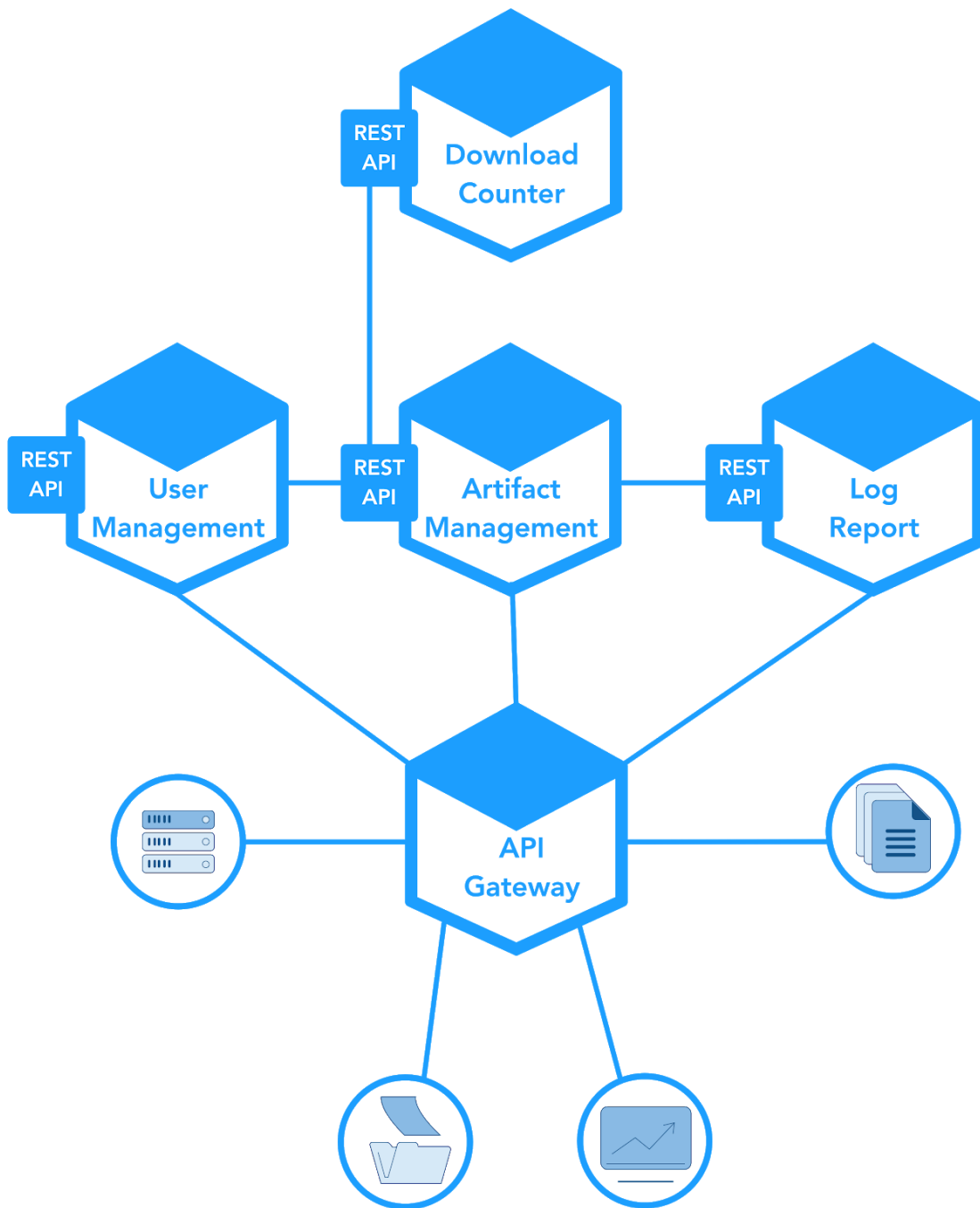


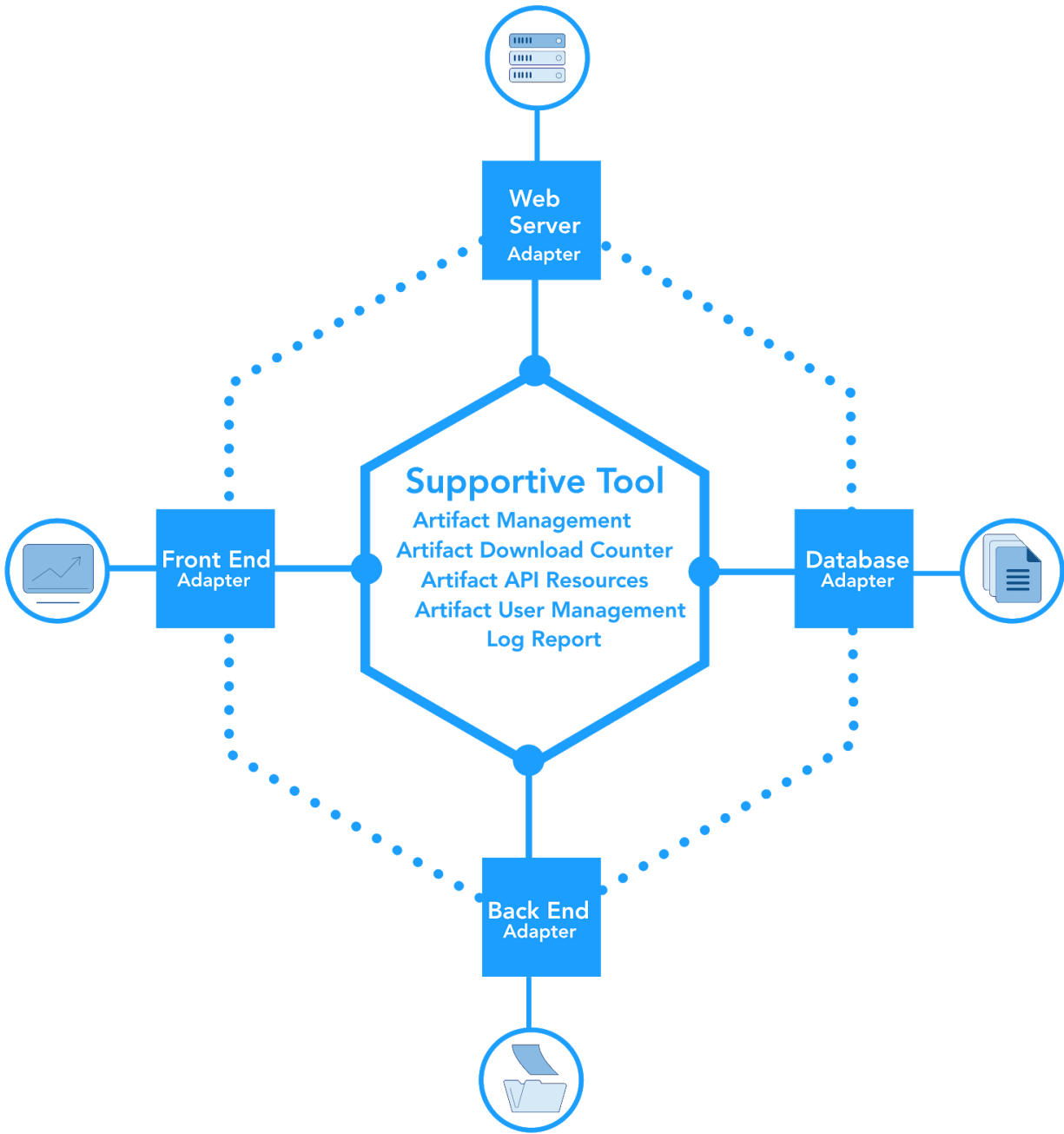


## Pengembangan Lecturers Supportive Tool Berbasis Teknologi Web Menggunakan Micro Framework dan Microservice Architecture (MSA)

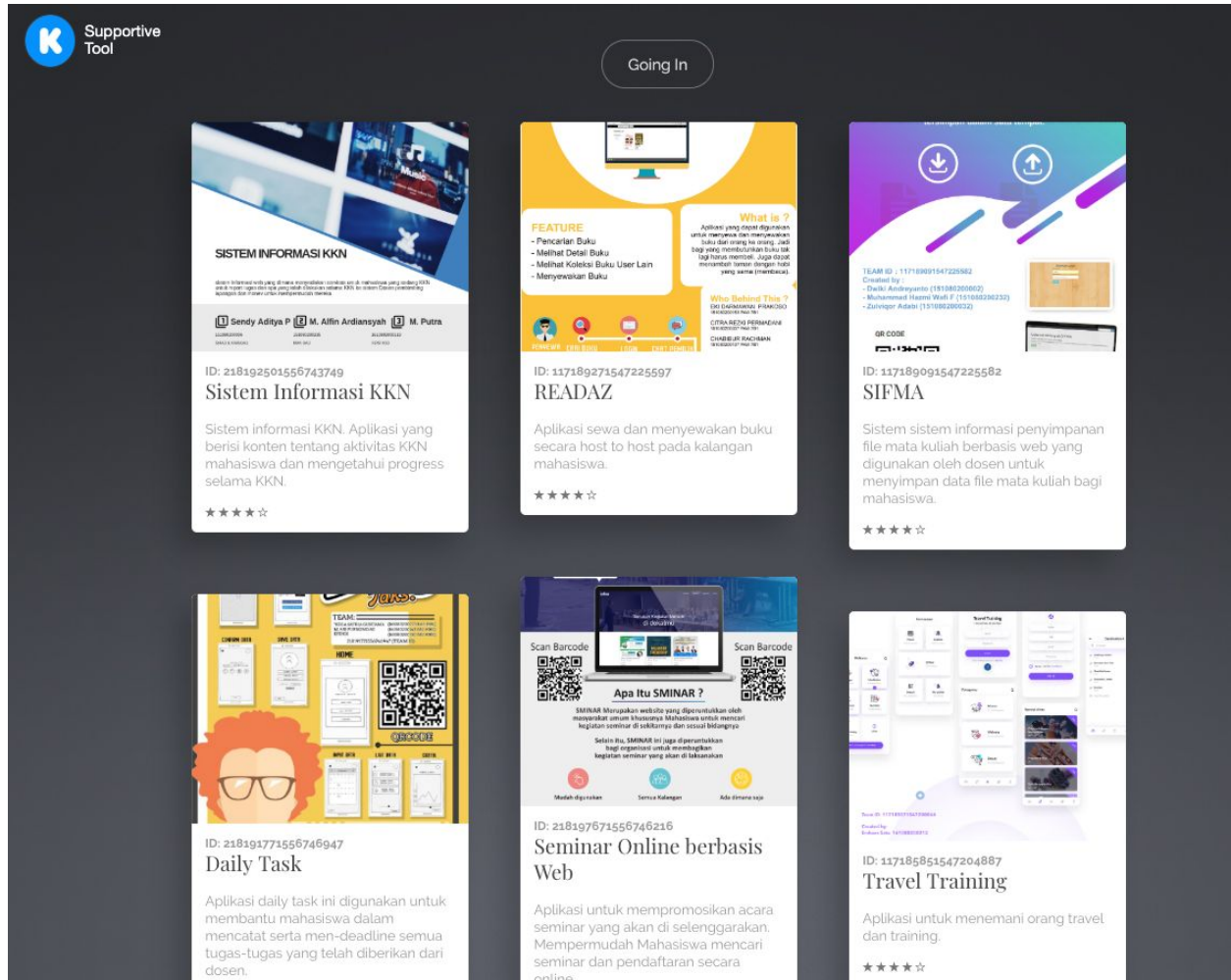
Platform Lecturers Supportive Tool merupakan platform yang berfungsi dan didesain untuk men-support dosen (lecturers) dalam rangka mengumpulkan karya akademiknya. Dengan mengembangkan platform tersebut dengan teknologi web, dosen sebagai pengguna dapat dilayani dengan kemudahan penggunaan dan interoperabilitas data/dokumen/artefak hasil kegiatan Tri Dharma Perguruan Tinggi. Pengembangan teknologi web, pada dasarnya membutuhkan sebuah Web Server. Untuk mengkonfigurasi web server dan file server, diperlukan keahlian khusus dalam bidang komputer yang tidak semua dosen memilikinya. Untuk itu, pemanfaatan kerangka kerja (*framework*) yang telah mengikutsertakan modul Web Server, sangatlah berguna untuk masalah tersebut. Untuk itu, dapat dipilih Flask Microframework yang dalam pemasangannya sudah terdapat modul Werkzeug, yang berfungsi sebagai web server. Disisi lain, kebutuhan akan data dosen yang terdistribusi, dapat menggunakan metode *Microservice Architecture*. Dengan menggunakan *Microservice Architecture*, platform Lecturers Supportive Tool dikembangkan dengan mentransformasi fitur-fitur yang dibutuhkan menjadi sebuah *service-service* yang independen. Dengan layanan yang independen, distribusi dan desentralisasi dokumen-dokumen/artefak dosen mempermudah akses terhadap karya-karya akademis/intelektual civitas akademik kepada masyarakat. Dalam pengembangan supportive tool ini, terdapat dua macam tipe platform. Yaitu Platform **SERVER** dan Platform **NODE**. Untuk Platform Server, dikembangkan dengan arsitektur *Microservice*. Dimana di dalam mesin server, terdapat beberapa *service* yang dijalankan secara terpisah dimana setiap *service* tersebut telah ditentukan fungsi layanannya melalui *Representational state transfer (REST) API*. Sedangkan untuk Platform Node, dikembangkan dengan arsitektur *monolithic* dengan *database engine* berbasis file seperti SQLite.



Gambar 1. Microservice Arsitektur Supportive Tool Tipe SERVER.



Gambar 2. Monolithic Arsitektur Supportive Tool Tipe NODE.



Gambar 3. Dokumen-dokumen karya akademik yang siap didistribusikan

```

@app.route('/aplikasi')
def aplikasi():
    uip = getIP()
    if 'username' in session:
        level=["1","10"]
        if session['level_akses'] in level:
            user=session['username']
            kk=getKodeKelompok(str(user))
            cur = mysql.connection.cursor()
            cur.execute("SELECT * FROM aplikasi WHERE kodekelompok=%s",[kk])
            rows = cur.fetchall()
            isiLog("aplikasi", user, uip)
            return render_template("aplikasi.html",user=user, level_akses=session["level_akses"], rows=rows)
        else:
            isiLog("aplikasi", "INTRUDER!", uip)
            return redirect(url_for('index'))
    else:
        isiLog("aplikasi", "INTRUDER", uip)
        return redirect(url_for('index'))

```

Gambar 4. Snapshot script Microservice Modul Artifact Management

```

@app.route('/kl')
def kl():
    if 'username' in session:
        level=["10"]
        if session['level_akses'] in level:
            user = session['username']
            cur = mysql.connection.cursor()
            sql = """SELECT kelompok.`kodekelompok`,aplikasi.`namaaplikasi`, aplikasi.`poster`, \
            aplikasi.`presentasi`, aplikasi.`laporan`, aplikasi.`nilaiangka`, aplikasi.`nilaihuruf` \
            FROM kelompok INNER JOIN aplikasi ON kelompok.`kodekelompok` = aplikasi.`kodekelompok` GROUP BY kelompok.`kodekelompok`"""
            rows = cur.fetchall()

            cur2 = mysql.connection.cursor()
            sql2 = "SELECT mahasiswa.`nim`, mahasiswa.`nama`, mahasiswa.`kodekelompok`, mahasiswa.`kelas` \
            FROM kelompok INNER JOIN mahasiswa ON kelompok.`kodekelompok` = mahasiswa.`kodekelompok`"
            rows2 = cur2.fetchall()

            return render_template('kelompok-list-all-2.html', rows=rows, rows2=rows2, user=user)
        else:
            return redirect(url_for('home'))
    else:
        return redirect(url_for('index'))

```

Gambar 5. Snapshot script Microservice Modul User Management

```

@app.route('/kl-save/<kk>', methods=['GET', 'POST'])
def klsave(kk):
    if 'username' in session:
        level=["10"]
        if session['level_akses'] in level:
            user = session['username']
            n_huruf = request.form['nilai']
            n_angka = getNilaiAngka(n_huruf)

            cur = mysql.connection.cursor()

            sql_aplikasi = """
                UPDATE aplikasi
                SET
                nilaiangka=%s,
                nilaihuruf=%s

                WHERE kodekelompok = %s
            """

            cur.execute(sql_aplikasi, [n_angka,n_huruf,kk])
            mysql.connection.commit()

            sql_mahasiswa = """
                UPDATE mahasiswa
                SET
                nilaiangka=%s,
                nilaihuruf=%s

                WHERE kodekelompok = %s
            """
            cur.execute(sql_mahasiswa, [n_angka,n_huruf,kk])
            mysql.connection.commit()
            return redirect(url_for('kl'))
        else:
            return redirect(url_for('home'))
    else:
        return redirect(url_for('index'))

```

Gambar 6. Snapshot script Microservice Modul Penilaian Dokumen

```

class BaseModel(Model):
    class Meta:
        database = database

class Users(BaseModel):
    email = CharField(unique=True)
    password = TextField()
    email = TextField()
    join_date = TextField()

class TabeLog(BaseModel):
    log_id = TextField(unique=True)
    activity = TextField()
    time_epoch = TextField()
    time_stamps = TextField()
    user_log = TextField()
    user_ip = TextField()

def create_tables():
    with database:
        database.create_tables([Users, TabeLog])

```

Gambar 7. Snapshot script Object DB Model Microservice Modul Log Report

```

def isiLog(actname, userlog, uip):
    vepoch = time.time()
    vtimeepoch = str(int(vepoch))
    vtimestamps = str(time.strftime('%d-%m-%Y %H:%M:%S', time.localtime(vepoch)))
    lid = getCountLogId()
    car_entry = TabeLog.create(
        log_id = lid,
        activity= actname,
        time_epoch=vtimeepoch,
        time_stamps = vtimestamps,
        user_log = userlog,
        user_ip = uip
    )
    return True

```

Gambar 8. Snapshot script Entry Log pada Microservice Modul Log Report



Show 10 entries

#	↑↓ Activities	↑↓ Time	↑↓ IP Address
1	/home	26-08-2019 12:03:07	139.228.41.226
2	login	26-08-2019 12:03:07	139.228.41.226
3	Log Out	26-08-2019 09:43:18	139.228.41.226
4	/home	26-08-2019 09:43:15	139.228.41.226
5	/home	26-08-2019 09:37:27	139.228.41.226
6	/home	26-08-2019 09:35:07	139.228.41.226
7	login	26-08-2019 09:35:07	139.228.41.226
8	Log Out	26-08-2019 09:34:52	139.228.41.226
9	/home	26-08-2019 09:34:49	139.228.41.226
10	/home	26-08-2019 09:30:46	139.228.41.226
#	Activities	Time	IP Address

Gambar 9. Output Microservice Modul Log Report

## Progress Report

#	Pekan	Kegiatan
1	Pekan ke 8	Sebelumnya kami diskusi, mengenai bagaimana nanti sistem pada aplikasi ini berjalan.. Membuat kerangka gambar alur sistem pada buku tulis. Membagi tugas membuat logo, use case diagram
2	Pekan ke 9	Kita berkumpul di kantin membahas user centered diagram dan berdiskusi.. lalu membahas bagaimana konsep untuk poster kita
3	Pekan ke 10	Berdiskusi terkait logo/maskot yang akan kita gunakan ke aplikasi E-Warung
4	Pekan ke 11	MEng fixkan logo untuk kita dan menyelesaikan latar belakang utk laporan
5	Pekan ke 12	Mencoba koding python utk maps
6	Pekan ke 13	Merevisi Use case diagram, dan merapikan laporan tugas akhir
7	Pekan ke 14	membuka situs konro yg sdah work kmbli utk dicopy qrcode diposter
8	Pekan ke 8	Link youtube : <a href="https://youtu.be/mvUlqT4dmTY">https://youtu.be/mvUlqT4dmTY</a>
9	Pekan ke 14	Link youtube : <a href="https://youtu.be/mvUlqT4dmTY">https://youtu.be/mvUlqT4dmTY</a>

Gambar 10. Output Microservice Modul Progress Report