macam pengukuran dimensi tubuh dan komposisi dari berbagai tingkat umur dan tingkat gizi.^[9]

Di masyarakat, cara pengukuran status gizi yang paling sering digunakan adalah antropometri gizi. Dewasa ini dalam program gizi masyarakat, pemantauan status gizi anak balita menggunakan metode antropometri, sebagai cara untuk menilai status gizi. Di samping itu pula dalam kegiatan penapisan status gizi masyarakat selalu menggunakan metode tersebut.^[9]

Dalam pemakaian untuk penilaian status gizi, antropometri disajikan dalam bentuk indeks, misalnya berat badan menurut umur (BB/U), tinggi badan menurut umur (TB/U) atau berat badan menurut tinggi badan (BB/TB), lingkaran lengan atas menurut umur (LLA/U) dan sebagainya.^[10] Karena antropometri sebagai indikator penilaian status gizi yang paling mudah yang dapat dilakukan dengan mengukur beberapa parameter, antara lain: umur, berat badan, tinggi badan, lingkaran lengan atas, lingkaran kepala, lingkaran dada, lingkaran pinggul dan tebal lemak di bawah kulit. Oleh karena itu, untuk mengetahui status gizi seseorang, maka dilakukan pengukuran antropometri ini.

C. WPF

WPF sebelumnya dikenal dengan sebagai Avalon, Windows Presentation Foundation (WPF) adalah subsistem grafis baru di Windows Vista dengan menyediakan sarana untuk mengombinasikan Antarmuka pengguna, gambar 2D dan 3D, dokumen, dan media digital. WPF dibangun di atas .Net Framework, WPF menyediakan lingkungan yang sudah teratur untuk pengembangan dengan sistem operasi Windows. Hal ini memberi keuntungan dari investasi yang dibuat oleh

Microsoft dalam .Net Framework, dan memungkinkan pengembang akrab dengan teknologi .Net agar cepat dalam memulai mengembangkan aplikasi yang menggunakan WPF.^[15] Pada Gambar 1 dapat dilihat posisi WPF pada .Net Framework.



Gambar 1. WPF di dalam .Net Framework 3.5

Gambar 1 menunjukkan bahwa WPF bermula dari .Net Framework 3.0. Windows Presentation Foundation (WPF) adalah sebuah sistem presentasi generasi berikutnya untuk membangun aplikasi klien Windows dengan user experiences yang menggunakan secara visual. Inti dari WPF adalah sebuah mesin render yang berbasis vektor dan tidak bergantung pada resolusi yang dibangun untuk memanfaatkan hardware grafis modern.^[16]

D. MVVM

Model-View-ViewModel adalah variasi dari Model-View-Controller (MVC) yang dirancang sebagai platform pengembangan antarmuka pengguna modern. Menurut Horn (2010), pola Model-View-ViewModel merupakan pola pengembangan terbaru turunan dari pola Model-View-Controller (MVC). Pada MVC, Controller berisi logika yang menangani event dari antarmuka pengguna dan menangani penampilan data di View. Pola desain MVVM memiliki kemampuan baru dalam data-binding (pengikatan data) yang terdapat pada ViewModel. Controller pada pola desain MVVM diganti ViewModel yang tugasnya mengontrol penampilan dari View.^[18]

MVVM juga merupakan variasi dari pola desain Presentation Model yang digagas oleh Martin Fowler.^{[19][20]} MVVM mengabstraksi keadaan dan perilaku dari layer view.^[21] Berbeda dengan Presentation Model dimana layer view diabstraksi menggunakan view model tanpa adanya ketergantungan pada platform dari antarmuka pengguna, MVVM yang dikembangkan oleh Microsoft secara spesifik bertujuan untuk menyederhanakan *event-driven programming* dari antarmuka bagi pengguna dengan mengeksplotasi beragam fitur dari WPF dan Silverlight.

MVVM memfasilitasi pemisahan fokus kerja antara proses pengembangan *Graphical User Interface* (GUI) dengan pengembangan modul logika bisnis atau modul pemodelan data. ViewModel pada MVVM dapat dikatakan sebagai pengkonversi nilai^[21], yang diartikan bahwa viewModel bertanggung jawab untuk mengekspos keadaan suatu objek dari pemodelan data yang mampu memudahkan layer view dalam memenej dan mengonsumsi objek tersebut.

E. ASP.NET Web API

ASP.NET Web API adalah framework yang digunakan untuk membangun layanan data berbasis HTTP yang dapat dikonsumsi oleh berbagai jenis platform klien, seperti aplikasi *browsers* di *desktop* ataupun perangkat bergerak. ASP.NET Web API adalah platform yang sangat ideal untuk membangun aplikasi *REST* (*Representational State Transfer*) di atas pondasi .NET Framework.^[22]

ASP.NET Web API memiliki suatu *interface* bernama *System.Web.IHttpAsyncHandler* atau dikenal sebagai *handler* yang menangani setiap permintaan HTTP secara asinkron dan menghasilkan respon sesuai permintaan klien. *Handler* juga merupakan landasan bagi ASP.NET MVC dalam menangani permintaan HTTP. Instansiasi ASP.NET MVC dan ASP.NET Web API dapat hidup berdampingan di atas ASP.NET *runtime* yang disediakan oleh IIS (Internet Information Service). ASP.NET akan menggunakan *event-handler* *HttpApplication.MapRequestHandler* untuk menentukan *handler* mana yang akan digunakan pada permintaan HTTP yang diterima. Dalam fase tersebut, terjadi proses pencocokan routing antara URL dari permintaan HTTP dengan tabel routing yang didefinisikan di kelas *RouteConfig* dan permintaan tersebut akan mengalir menuju *IRouteHandler* yang relevan dengan hasil routing. Tujuan utama dari *IRouteHandler* adalah untuk menghasilkan objek yang mengimplementasi *IHttpHandler* untuk menangani permintaan yang dimaksud.^[23]

F. ORM Entity Framework

ORM (Object Relational Mapping) adalah suatu metode/teknik pemrograman yang digunakan untuk mengkonversi data dari lingkungan bahasa pemrograman berorientasi objek (OOP) dengan lingkungan database relasional. Seperti kita ketahui, dalam aplikasi enterprise kedua lingkungan tersebut berada pada sistem yang berbeda, yaitu OOP berada pada sisi pemrograman aplikasi, sedangkan database relasional berada pada sisi sistem database. Misi utama dari ORM ini adalah menjembatani kedua sistem yang berbeda tersebut.^[24]

ORM memiliki kemampuan untuk menciptakan objek database virtual, yaitu suatu model database yang di representasikan kedalam sebuah objek pada bahasa pemrograman OOP. Berikut ini adalah beberapa kelebihan yang dimiliki ORM, yaitu:^[24]

1. Mempercepat pengembangan program. Contohnya, mengurangi perulangan kode query, memudahkan pemakaian karena tabel-tabel ter-representasikan dalam bentuk objek
2. Membuat akses data menjadi lebih abstrak dan portable. Hal ini dikarenakan ORM menghandle *generate-an syntax SQL* berdasarkan vendor database-nya.
3. Mendukung pengapsulan business rule pada lapisan Data Access.
4. Menghasilkan *boilerplate code* (unit kode yang reusable) untuk fungsi dasar CRUD (Create, Read, Update, Delete).

Entity Framework adalah sekumpulan teknologi ADO.NET yang membantu proses mapping antara pengembangan object-oriented dan database. Metode ini dikenal dengan ORM (Object-Rational Mapping). Kalau diteliti lebih dalam Entity Framework lebih sekedar dari ORM karena Entity Framework juga dapat bekerja sebagai ERM (Entity Relationship Model). Selain Entity Framework, kita juga dapat menjumpai solusi lain yaitu:^[25]

1. NHibernate. Solusi persistent .NET object ke relational database. NHibernate merupakan hasil porting dari Hibernate java ke .NET Framework
2. SPRING.net. Framework berbasis open source yang merupakan hasil porting Spring Framework dimana ini memungkinkan kita untuk membuat komponen yang terintegrasi kedalam multi-tier application

G. Agile Development Methods

Agile Development Methods adalah sekelompok metodologi pengembangan perangkat lunak memerlukan adaptasi cepat dari pengembang terhadap perubahan dalam bentuk apapun. Agile development methods merupakan salah satu dari Metodologi pengembangan perangkat lunak yang digunakan dalam pengembangan perangkat lunak. Saat membuat perangkat lunak dengan menggunakan agile development methods diperlukan inovasi dan tanggung jawab yang baik antara tim pengembang dan klien agar kualitas dari perangkat lunak yang dihasilkan bagus dan kelincahan dari tim seimbang.^[28]

Story adalah daftar kebutuhan atau fitur yang nanti akan dibuat. Story berisi apa yang klien kehendaki, dan ditulis dalam bahasa yang dimengerti klien. Dengan kata lain dapat disimpulkan Story adalah bagian terpenting dari Scrum. Story terdiri dari kolom-kolom berikut ini^[34]:

1. ID – Identifikasi unik, biasanya berupa nomor urut. Hal ini untuk menghindari kehilangan jejak story kalau kita mengganti namanya.
2. Persona – Berbagai pihak yang ingin menggunakan sistem untuk menjalankan aksi-aksi tertentu.
3. Aksi – Aktivitas yang ingin dilaksanakan oleh klien dalam sistem yang dibangun untuk menghasilkan suatu nilai guna.
4. Kegunaan – Nilai guna yang didapatkan oleh suatu persona saat menjalankan aksi-aksi tertentu.

H. User Acceptance Testing

Acceptance Test-Driven Development (ATDD) adalah metodologi pengujian sistem berdasarkan komunikasi antara pengguna proses bisnis, pengembang, dan penguji.^[1] ATDD dapat merujuk ke beberapa pendekatan seperti *Specification by Example*,^[2]^[3] *Behavior Driven Development (BDD)*,^[4] *Example-Driven Development (EDD)*,^[5] and *Story Test-Driven Development (SDD)*.^[6] Semua proses tersebut bertujuan agar pengembang dan penguji dapat memahami skala prioritas kebutuhan pengguna proses bisnis dari sistem yang dibangun dan memberikan kesempatan bagi pengguna untuk turut aktif memberikan *feedback* dari setiap pengujian menggunakan bahasa sehari-hari.

Acceptance criteria adalah deskripsi dari apa yang akan dicek pada saat pengujian. Diketahui sebuah kebutuhan dimana, “Sebagai pengguna, saya ingin meminjam buku dari perpustakaan”, sebuah *acceptance criteria* dapat berupa, “Verifikasi buku dengan memberikan tanda bahwa buku telah dipinjam”. Sebuah *acceptance test* untuk kebutuhan tersebut memberikan informasi lebih lanjut sehingga pengujian dapat dilakukan berulang kali dengan pengaruh yang sama. *Acceptance tests* umumnya mengikuti format sebagai berikut:^[1]

Diketahui (keadaan)

Suatu keadaan tertentu pada program

Ketika (pemicu)

Sebuah aksi atau *event* yang terjadi

Maka (verifikasi)

Keadaan dari program telah berubah atau telah menghasilkan suatu keluaran

Untuk contoh yang telah dijelaskan sebelumnya, maka dapat ditulis dengan format sebagai berikut:

Diketahui:

Buku yang belum dipinjam

Pengguna yang telah terdaftar pada sistem

Ketika:

Pengguna meminjam buku

Maka:

Sistem memberikan tanda bahwa buku telah dipinjam

Contoh tersebut tidak menunjukkan adanya contoh data spesifik, untuk itu perlu ditambahkan hingga membentuk sebuah *test* yang lengkap seperti contoh di bawah ini:

Diketahui:

Buku yang belum dipinjam

Tabel 1. Entitas buku

Judul	Sudah dipinjam
<i>Great book</i>	Belum

Pengguna yang telah terdaftar pada sistem

Tabel 2. Entitas peminjam

Nama	Alamat
Sam	Semarang

Ketika:

Pengguna meminjam buku

Tabel 3. Entitas aksi

<i>Checkout action</i>			
Pengguna	Sam	Meminjam	<i>Great book</i>

Maka:

Sistem memberikan tanda bahwa buku telah dipinjam

Tabel 4. Relasi entitas buku dengan entitas peminjam

Judul	Sudah dipinjam	Peminjam
<i>Great book</i>	Ya	Sam

III. METODOLOGI PENGEMBANGAN DAN TAHAPAN ITERASI 1

Metode pengembangan perangkat lunak yang digunakan dalam penelitian ini adalah *Agile Software Development*. Dan bab ini membahas langkah-langkah pengembangan sistem pada iterasi pertama sesuai dengan tahapan yang terdapat pada metode *Agile*. Langkah-langkah metode *agile* yaitu mulai dari perencanaan awal dan pelaksanaan beberapa iterasi yang masing-masing

iterasi terdiri dari tahapan perencanaan, desain, implementasi, dan pengujian. Sistem dikatakan siap untuk dirilis apabila pada tahapan akhir dalam iterasi menunjukkan bahwa semua kebutuhan pengguna yang didokumentasikan dalam *sprint backlog* berhasil dipenuhi.

A. Perencanaan dan Perancangan Iterasi 1

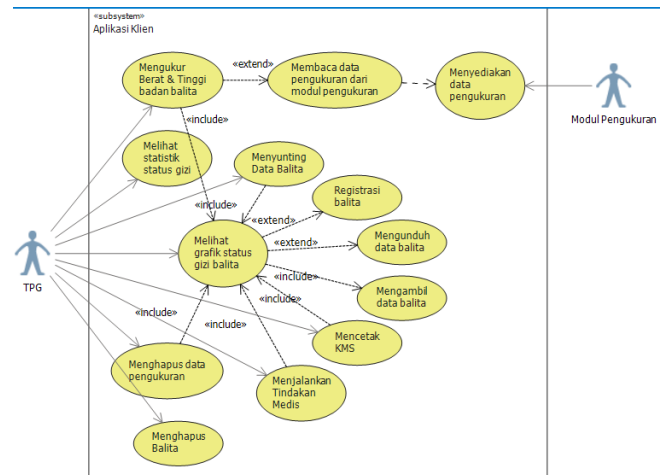
Pada tahapan perencanaan, seluruh artefak yang diperlukan dalam proses pengembangan dikumpulkan. Artefak tersebut meliputi *user story*, lembar KMS, dokumen keputusan Menkes RI tentang standar antropometri penilaian status gizi anak (Nomor: 1995/MENKES/SK/XII/2010) dan dokumen peraturan Menkes RI tentang penggunaan KMS bagi balita (Nomor: 155/MENKES/PER/I/2010). *User story* didapat dari hasil wawancara dengan Tenaga Pelaksana Gizi (TPG) dan pakar gizi untuk kemudian disusun dalam sebuah *product backlog*.

Perancangan meliputi analisa kebutuhan pengguna dan sistem pada iterasi pertama. Analisa kebutuhan mendeskripsikan kebutuhan pengguna yang ditargetkan untuk dapat dipenuhi oleh sistem dan kebutuhan sumber daya perangkat yang diperlukan oleh sistem agar dapat berjalan optimal. Untuk mempermudah dokumentasi sistem, analisa kebutuhan dibagi menjadi dua jenis, yaitu analisa kebutuhan pengguna yang mendeskripsikan apa yang diinginkan oleh pengguna pada sistem yang akan dibangun, dan analisa kebutuhan sistem yang mendeskripsikan spesifikasi infrastruktur teknologi informasi, seperti spesifikasi perangkat keras, sistem operasi, dan jaringan yang dibutuhkan agar sistem dapat berjalan sesuai keinginan pengguna. Untuk mempermudah penyebutan kesatuan sistem yang dibangun, dalam pembahasan selanjutnya sistem diberi nama **Ototim**, yang merupakan kepanjangan dari Otomatisasi Penimbangan.

B. Desain Sistem

Dalam tahapan desain sistem, dihasilkan pemodelan sistem dengan standar pemodelan UML 2.0, berupa Use Case, EDM (Entity Data Model), Class Diagram, Activity Diagram dan Component Diagram.

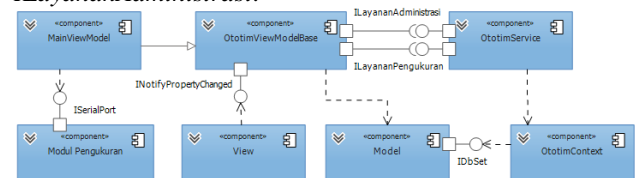
Dalam diagram use case di bawah ini, terdapat 2 aktor yaitu TPG dan Modul Pengukuran. TPG dan Modul Pengukuran menggunakan subsistem Aplikasi Klien seperti yang tertera pada gambar dibawah ini:



Gambar 2. Use Case dari Sistem yang akan Dibangun

Gambar 4 menunjukkan *class diagram* dari lapisan Model. Setiap kelas pada lapisan ini akan diproyeksikan dalam sebuah tabel melalui layanan yang disediakan oleh *ORM Entity Framework*. Setiap *method Get* yang terdapat pada setiap kelas masing-masing digunakan untuk mendapatkan nilai dari atribut entitas tersebut, sedangkan *method Set* digunakan untuk meingikan atribut dengan suatu nilai. Gambar 5 menunjukkan *class diagram* dari lapisan *Data Access Layer* yang digunakan oleh lapisan *View Model* untuk mengenakan tindakan *CRUD* pada suatu entitas.

Terlihat bahwa terdapat 2 kelas utama yang terdiri dari *OtotimContext* dan *OtotimService*. *OtotimContext* adalah kelas yang memiliki referensi terhadap entitas yang merupakan proyeksi dari sebuah tabel. Instansi dari *OtotimContext* akan digunakan oleh *OtotimService* untuk menerapkan aturan bisnis yang ada pada proses manipulasi data pada basis data. *OtotimService* mengimplementasi *interface I LayananPengukuran* dan *I LayananAdministrasi*.

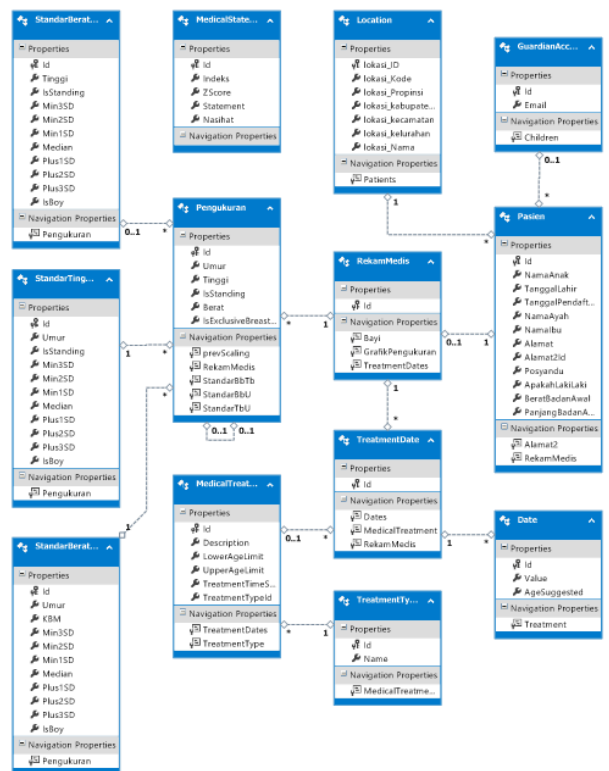


Gambar 3. Component diagram ototim untuk iterasi 1

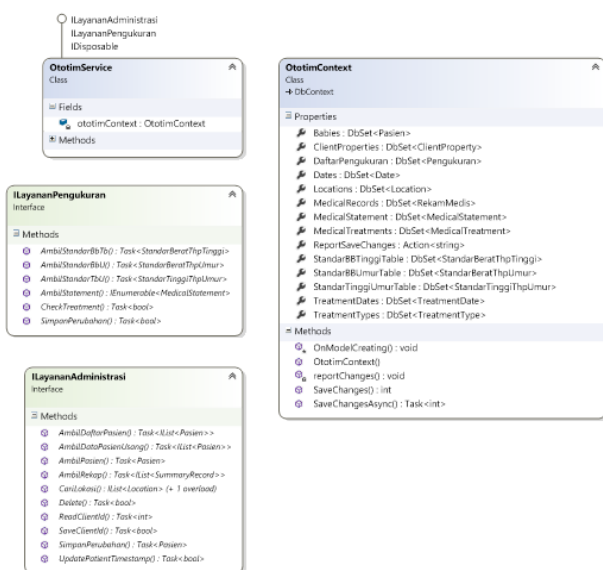
Iterasi 1 berfokus pada pengembangan aplikasi di sisi klien, yang menggunakan platform *Windows Desktop Application*. Dalam pengembangan, digunakan kerangka kerja MVVM. Setiap lapisan pada kerangka kerja didefinisikan oleh satu atau beberapa komponen. Untuk lapisan View, komponen View memerlukan sebuah komponen yang mengimplementasi *INotifyPropertyChanged* sebagai lapisan *ViewModel*. *Interface INotifyPropertyChanged* digunakan untuk memberi notifikasi apabila terjadi perubahan nilai *member* yang dimuat oleh lapisan *ViewModel* kepada setiap *control* yang dimuat dalam *view*, yang beberapa propertinya dikenakan proses *data-binding* untuk *member* tersebut. Notifikasi akan ditindak lanjuti oleh *view* dengan memuat ulang *control* terkait agar memunculkan data terkini pada perubahan terakhir.



Gambar 4. Class diagram dari lapisan Model



Gambar 6. Entity Data Model



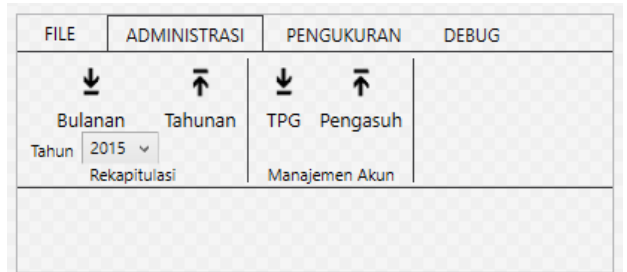
Gambar 5. Class diagram dari lapisan Data Access Layer

Dalam menjalankan fungsinya, aplikasi klien mengenakan operasi CRUD (*Create, Read, Update, Delete*) pada data yang diperlukan oleh TPG. Untuk memudahkan operasi tersebut, perlu dirancang suatu skema basis data ternormalisasi dengan entitas sesuai dengan kebutuhan yang tercantum pada *product backlog*. Skema basis data dapat dimodelkan dalam bentuk EDM (Entity Data Model), dimana setiap entitas merepresentasikan satu jenis record yang dapat tersimpan di sebuah tabel. Gambar dibawah ini merupakan EDM dari hasil normalisasi akhir basis data:

C. Implementasi

Platform perangkat lunak yang digunakan pada sisi klien adalah Windows Desktop Application. *Design Pattern* MVVM (Model View View-Model) digunakan sebagai kerangka kerja pembuatan perangkat lunak yang ditulis dalam bahasa C#. Pembuatan perangkat lunak akan dibagi ke dalam beberapa lapisan, yang terdiri dari lapisan Model, View, View-Model, dan Data Access Layer. Pembuatan Data Access Layer

Lapisan view disusun oleh beberapa komponen *usercontrol* hingga membentuk suatu antarmuka yang dapat berinteraksi dengan pengguna. *View* MainWindow merupakan *master layout* yang berfungsi sebagai *container* dari *user control* yang telah dibuat dan merupakan *view* yang pertama kali dipanggil saat aplikasi dijalankan. Penggunaan *master layout* diperlukan untuk menghasilkan antarmuka yang konsisten untuk setiap interaksi.

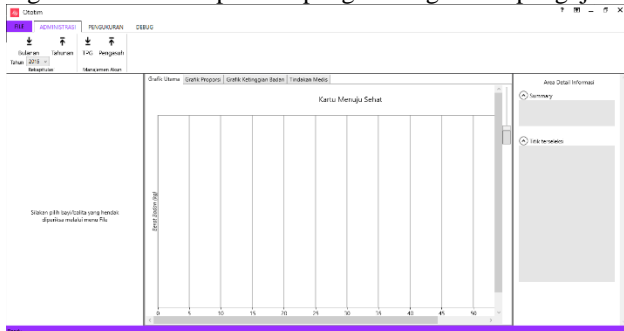


Gambar 7. User control ribbon

MainWindow tersusun atas *ApplicationMenu*, *BodyGrid*, *Ribbon* dan *StatusBar*. Konsistensi yang dihasilkan menyebabkan menu dan status bar untuk tetap tampil di posisi dan keadaan yang sama untuk konten aplikasi yang berbeda. Konten aplikasi dapat berupa salah

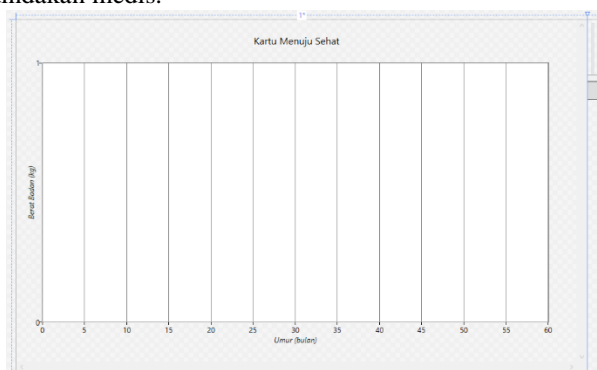
satu dari tiga jenis grafik pengukuran atau daftar tindakan medis. Berikut ini adalah *view* dari *Ribbon*.

Ribbon terdiri dari 3 menu utama, yaitu *File*, *Administrasi*, dan *Pengukuran*. Menu *Debug* hanya akan tampil apabila aplikasi dijalankan pada CLR dengan menyematkan *debugger*, atau suatu *process host* yang mampu menghentikan eksekusi instruksi pada program yang memunculkan suatu eksepsi. Menu *Debug* hanya digunakan untuk keperluan pengembangan dan pengujian.



Gambar 8. *Main window*

Di sisi kiri dari konten aplikasi terdapat informasi bayi/balita yang terasosiasi dengan grafik terkait dan di sisi kanan terdapat informasi pengukuran dari titik yang terseleksi pada grafik pengukuran terkait. Setiap grafik pengukuran diletakkan pada *BodyGrid* yang masing-masing dibuat terpisah sebagai *usercontrol*. Berikut ini akan dijelaskan tampilan *usercontrol* grafik berat badan terhadap umur, grafik tinggi badan berbanding umur, grafik berat badan berbanding tinggi badan dan daftar tindakan medis.



Gambar 9. *User control Kms Chart*

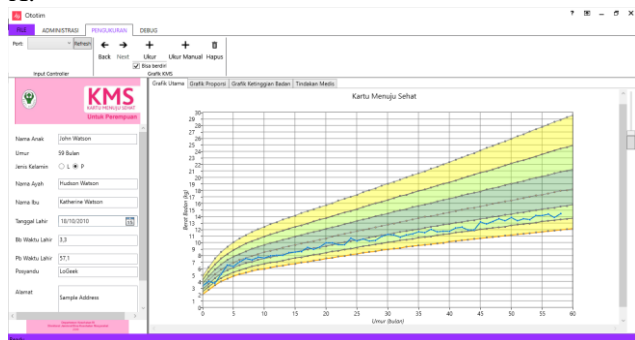
User control di atas digunakan untuk menampilkan grafik antropometri penilaian status gizi menurut berat badan yang ditinjau berdasarkan umur. Setiap grafik antropometri digambarkan dalam bentuk diagram kartesius. Terlihat bahwa pada sumbu X, satuan bulan digunakan untuk umur dan pada sumbu Y, satuan kg digunakan untuk berat badan. Sesuai dengan format KMS yang ditentukan oleh Menteri Kesehatan, umur maksimal balita yang dapat ditinjau adalah 60 bulan, berat badan maksimal laki-laki yang dapat diukur sebesar 29,7 kg dan untuk perempuan sebesar 31,2 kg. Maka pada grafik diatas, untuk anak laki-laki, koordinat pengukuran yang dapat dimunculkan adalah x,y dimana

$$\{x \in X : 0 \leq n \leq 60\} \text{ dan } \{y \in Y : 0 \leq n \leq 29,7\}$$

Dan untuk anak perempuan, koordinat pengukuran yang dapat dimunculkan adalah x,y dimana

$$\{x \in X : 0 \leq n \leq 60\} \text{ dan } \{y \in Y : 0 \leq n \leq 31,2\}$$

Konten pada grafik, terdiri dari 7 *area series* yang masing-masing memuat nilai standar berat badan dengan $ZScore +3SD, +2SD, +1SD, Median, -1SD, -2SD, -3SD$, dan 1 *line series* untuk memuat nilai hasil pengukuran. Dalam pengembangan antarmuka menggunakan *WPF framework*, *user control chart* dapat mengimplementasi berbagai jenis *series* termasuk *area series* dan *line series*. Kedua *series* tersebut memuat titik koordinat yang membutuhkan nilai *dependentvalue* sebagai nilai pada sumbu Y dan *independentvalue* sebagai nilai pada sumbu X.



Gambar 10. *User control Kms Chart* yang dimunculkan pada *Main Window*

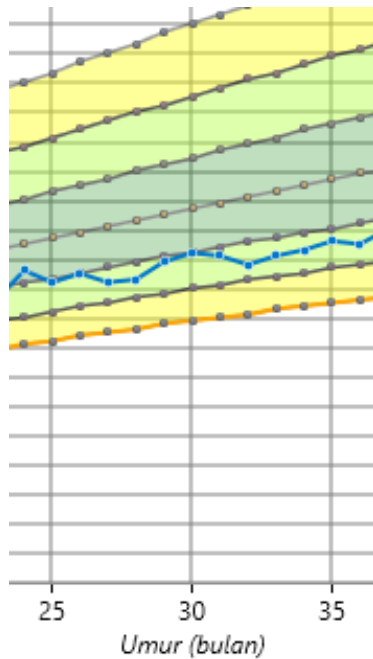
Setiap *series* memiliki kumpulan titik koordinat yang nilainya di-*binding* dengan nilai yang ada di *MainViewModel*. Seperti yang telah dijelaskan pada sub bab pembuatan model, untuk entitas *StandarBeratThpUmur*, digunakan *member* Umur sebagai *key* dan setiap *member* pada *baseclass* yang namanya berakhir dengan *char sequence SD* dapat dijadikan *value*. Pada *MainViewModel* yang akan dijelaskan secara mendalam pada sub bab selanjutnya, seluruh *record* pada tabel berentitas *StandarBeratThpUmur* akan diambil oleh instansi *MainViewModel* dan disimpan pada *member* yang mengimplementasi *IList<T>* dimana dalam hal ini *T* adalah tipe data *StandarBeratThpUmur*. *Member* tersebut dapat membentuk sebuah *logical table* sebagai berikut:

Tabel 5. *Logical table* dari nilai sebuah *member* bertipe data

IList<StandarBeratThpUmur>

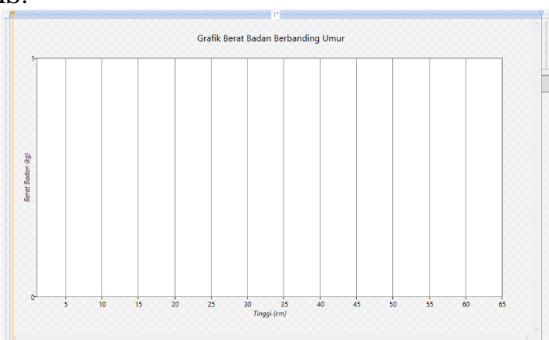
		Value (<i>Dependent Value</i>)				
		-2SD	-3SD	-1SD	...	+3SD
Key (<i>Independent Value</i>)	1					
	2					
	3					
	...					
	60					

User control chart memiliki *member data source* yang akan diarahkan ke *member* tersebut, dan setiap item pada *series +3SD* pada grafik akan di-*binding* dengan *member* umur sebagai *independentvalue* dan *member Plus3SD* sebagai *dependentvalue*. Berikut ini adalah potongan gambar yang akan menjelaskan mekanisme pembentukan grafik.



Gambar 11. Beberapa titik koordinat yang ditampilkan untuk x bernilai 30

Pada umur 30 bulan, terdapat 8 titik koordinat. Titik koordinat berwarna biru menunjukkan hasil pengukuran, dan titik koordinat lain menunjukkan standar berat badan untuk ketujuh nilai *ZScore*. Untuk +3SD, ditunjukkan oleh kumpulan titik koordinat yang berada paling atas, dan -3SD ditunjukkan oleh kumpulan titik koordinat yang berada paling bawah dan dihubungkan oleh garis berwarna jingga. Dengan merujuk pada *logical table*, titik koordinat yang menunjukkan nilai berat badan dengan *ZScore* +3SD pada umur 30 bulan, akan mempunyai *dependentvalue* dan *independentvalue* yang dibinding dengan *member* yang bersumber pada entitas *StandarBeratThpUmur* dengan *member* umur bernilai 30. Mekanisme yang sama berlaku pula untuk titik koordinat lain hingga membentuk grafik KMS.



Gambar 12. *User control* grafik berat badan terhadap tinggi badan

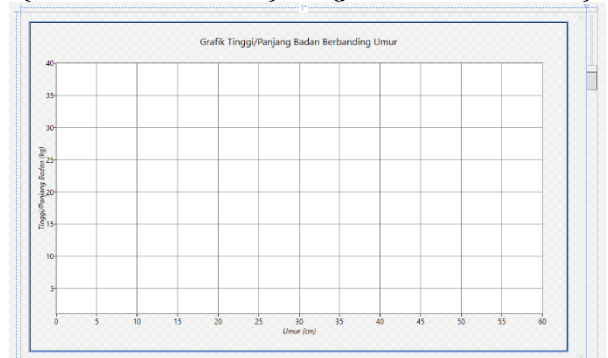
Sama halnya dengan grafik KMS, Pembentukan grafik berat badan berbanding tinggi badan akan menggunakan kumpulan entitas *StandarBeratThpTinggi* dan entitas *Pengukuran* dari rekam bayi/balita yang dibuka. Terlihat bahwa pada sumbu X, satuan cm digunakan untuk mengukur tinggi badan dan pada sumbu Y, satuan kg digunakan untuk mengukur berat badan. Sesuai dengan tabel antropometri yang dibuat oleh Menteri Kesehatan, tinggi maksimal balita yang dapat ditinjau adalah 120 cm, berat badan maksimal laki-laki yang dapat diukur sebesar

29,7 kg dan untuk perempuan sebesar 31,2 kg. Maka pada grafik diatas, untuk anak laki-laki, koordinat pengukuran yang dapat dimunculkan adalah x,y dimana

$$\{x \in X : 0 \leq n \leq 120\} \text{ dan } \{y \in Y : 0 \leq n \leq 29,7\}$$

Dan untuk anak perempuan, koordinat pengukuran yang dapat dimunculkan adalah x,y dimana

$$\{x \in X : 0 \leq n \leq 120\} \text{ dan } \{y \in Y : 0 \leq n \leq 31,2\}$$



Gambar 13. *User control* grafik tinggi/panjang badan terhadap umur

Untuk grafik tinggi/panjang badan berbanding umur, terlihat bahwa pada sumbu X, satuan bulan digunakan untuk mengukur umur dan pada sumbu Y, satuan cm digunakan untuk mengukur tinggi badan. Sesuai dengan tabel antropometri yang dibuat oleh Menteri Kesehatan, umur maksimal balita yang dapat ditinjau adalah 60 bulan, tinggi badan maksimal laki-laki yang dapat diukur sebesar 125 cm dan untuk perempuan sebesar 120 cm. Maka pada grafik diatas, untuk anak laki-laki, koordinat pengukuran yang dapat dimunculkan adalah x,y dimana

$$\{x \in X : 0 \leq n \leq 60\} \text{ dan } \{y \in Y : 0 \leq n \leq 125\}$$

Dan untuk anak perempuan, koordinat pengukuran yang dapat dimunculkan adalah x,y dimana

$$\{x \in X : 0 \leq n \leq 60\} \text{ dan } \{y \in Y : 0 \leq n \leq 120\}$$

Setiap *chart* dapat diperbesar atau diperkecil dengan mengubah posisi *slider* pada *user control slider* yang diposisikan tepat di samping kanan *chart*, dimana semakin tinggi posisi *slider*, semakin besar nilai pembesaran. Terdapat *user control button* yang diposisikan tepat dibawah *slider*, yang digunakan untuk memuat ulang grafik yang telah ditampilkan.

Umur (bulan)	Deskripsi	Tanggal

Gambar 14. *User control* tabel riwayat tindakan medis

Keempat konten aplikasi tersebut dipisahkan oleh kontrol tabulasi dan antara konten, dan *user control* di sisi kiri dan kanan konten terdapat pemisah yang dapat digeser ke kanan atau ke kiri untuk mengatur area konten yang dapat ditampilkan.

D. Pengujian

Pengujian dilakukan menggunakan metode *User Acceptance Testing (UAT)*. Entitas yang digunakan pada pengujian meliputi entitas *Pasien*, entitas *Pengukuran* dan entitas *TindakanMedis*. Semua *user story* yang

didefinisikan pada sub bab sebelumnya akan dijadikan acuan dalam membentuk *test case* yang tepat untuk mengetahui apakah kebutuhan user telah terpenuhi.

Tabel 6 Daftar *test case* untuk UAT iterasi 1

Id User Story	Diketahui	Ketika	Maka
1, 2	TPG belum memilih bayi/balita yang hendak ditimbang	TPG memulai pengukuran baru secara manual atau otomatis	Proses pengukuran tidak dapat berjalan, Grafik tumbuh kembang balita tidak tampil
2	TPG telah memilih bayi/balita yang hendak ditimbang, Port dengan deskripsi Arduino tidak ditemukan	TPG memulai pengukuran baru secara otomatis	Proses pengukuran tidak dapat berjalan
2	TPG telah memilih bayi/balita yang hendak ditimbang, Port dengan deskripsi Arduino tidak ditemukan	TPG memulai pengukuran secara manual	Proses pengukuran dapat berjalan
2	TPG telah memilih bayi/balita yang hendak ditimbang, Port dengan deskripsi Arduino ditemukan	TPG memulai pengukuran secara otomatis atau manual	Proses pengukuran dapat berjalan
1	Terdapat lebih dari satu isian data bayi/balita yang tidak sesuai dengan ketentuan	TPG menyimpan rekam bayi/balita	Proses penyimpanan tidak dapat berjalan
1	Semua isian data bayi/balita sesuai dengan ketentuan	TPG menyimpan rekam bayi/balita	Penyimpanan berhasil

1	Pada basisdata telah tersimpan rekam bayi/balita yang hendak dilihat oleh TPG	TPG mencari rekam bayi/balita yang hendak dilihat	Rekam bayi/balita ditemukan
1	Terdapat satu atau beberapa rekam bayi/balita yang ditemukan sesuai dengan kriteria pencarian	TPG membuka rekam bayi/balita	Data tumbuh kembang balita ditampilkan
1, 9	Terdapat satu atau beberapa rekam bayi/balita dalam basisdata	TPG melihat hasil rekapitulasi status gizi	Rekapitulasi data dapat ditampilkan
1, 9	Tidak terdapat rekam bayi/balita dalam basisdata	TPG melihat hasil rekapitulasi	Rekapitulasi data tidak dapat ditampilkan
3, 4	Terdapat satu atau lebih pengukuran pada grafik antropometri	TPG menyeleksi titik pengukuran	Deskripsi titik pengukuran ditampilkan
3	TPG menyeleksi titik pengukuran dimana pada bulan sebelumnya, tidak dilakukan pengukuran	TPG menyeleksi titik pengukuran	Data kenaikan berat badan tidak ditampilkan
3	TPG menyeleksi titik pengukuran dimana pada	TPG menyeleksi titik pengukuran	Data kenaikan berat badan ditampilkan

	bulan sebelumnya, juga dilakukan pengukuran		
3	Masih terdapat pengukuran pada 1 atau beberapa bulan setelah dilakukan pengukuran yang diseleksi TPG	TPG menekan tombol <i>Next Scaling</i>	Deskripsi titik pengukuran selanjutnya, ditampilkan
3	Tidak terdapat pengukuran setelah dilakukan pengukuran yang diseleksi TPG	TPG menekan tombol <i>Next Scaling</i>	Tidak terjadi perubahan keadaan pada program
3	Masih terdapat pengukuran pada 1 atau beberapa bulan sebelum dilakukan pengukuran yang diseleksi TPG	TPG menekan tombol <i>Previous Scaling</i>	Deskripsi titik pengukuran sebelumnya, ditampilkan
3	Tidak terdapat pengukuran sebelum dilakukan pengukuran yang diseleksi TPG	TPG menekan tombol <i>Previous Scaling</i>	Tidak terjadi perubahan keadaan pada program
1	Pada bulan ini sudah dilakukan pengukuran	TPG memulai pengukuran baru atau manual	Data pengukuran lama ditimpa dengan data pengukuran baru
1	Pada bulan ini belum dilakukan pengukuran	TPG memulai pengukuran baru atau manual	Data pengukuran baru disimpan

1	Pada 1 bulan sebelumnya tidak dilakukan pengukuran	TPG memulai pengukuran baru atau manual	Titik pengukuran baru pada grafik tidak memiliki garis penghubung dengan pengukuran sebelumnya
1	Pada 1 bulan sebelumnya, telah dilakukan pengukuran	TPG memulai pengukuran baru atau manual	Titik pengukuran baru pada grafik memiliki garis penghubung dengan pengukuran sebelumnya
1	Terdapat 2 atau lebih titik pengukuran pada rekam tumbuh kembang bayi/balita	TPG menghapus pengukuran terakhir	Pengukuran terakhir dihapus dari basisdata
5	Hanya terdapat 1 titik pengukuran yang merupakan berat dan panjang badan bayi sewaktu lahir	TPG menghapus pengukuran terakhir	Tidak terjadi perubahan keadaan pada program
5	Terdapat 1 atau lebih tindakan medis yang perlu dilakukan untuk rekam bayi/balita yang hendak dibuka oleh TPG	TPG membuka rekam bayi/balita	Notifikasi penindakan medis, ditampilkan
5	Tidak terdapat tindakan medis yang perlu dilakukan untuk rekam bayi/balita yang hendak	TPG membuka rekam bayi/balita	Notifikasi penindakan medis, tidak ditampilkan

	dibuka oleh TPG		
5	Bayi/balita belum dikenakan tindakan medis yang diwajibkan pada jadwal yang telah ditentukan	TPG melakukan <i>double click</i> pada baris tindakan medis	Tindakan medis dianggap telah dilakukan pada tanggal saat itu, dan riwayat tindakan medis diperbaharui
5	Bayi/balita sudah dikenakan tindakan medis yang diwajibkan pada jadwal yang telah ditentukan	TPG melakukan <i>double click</i> pada baris tindakan medis	Tidak terjadi perubahan keadaan pada program
6	Bayi masih berada dalam rentang umur 0-6 bulan	TPG membuka rekam bayi	<i>Checkbox</i> pemberian ASI Eksklusif ditampilkan
6	Bayi sudah berumur di atas 6 bulan	TPG membuka rekam bayi	<i>Checkbox</i> pemberian ASI Eksklusif tidak ditampilkan
6	<i>Checkbox</i> pemberian ASI Eksklusif ditampilkan	TPG menekan <i>checkbox</i> sehingga tercentang	Bayi dianggap masih diberikan ASI Eksklusif pada umur dilakukannya pengukuran
6	<i>Checkbox</i> pemberian ASI Eksklusif ditampilkan	TPG menekan <i>checkbox</i> sehingga tidak tercentang	Bayi dianggap tidak diberikan ASI Eksklusif pada umur dilakukannya pengukuran
7	TPG belum memilih bayi/balita yang hendak ditimbang	TPG menekan tombol <i>print</i>	Tidak terjadi perubahan keadaan pada program
7	TPG sudah memilih bayi/balita yang hendak ditimbang	TPG menekan tombol <i>print</i>	Mencetak KMS dalam rentang 10 titik pengukuran terbaru, menggunakan perangkat yang dipilih TPG

Dengan berakhirnya pengujian sistem pada iterasi pertama yang menunjukkan bahwa setiap kebutuhan *user story* pada iterasi pertama terpenuhi, pengembangan sistem dapat dilanjutkan pada iterasi kedua yang dibahas pada bab selanjutnya.

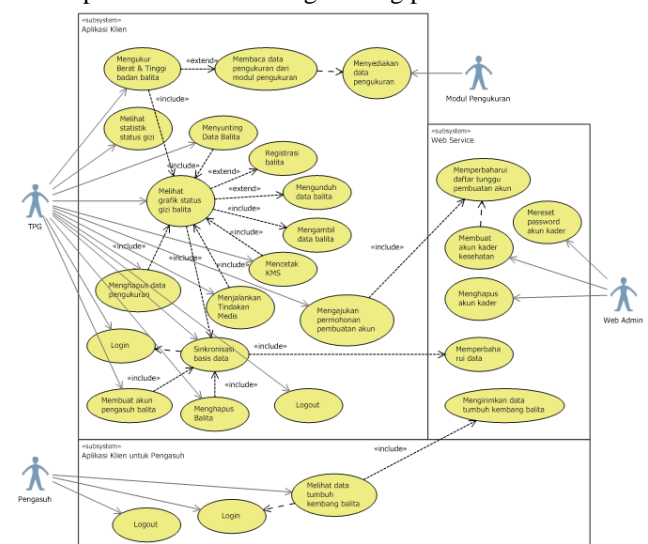
IV. TAHAPAN ITERASI 2

A. Perancangan

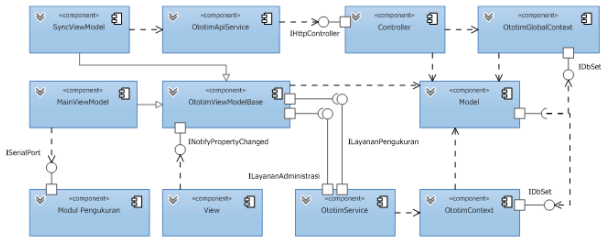
Iterasi kedua berfokus pada pengembangan sistem di sisi server. Dokumentasi perancangan meliputi *user story*, *use case*, *entity data model*, *activity diagram* dan *sequence diagram*. Platform perangkat lunak yang digunakan pada sisi server adalah *ASP.NET MVC Web Application*. *Design Pattern MVC* (Model View Controller) digunakan sebagai kerangka kerja pembuatan perangkat lunak yang ditulis dalam bahasa C#. Pembuatan perangkat lunak akan dibagi ke dalam beberapa lapisan, yang terdiri dari lapisan Model, View, Controller. Setelah setiap lapisan dibangun, langkah selanjutnya adalah pengujian menggunakan metode *User Acceptance Testing (UAT)*.

B. Desain Sistem

Dalam tahapan desain sistem, dihasilkan pemodelan sistem dengan standar pemodelan UML 2.0, berupa Use Case, EDM (Entity Data Model), Activity Diagram dan Component Diagram. Pada iterasi ke-2, terdapat perubahan pada *use case* yang telah dirancang sebelumnya karena 2 persona baru, yaitu Web Admin dan Pengasuh telah ditambahkan pada *product backlog*. Kedua persona tersebut memiliki kebutuhan yang perlu dimodelkan dalam *use case* oleh subsistem yang berhubungan langsung dengan kedua persona tersebut. Tentunya penambahan *use case* tersebut berimbas pada pembaharuan *use case* yang terasosiasi dengan persona TPG, karena diantara Web Admin, Pengasuh, dan TPG terdapat kolaborasi untuk mencapai kebutuhan masing-masing persona.



Gambar 15. Use Case dari Sistem yang akan Dibangun

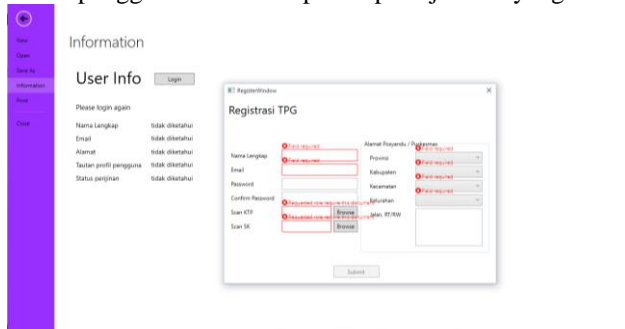


Gambar 16. Component diagram dari sistem yang akan dibangun

Pada iterasi kedua, pengembangan berfokus pada pembuatan *web service*. Terlihat pada *component diagram* diatas bahwa beberapa *component* telah ditambahkan. Pada aplikasi klien, terlihat penambahan komponen berupa *SyncViewModel* dan *OtotimApiService*. *SyncViewModel* bergantung pada komponen *OtotimApiService* untuk dapat mengonsumsi layanan yang diberikan *web service* melalui antarmuka *IHttpController* yang disediakan oleh komponen *Controller*. *Web service* menggunakan kerangka kerja MVC (*Model View Controller*) dimana setiap lapisan membentuk suatu komponen. Penggunaan ulang komponen *model* dapat mempersingkat waktu pengembangan dan untuk menjamin bahwa skema antara basisdata aplikasi klien dan basisdata pada *web service* memiliki kesamaan. Hal ini diperlukan untuk meningkatkan kompatibilitas data saat terjadi operasi *CRUD*.

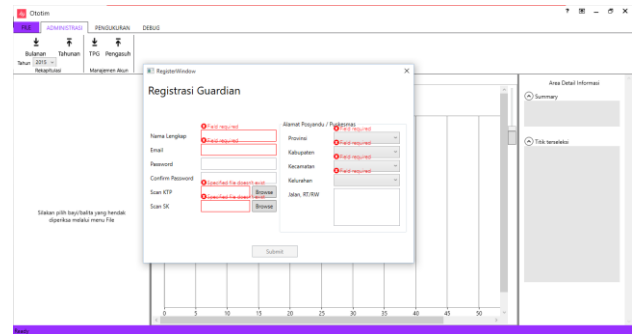
C. Implementasi

Gambar dibawah ini menunjukkan antarmuka yang digunakan untuk menjalankan aktifitas registrasi akun dengan *role* TPG. Terlihat bahwa untuk permohonan pembuatan akun dengan *role* TPG, terdapat isian untuk menyematkan dokumen KTP dan surat pernyataan legalitas TPG yang harus diberikan. Antarmuka dibawah ini dapat dimunculkan dengan menekan tombol *register* pada *view loginwindow*. Apabila proses pengiriman permohonan pembuatan akun berhasil maka informasi singkat mengenai rekam akun, seperti alamat email dan tautan pengguna akan ditampilkan pada jendela yang sama.



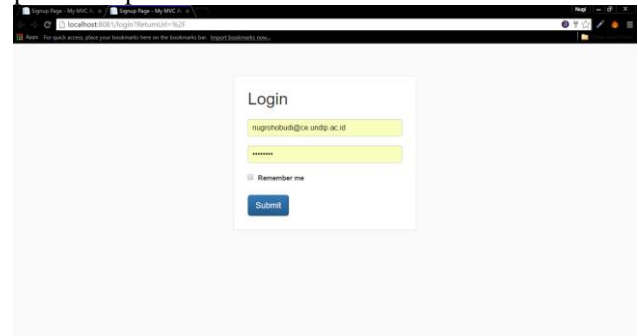
Gambar 17. Antarmuka registrasi TPG

Gambar dibawah ini menunjukkan antarmuka yang digunakan untuk menjalankan aktifitas registrasi akun dengan *role* Guardian yang dapat dimunculkan dengan menekan tombol 'Buat akun baru' pada *view ManageGuardianWindow*.



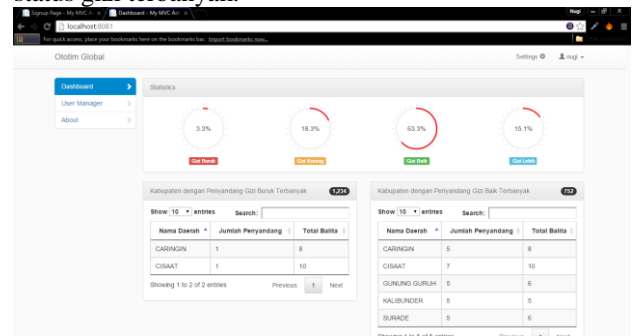
Gambar 18. Antarmuka registrasi pengasuh

Gambar dibawah ini menunjukkan antarmuka yang digunakan untuk menjalankan aktifitas login yang terjadi pada sisi aplikasi web.



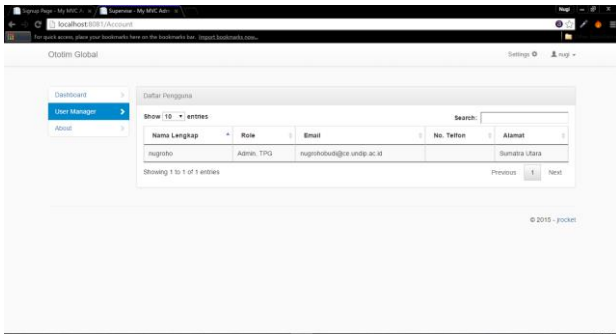
Gambar 19. Antarmuka login pada aplikasi web

Gambar dibawah ini menunjukkan antarmuka yang menampilkan data statistik status gizi. Pada konten baris pertama, data statistik berupa persentase penyandang untuk setiap jenis status gizi. Pada konten baris kedua, data statistik berupa daftar daerah dengan jumlah penyandang status gizi terbanyak.



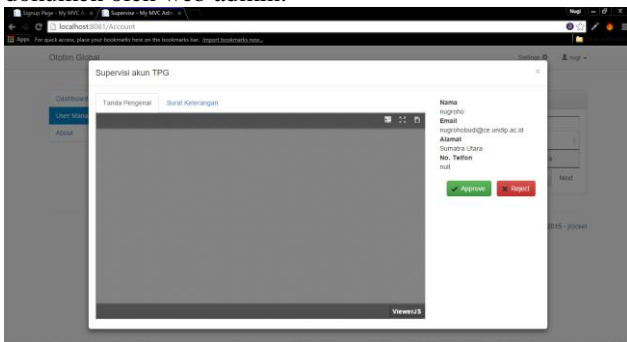
Gambar 20. Antarmuka beranda pada aplikasi web

Gambar dibawah ini menunjukkan antarmuka yang digunakan untuk menampilkan daftar pengguna aplikasi web. Apabila salah satu rekam pengguna aplikasi web dengan *role* TPG ditekan maka dokumen yang terkait dengan akun tersebut akan ditampilkan melalui *modal dialog*. Antarmuka ini hanya akan muncul jika pengguna yang telah login memiliki *role* web admin.



Gambar 21. Antarmuka daftar pengguna pada aplikasi web

Gambar dibawah ini menunjukkan antarmuka yang digunakan untuk menampilkan dokumen yang terkait dengan rekam akun TPG yang diseleksi pada antarmuka sebelumnya. Pada antarmuka ini juga terjadi bagian dari aktifitas registrasi akun, yang berupa proses verifikasi dokumen oleh web admin.



Gambar 22. Antarmuka verifikasi dokumen pada aplikasi web

Jika tombol reject ditekan maka akun yang bersangkutan akan dihapus dari basis data, dan pemohon pembuatan akun tersebut dianggap tidak memenuhi persyaratan.

D. Pengujian

Integrasi dilakukan dengan mengarahkan *url* dari setiap *method* yang digunakan oleh kelas *OtotimApiService*, ke setiap *action* pada kelas *controller* yang terdapat pada aplikasi web sesuai dengan fungsi yang diekspektasikan. Pengujian dilakukan menggunakan metode *User Acceptance Testing (UAT)*. Semua *user story* yang didefinisikan pada sub bab sebelumnya akan dijadikan acuan dalam membentuk *test case* yang tepat untuk mengetahui apakah kebutuhan *user* telah terpenuhi.

Tabel 7. UAT iterasi 2

Id User Story	Diketahui	Ketika	Maka
10	Web Admin belum memberikan otoritas kepada TPG untuk memperbaharui data gizi di <i>server</i>	TPG memperbaharui data gizi di <i>server</i>	Tidak terjadi pembaharuan data gizi di <i>server</i>

10	Web Admin telah memberikan otoritas kepada TPG untuk memperbaharui data gizi di <i>server</i>	TPG memperbaharui data gizi di <i>server</i>	Terjadi pembaharuan data gizi di <i>server</i>
11	Pengguna telah diberikan otoritas oleh Web Admin untuk mengakses sumber daya server, Web Admin menghapus akun pengguna	Pengguna meminta aplikasi web untuk mengirimkan sumber daya pada <i>server</i>	Aplikasi web tidak memberikan sumber daya yang diminta oleh pengguna
12	Tidak terdapat kesalahan data yang diberikan oleh pengguna pada <i>form</i> permohonan	TPG mengirimkan <i>form</i> permohonan pembuatan akun untuk mengakses sumber daya di <i>server</i>	Akun TPG tersimpan pada basis data di <i>server</i>
12	Terdapat kesalahan data yang diberikan oleh pengguna pada <i>form</i> permohonan	TPG mengirimkan <i>form</i> permohonan pembuatan akun untuk mengakses sumber daya di <i>server</i>	Akun TPG tidak tersimpan pada basis data di <i>server</i>
13	TPG belum mendapatkan otoritas dari Web Admin	TPG mengunggah data gizi yang terdapat pada basis data lokal ke dalam basis data pusat	Data gizi tidak tersimpan pada basis data pusat

Dengan berakhirnya pengujian sistem pada iterasi kedua yang menunjukkan bahwa setiap kebutuhan *user story* pada iterasi kedua terpenuhi, maka seluruh kebutuhan yang tercantum pada *product backlog* berhasil dipenuhi.

V. PENUTUP

A. Kesimpulan

Dari hasil pengujian dan analisis Sistem Informasi Antropometri dapat disimpulkan hal-hal sebagai berikut:

1. Dengan melihat pada dinamika pembuatan sistem, yang terus berkembang tanpa perlu dilakukan perombakan secara masif pada program yang telah

dibuat sebelumnya, penggunaan *agile* sebagai metodologi pengembangan dirasa tepat untuk memberikan gambaran umum terhadap perubahan-perubahan yang perlu diimplementasi secara inkremental sesuai dengan skala prioritas kebutuhan pengguna

2. Meskipun *agile* lebih banyak diperuntukkan sebagai metodologi pengembangan untuk sebuah tim, beberapa komponen *agile* masih dapat diterapkan untuk pengembangan sistem yang dilakukan oleh satu orang, seperti penggunaan *user story* sebagai acuan pengembangan, penerapan iterasi sebagai bentuk disiplin waktu kerja untuk dapat menghasilkan produk jadi dengan cepat, dan pelibatan pengguna dalam setiap tahapan pengembangan agar *feedback* dapat ditanggapi dengan cepat dalam bentuk *user acceptance testing* yang penting dalam proses evaluasi dan refaktorisasi senarai pada sistem yang dibangun.
3. Pembuatan sistem informasi antropometri dapat digunakan untuk membantu proses penimbangan dan pengawasan tumbuh kembang balita.
4. Pembuatan sistem terintegrasi yang dinamis sangat diperlukan sehingga ketika membuat suatu aplikasi terintegrasi, aplikasi tersebut dapat diimplementasikan diberbagai sistem yang berbeda tanpa harus melakukan perubahan secara drastis terhadap sistem yang sudah ada.
5. Pembuatan sistem informasi terintegrasi mampu memberikan analisa data statistik yang cepat dan akurat, dengan tingkat varian pengambilan data yang luas
6. Sistem terintegrasi bermanfaat dalam membantu proses perekapan data status gizi, yang memberikan efisiensi waktu karena pengubahan data pada semua sistem secara *realtime*.
7. Sistem terintegrasi memberikan efektifitas kerja kepada TPG karena dengan data yang terintegrasi, informasi yang dibutuhkan oleh TPG dapat langsung dilihat.

B. Saran

Saran yang diberikan dalam upaya pengembangan aplikasi yang lebih baik dikemudian hari.

1. Penambahan fitur kirim data tumbuh kembang balita ke perangkat bergerak pengasuh melalui NFC.
2. Penambahan fitur obrolan dan pesan untuk memberikan kemudahan dalam melakukan komunikasi dari TPG dan Pusat.
3. Penerapan sistem pada perangkat yang lebih terjangkau bagi TPG, seperti *platform raspberry pi*.
4. Penambahan fitur untuk membuat varian pengambilan data baru agar dapat mengadaptasi proses pelaporan data status gizi yang dibutuhkan pusat secara dinamis

DAFTAR PUSTAKA

- [1] Fadjar, E., "Prevalensi Gizi Buruk Balita Meningkat di 2014" [Online].

- Available:<http://www.tempo.co/read/news/2015/01/25/174637469/Prevalensi-Gizi-Buruk-Balita-Meningkat-di-2014/>. [Accessed: 24 Februari 2015].
- [2] Peraturan Menteri Kesehatan Republik Indonesia Nomor 155/Menkes/Per/I/2010 Tentang Kartu Menuju Sehat (KMS) bagi Balita.
 - [3] Zahraini, Y., "Bagaimana Penerapan Standar Antropometri Penilaian Status Gizi Balita pada TPG Terlatih?" [Online]. Available: <http://gizi.depkes.go.id/bagaimana-penerapan-standar-antropometri-penilaian-status-gizi-balita-pada-tpg-terlatih/>. [Accessed: 29 Juni 2015]
 - [4] Kuntari, T., "Mengoptimalkan KMS untuk Mengurangi Gizi Buruk pada Balita" [Online]. Available: http://kuliah.fkuii.otg/index.php?option=com_content&view=article&id=96:mengoptimalkan-fungsi-kms-untuk-mengurangi-gizi-buruk-pada-balita&catid=87&Itemid=522/. [Accessed: 24 Februari 2015].
 - [5] Petolawal, M., Chaeruddin, "Hubungan Pemanfaatan KMS dan Pengetahuan Ibu dengan Tumbuh Kembang Anak Balita di Desa Bateleme Kec. Lembo Kab. Morowali," *J. Ilmiah Kesehatan.*, vol. 1, no. 4, 2012.
 - [6] Priskila, O. dan Wibowo A., "Efektifitas Kartu Menuju Sehat (KMS) Elektronik untuk Meningkatkan Kecepatan Pelayanan, Mempermudah Pendataan dan Pengambilan Keputusan Status Kesehatan di Posyandu," *J. Biokep.*, vol. 2, no. 1, 2013.
 - [7] Widiyanto, E. D. dan Eridani, D., "Simulasi Aplikasi Posyandu berdasarkan Konsep RFID (Radio Frequency Identification)," *J. Sistem Komputer.*, vol. 4, no. 2, 2014.
 - [8] Sandjaja, "Kamus Gizi: pelengkap kesehatan keluarga". Penerbit Buku Kompas, Jakarta, 2009.
 - [9] Supariasa, I. D., "Penilaian status gizi", EGC, Jakarta, 2001.
 - [10] Barasi, M., "Nutrition at a Glance", Penerjemah: Hermin. 2009, "At a Glance: Ilmu Gizi", Erlangga, Jakarta, 2007.
 - [11] Almatsier, S., "Prinsip dasar ilmu gizi", PT. Gramed Pustaka Utama, Jakarta, 2004.
 - [12] Jahari, A. B. dan Sumarno, I., "Status Gizi Penduduk Indonesia", *Majalah Pangan* No.38/XI/Jan/2002.
 - [13] Gibson, R. S., Ferguson, E. L. dan Lehrfeld, J., "Complementary foods for infant feeding in developing countries : their nutrient adequacy and improvement", 1998
 - [14] Soekirman, "Ilmu gizi dan aplikasinya", Dirjen Pendidik Tinggi Departemen Pendidikan Nasional, Jakarta, 2000.
 - [15] Andrade, C., Livermore, S., Meyers, M. dan Vliet, S. V., "Professional WPF Programming: .NET Development with the Windows® Presentation Foundation", Wiley Publishing Inc., Indiana, 2007.
 - [16] MSDN Library, "Introduction to Windows Presentation Foundation" [Online]. Available:

- <http://msdn.microsoft.com/en-us/library/aa970268.aspx>. [Accessed: 29 Juni 2015].
- [17] MacDonald, M., "Pro Silverlight 2 in C#", Apress, New York, 2009.
- [18] Horn, S., "Microsoft Silverlight 3: A Beginner's Guide", McGraw-Hill Companies, New York, 2010.
- [19] Smith, J., "WPF Apps with the Model-View-View-View-View-View Design Pattern", MSDN Magazine, 2009.
- [20] Fowler, M., "Presentation Model" [Online]. Available: <http://martinfowler.com/eaDev/PresentationModel.html>. [Accessed: 29 Juni 2015].
- [21] Google Groups, "Thought: MVVM eliminates 99% of the need for ValueConverters" [Online]. Available: https://groups.google.com/forum/#!topic/wpf-disciples/P-JwzRB_GE8. [Accessed: 29 Juni 2015].
- [22] MSDN Library, "ASP.NET Web API Reference" [Online]. Available: [https://msdn.microsoft.com/en-us/library/hh849329\(v=vs.108\).aspx](https://msdn.microsoft.com/en-us/library/hh849329(v=vs.108).aspx). [Accessed: 30 Juni 2015]
- [23] Wojcieszyn, F., "Web API 2 Recipes", Apress, New York, 2014
- [24] Mikrolima, "ORM (Object Relational Mapping)" [Online]. Available: <https://mikrolima.wordpress.com/2010/02/22/orm-object-relational-mapping/>. [Accessed: 30 Juni 2015]
- [25] Kurniawan, A., "Mengenai Entity Framework 4.0" [Online]. Available: <http://blog.aguskurniawan.net/post/Mengenai-Entity-Framework-40.aspx>. [Accessed: 30 Juni 2015]
- [26] Yanti, R., "Cara Praktis Menentukan Status Gizi Anak" [Online]. Available: <https://proyekruspita.wordpress.com/tag/indeks-antropometri/>. [Accessed: 30 Juni 2015]
- [27] Pustekkom Depdiknas, "Kelebihan dan Kelemahan USB" [Online]. Available: <http://idkf.bogor.net/yuesbi/e-DU.KU/edukasi.net/Elektro/USB/materi3.html>. [Accessed: 9 Juli 2015]
- [28] Proboyekti, U. "Bahan Ajar Rekayasa Perangkat Lunak Agile Software Development", Indonesia, 2008.
- [29] Elliott, G., "Global Business Information Technology: an integrated systems approach", Pearson Education, New York, 2004.
- [30] Jeffrey, L. W., Bentley, L. D., Dittman, K. C., "Systems Analysis and Design Methods 6th edition", Addison-Wesley, USA, 2003.
- [31] Highsmith, J., "History: The Agile Manifesto" [Online]. Available: <http://www.agilemanifesto.org>. [Accessed: 29 Juni 2015]
- [32] Collier, K., "Agile Analytics: A Value-Driven Approach to Business Intelligence and Data Warehousing", Addison-Wesley, USA, 2011.
- [33] Silverburg, A., "Agile Analytics in Higher Education", Phytion, USA, 2012.
- [34] Kniberg, H., "Scrum and XP Practice", C4Media, USA, 2007.
- [35] MSDN Library, "XAML Overview" [Online]. Available: [https://msdn.microsoft.com/enus/library/vstudio/ms752059\(v=vs.100\).aspx](https://msdn.microsoft.com/enus/library/vstudio/ms752059(v=vs.100).aspx). [Accessed: 29 Juni 2015].
- [36] Pugh, K., "Lean-Agile Acceptance Test-Driven Development: Better Software Through Collaboration", Addison-Wesley, USA, 2011.
- [37] Adzic, G., "Bridging the Communication Gap: Specification by Example and Agile Acceptance Testing", Neuri Limited, Britania Raya, 2009.
- [38] Adzic, G., "Specification by example: How successful teams deliver the right software", Manning, New York, 2011.
- [39] Chelimsky, D., Dave, A., Zach, D., Aslak, H., Bryan, H., Dan, N., "The RSpec Book: Behaviour Driven Development with RSpec, Cucumber, and Friends", The Pragmatic Bookshelf, North Caroline, 2011.