DAFTAR PUSTAKA

[1] Aswin dan T. Surungan. "Aplikasi Metoda Wang-Landau pada Studi Perubahan Fase Model Ising Dua Dimensi dengan Interaksi Ekstra". *Seminar Nasional Fisika Makassar*, hal. 41-44, Makassar, 10 Oktober 2015.

[2] T. Surungan. *Fisika Statistik*. Diktat, Departemen Fisika Unhas, Makassar, 2013.

[3] M. Jufrin, T. Surungan dan Bangsawang. BJ. "Sifat Kritis Model Magnetik Simetri Polyhedral Pada Kisi Segitiga". 2010.

[4] T. Surungan, Bangsawang. BJ dan M. Yusuf. "Critical properties of the antiferromagnetic Ising model on rewired square lattices". *IOP Conf. Series: Journal of Physics: Conf. Series* 1011 012079, 2018.

[5] T. Surungan, Bangsawang. BJ and M. Yusuf. "Critical properties of the antiferromagnetic Ising model on rewired square lattices". *IOP Conf. Series: Journal of Physics: Conf. Series* 979. 012086, 2018.

[6] S. Sivasankaran, P. K. Nayak and E. Gunay. *Solid State Physics: Metastable, Spintronics Materials and Mechanics of Deformable Bodies-Recent Progress*. IntechOpen, London, 2020.

[7] B. A. Cipra. "An Introduction to the Ising Model". *The American Mathematical Monthly*, Vol. 94, No. 10, 1987.

[8] M. Jufrin. *Sifat Kritis Model Magnetik Simetri Polyhedral Pada Kisi Segitiga*. Skripsi, Departemen Fisika Unhas, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Hasanuddin, Makassar, 2015.

[9] F. W. Sears and G. L. Salinger. *Thermodynamics, Kinetic Theory, and Statistical Thermodynamics*. Addision-Wesley Publishing Company, Inc, Philippines, 1975.

[10] L. E. Reichl. *A Modern Course in Statistical Physics*.Willey-VCH Verlag GmbH & Co. KGaA, Germany, 2016.

[11] F. Schwabl. *Statistical Mechanics.* Springer-Verlag, Berlin, Germany, 2006.

[12] T. Surungan. “Phase diagram of six-state clock model on rewired square lattices”. *IOP Conf. Series: Journal of Physics: Conf. Series* 1290 012028, 2019.

[13] J. M. Yeomans. *Statistical Mechanics of Phase Transitions.* Clarendon Press, Oxford, 1992.

[14] H. Nishimori. *Elements of Phase Transitions and Critical Phenomena.* Oxford University Press, New York, 2017.

[15] D. J. Griffith. *Introduction to Electrodynamics.* Cambridge University Press, New York, 1992.

[16] A. A. C. Tenribali. *Studi Sifat Kritis Model Spin XY Menggunakan Algoritma Metropolis Dengan Spin Update Berbasis Parameter Temperatur*. Skripsi, Departemen Fisika Unhas, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Hasanuddin, Makassar, 2020.

[17] S. Blundell. *Magnetism in Condensed Matter.* Oxford University Press, New York, 2001.

[18] T. Surungan. “Search for the Heisenberg Spin Glass on Rewired Square Lattices with Antiferromagnetic Interaction”. *AIP Conference Proceedings, 1719 030006, 2016.*

[19] D. P. Landau and K. Binder. *A Guide to Monte Carlo Simulation in Statistical Physics.* Cambridge University Press, New York, 2000.

[20] J. M. Kosterlitz and D. J. Thouless. “Ordering, Metasbility And Phase Transition In Two-Dimensional System”. *Journal of Physics C: Solid State Physics*, 6, 1181, 1973.

[21] J. M. Kosterlitz. “The Crirical Properties of the Two-Dimensional XY Model”. *Journal of Physics C: Solid State Physics*, 7, 1046, 1974.

[22] T. Surungan , Y. Komura, dan Y. Okabe. “Probing Phase Trantition Order Of Q-State Potts Models Using Wang-Landau Algorithm”. *Aip Conference Proceeding*, 2014.

[23] M. Abdullah. *MEKANIKA STATISTIK.* Institut Teknologi Bandung, 2017.

LAMPIRAN A

Program Utama

1. Algoritma Wang-Landau

```
## Programmed by KH 2010/2/28 Wang-Landau algorithm for 2D Clock model
    from scipy import *
    # from scipy import weave
    import sys
    from pylab import *
    # from numba import jit
    # from openpyxl import *

    def kisi(N,teta):              #Membuat matrik Random
        a = np.zeros((N,N))
        for i in range(N):
            for j in range(N):
                a[i,j] = np.random.choice(teta)
        return a

    def CEnergy(latt):             #Menghitung Hamiltonian sistem
        Ene = 0
        for i in range(N):
            for j in range(N):
                S = latt[i,j]
                WF = latt[(i+1)%N, j] + latt[i,(j+1)%N] + latt[(i-1)%N,j] + latt[i,(j-
1)%N]
                Ene += -WF*S
        return int(Ene/2.)

    def Thermod(T, lngE, Energies, E0):        #Ekstrak energi & panas jenis
        Z=0
```

```
    Ev=0
    E2v=0
    for i,E in enumerate(Energies):
        w = exp(lngE[i]-lngE[0] - (E+E0)/T)
        Z   += w
        Ev  += w * E
        E2v += w * E**2
    Ev *= 1./Z
    cv = (E2v/Z - Ev**2)/T**2
    return (Ev/N2, cv/N2)

def SamplePython(Nitt, N, N2, indE, E0, flatness):
    "Wang Landau algorithm in Python"
    latt = kisi(N,teta)
    Ene = CEnergy(latt)
    lngE = np.zeros(len(Energies), dtype=float)+1
    Hist = np.zeros(len(Energies), dtype=float)

    lnf = 1.0
    for itt in range(Nitt):
        ii = int(np.random.rand() * N2)
        (i, j) = (ii % N, ii // N)
        S = latt[i, j]
        WF = latt[(i + 1) % N, j] + latt[i, (j + 1) % N] + latt[(i - 1) % N, j] + latt[i,
(j - 1) % N]
        Enew = int(Ene + 4 * S * WF)

        P = np.exp(lngE[indE[Ene + E0]] - lngE[indE[Enew + E0]])  #
Probability to accept according to Wang-Landau
        if P > np.random.rand():
            #latt[i, j] = -S
```

```
            latt[i, j] = np.random.choice(teta)
            Ene = Enew

        Hist[indE[Ene + E0]] += 1.
        lngE[indE[Ene + E0]] += lnf
        if itt % 100 == 0:
            aH = np.sum(Hist) / (N2 + 0.0)
            mH = min(Hist)
            if mH > aH * flatness:
                Hist = np.zeros(len(Hist))
                lnf /= 2.
                print(itt, 'histogram is flatt', mH, aH, 'f=', np.exp(lnf))
    return (lngE, Hist)

if __name__ == '__main__':
    iterasi = 50000
    while iterasi < 70000 :
        try:
            print(iterasi)
            Nitt =4000
            N = 8
            flatness = 0.9
            N2 = N * N
            NTe = N2 - 1
            teta = [1, 0.5, -0.5, -1, -0.5, 0.5]

            # Possible energies of the Ising model
            Energies = (np.arange(4*N2+1)-2*N2).tolist()
            # Energies.pop(1)
            # Energies.pop(-2)
```

```
        # Maximum energy
        E0 = Energies[-1]
        #     Index array which will give us position in the Histogram array from knowing the Energy
        indE = -np.ones(E0 * 2 + 1, dtype=int)
        for i, E in enumerate(Energies):
            indE[E + E0] = i

        (lngE, Hist) = SamplePython(Nitt, N, N2, indE, E0, flatness)

        # Normalize the density of states, knowing that the lowest energy state is double degenerate
        # lgC = log( (exp(lngE[0])+exp(lngE[-1]))/4. )
        if lngE[-1]<lngE[0]:
            lgC = lngE[0] + log(1+ exp(lngE[-1]-lngE[0])) - log(4.)
        else:
            lgC = lngE[-1] + log(1+ exp(lngE[0]-lngE[-1])) - log(4.)
        lngE -= lgC
        for i in range(len(lngE)):
            if lngE[i]<0: lngE[i]=0

        # Normalize the histogram

        Hist *= len(Hist) / float(np.sum(Hist))

        #Save to csv
        csv_lngE = [Energies, lngE]
        a = list(zip(*csv_lngE))
        np.savetxt(f"./17_agustus/minesrandom/lnge_{iterasi}.csv",          a, delimiter="    ", comments="#Energies    lngE", fmt='% s')
```

```
            iterasi += 1
        except :
            pass

    # csv_Hist = [Energies, Hist]
    # b = list(zip(*csv_Hist))
    #     np.savetxt("./csv/csv_Hist.csv",     b,     delimiter="                ",
comments="#Energies    Hist", fmt='% s')

    Te = linspace(0.0, 5., NTe)
    Thm = []
    for T in Te:
        Thm.append(Thermod(T, lngE, Energies, E0))
    Thm = array(Thm)
```

2. Menghitung ekstrak Energi (E) dan Panas Jenis (CV)

```
import numpy as np
import matplotlib.pyplot as plt

def get_energies_and_lnge(kisi):
    energies = []
    lngE = []
    a = np.genfromtxt(f"./kisi{kisi}.csv", delimiter="   ").tolist()
    # print(a)
    for j in range(len(a)):
        energies.append(a[j][0])
        lngE.append(a[j][1])
    return energies,lngE

def Thermod(T, lngE, energies, E0):
    "Thermodynamics using density of states"
```

```
    Z_fero = 0
    Ev_fero = 0
    E2v_fero = 0
    Z_antifero = 0
    Ev_antifero = 0
    E2v_antifero = 0
    for i, E in enumerate(energies):
        E /= 2
        w_fero    = np.exp(lngE[i] - (E/T))
        Z_fero   += w_fero
        Ev_fero  += w_fero * E
        E2v_fero += w_fero * E**2
        w_antifero = np.exp(lngE[i] + (E /T))
        Z_antifero += w_antifero
        Ev_antifero += w_antifero * E
        E2v_antifero += w_antifero * E ** 2
    Ev_fero *= 1./Z_fero
    cv_fero = (E2v_fero/Z_fero - Ev_fero**2)/T**2
    Ev_antifero *= 1. / Z_antifero
    cv_antifero = (E2v_antifero / Z_antifero - Ev_antifero ** 2) / T ** 2
    return (Ev_fero/N2, cv_fero/N2, Ev_antifero/N2, cv_antifero/N2)

if __name__ == '__main__':
    n = [14]
    # igreneF = []
    # igreneAF = []
    # vcF = []
    # vcAF = []
    for a in n:
        kisi = a
        N2 = kisi**2
```

```
    NTe = N2-1
    T = np.linspace(0.01,2.4,1000)
    energies, lngE = get_energies_and_lnge(kisi)
    E0 = energies[-1]
    energi_fero = []
    cv_fero = []
    energi_antifero = []
    cv_antifero = []
    for i in range(len(T)):
        energis_fero, cvs_fero, energis_antifero, cvs_antifero =
Thermod(T[i], lngE, energies, E0)
        energi_fero.append(energis_fero)
        cv_fero.append(cvs_fero)
        energi_antifero.append(energis_antifero)
        cv_antifero.append(cvs_antifero)
```

LAMPIRAN B

Data Pengukuran Untuk Ukuran Kisi Linier L = 8, 10, 12, 14 sebanyak 20 sampel.

1. Ekstrak Energi (E) dan Panas Jenis (CV)

   Untuk L = 8

Temperatur (T)	Antiferomagnet		Feromagnet	
	Energi (E)	Panas Jenis (CV)	Energi (E)	Panas Jenis (CV)
0.577	1.79869	1.28362	-1.80380	1.28006
0.60092	1.76821	1.25834	-1.77312	1.27682
0.70140	1.65474	0.99918	-1.65623	1.03361
0.80188	1.56122	0.89694	-1.55990	0.91925
0.90236	1.46906	0.95811	-1.46604	0.96868
1.00045	1.36799	1.11481	-1.36460	1.11280
1.10093	1.24632	1.29924	-1.24339	1.29372
1.20141	1.11174	1.34264	-1.10935	1.33719
1.30189	0.98371	1.17981	-0.98190	1.17408
1.40237	0.87694	0.94606	-0.87561	0.94251
1.50046	0.79405	0.75292	-0.79293	0.75193
1.60094	0.72607	0.60834	-0.72497	0.60878
1.70142	0.67043	0.50465	-0.66926	0.50546
1.80190	0.62375	0.42819	-0.62250	0.42888
1.90238	0.58378	0.36977	-0.58248	0.37021
2.00047	0.54980	0.32471	-0.54846	0.32492
2.10095	0.51912	0.28728	-0.51777	0.28732
2.20143	0.49185	0.25639	-0.49051	0.25631
2.30191	0.46743	0.23049	-0.46609	0.23032
2.4	0.44590	0.20896	-0.44459	0.20874

Untuk L = 10

Temperatur (T)	Antiferomagnet		Feromagnet	
	Energi (E)	Panas Jenis (CV)	Energi (E)	Panas Jenis (CV)
0.577	1.79727	1.32037	-1.79443	1.33189
0.60092	1.76574	1.30713	-1.76305	1.28363
0.70140	1.64792	1.03076	-1.64951	1.00704
0.80188	1.55164	0.92072	-1.55349	0.92814
0.90236	1.45740	0.97914	-1.45881	0.97800
1.00045	1.35347	1.15530	-1.35549	1.14479
1.10093	1.22647	1.35889	-1.22991	1.34341
1.20141	1.08796	1.34807	-1.09216	1.35096
1.30189	0.96302	1.12187	-0.96618	1.13685
1.40237	0.86272	0.88277	-0.86436	0.89659
1.50046	0.78519	0.70806	-0.78577	0.71538
1.60094	0.72072	0.58256	-0.72091	0.58344
1.70142	0.66696	0.49208	-0.66725	0.48978
1.80190	0.62112	0.42310	-0.62168	0.42035
1.90238	0.58146	0.36822	-0.58226	0.36624
2.00047	0.54756	0.32441	-0.54851	0.32343
2.10095	0.51689	0.28716	-0.51789	0.28709
2.20143	0.48965	0.25601	-0.49062	0.25664
2.30191	0.46528	0.22974	-0.46616	0.23087
2.4	0.44385	0.20790	-0.44460	0.20937

Untuk L = 12

Temperatur (T)	Antiferomagnet		Feromagnet	
	Energi (E)	Panas Jenis (CV)	Energi (E)	Panas Jenis (CV)
0.577	1.79644	1.34921	-1.79417	1.33640
0.60092	1.76419	1.33614	-1.76261	1.29355

0.70140	1.64544	1.02614	-1.64715	1.02014
0.80188	1.55060	0.90610	-1.55147	0.91726
0.90236	1.45650	0.98983	-1.45731	0.98295
1.00045	1.35035	1.19521	-1.35151	1.19496
1.10093	1.21729	1.42723	-1.21893	1.41875
1.20141	1.07468	1.35143	-1.07630	1.36319
1.30189	0.95248	1.07549	-0.95253	1.08913
1.40237	0.85670	0.84535	-0.85612	0.84470
1.50046	0.78176	0.69151	-0.78160	0.68553
1.60094	0.71835	0.57601	-0.71869	0.57263
1.70142	0.66513	0.48702	-0.66560	0.48764
1.80190	0.61982	0.41771	-0.62011	0.42037
1.90238	0.58069	0.36319	-0.58070	0.36599
2.00047	0.54723	0.32042	-0.54700	0.32251
2.10095	0.51690	0.28448	-0.51650	0.28570
2.20143	0.48986	0.25454	-0.48938	0.25502
2.30191	0.46559	0.22923	-0.46509	0.22917
2.4	0.44417	0.20808	-0.44369	0.20764

Untuk L = 14

Temperatur (T)	Antiferomagnet		Feromagnet	
	Energi (E)	Panas Jenis (CV)	Energi (E)	Panas Jenis (CV)
0.577	1.79709	1.37094	-1.79679	1.33725
0.60092	1.76434	1.35408	-1.76476	1.33144
0.70140	1.64524	1.02658	-1.64624	1.02203
0.80188	1.54961	0.92048	-1.54979	0.93418
0.90236	1.45421	0.99746	-1.45372	1.00114
1.00045	1.34595	1.24896	-1.34526	1.24055
1.10093	1.20412	1.51855	-1.20545	1.50082
1.20141	1.05774	1.33201	-1.05985	1.32539

1.30189	0.94051	1.01170	-0.94295	1.01793
1.40237	0.85066	0.79517	-0.85150	0.81636
1.50046	0.77950	0.66477	-0.77861	0.67619
1.60094	0.71773	0.56853	-0.71646	0.56561
1.70142	0.66471	0.48888	-0.66409	0.48041
1.80190	0.61908	0.42140	-0.61929	0.41390
1.90238	0.57963	0.36562	-0.58047	0.36078
2.00047	0.54602	0.32128	-0.54722	0.31859
2.10095	0.51566	0.28424	-0.51705	0.28305
2.20143	0.48868	0.25373	-0.49013	0.25351
2.30191	0.46450	0.22818	-0.46595	0.22859
2.4	0.44319	0.20692	-0.44457	0.20773